# A RELATIONAL THEORY OF DATATYPES

Chritiene Aarts[1] Roland Backhouse[2] Paul Hoogendijk[*]
Ed Voermans[*] Jaap van der Woude[*3]

December, 1992

[1]Department of Computing Science, Utrecht University, The Netherlands
[2]Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.
[3]CWI, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands.

# Contents

# Chapter 1

# Introduction

Since the observation was first made (e.g. by Hoare [50]) that program structure is related to data structure the notion of type has pervaded many theories of program design, so much so that in our view such a notion has become indispensable. In line with its perceived importance there is now an abundance of type theories, each drawing substance from one or more established areas of mathematics — including category theory, intuitionism and the second order lambda calculus. This monograph explores yet another type theory, this time based on an axiomatic presentation of the theory of binary relations.

Our reasons for embarking on this exploration involved an element of satisfaction and an element of dissatisfaction with current programming research. The element of satisfaction comprises, first, the ever-growing knowledge and understanding of theories of type, second, the pioneering work of Bird and Meertens on economical notations for functional programming and, third, the now well-established literature on the calculation of imperative programs. The element of dissatisfaction arose from a growing frustration with the fundamental limitations of the *functional* programming paradigm within which almost all type theories have been developed up till now, and with the continuing disparity in scale between formal and informal program development. Let us begin with the element of satisfaction.

## 1.1 Type Theory, Category Theory and the Bird-Meertens Formalism

The history of research into type structure as it pertains to programming is something that we do not care or dare to trace. Our own understanding has, however, been substantially influenced from two directions: the work of the "intuitionists", in particular Martin-Löf [66], the Göteborg group [77] and the NuPRL group [29] on a theory of types based on the notion of "propositions-as-types" (this work now being known to have strong connections to the Automath project led by de Bruijn [25]), and the work of category theoreticians on algebraic approaches to program specification [43, 73].

Martin-Löf's theory of types can be characterised as a theory of inductively-defined types. A major attraction of his theory is that there is an elegant scheme underlying the definition of individual types that encourages and facilitates the construction of new types. A contribution of members of the current consortium was to recognise and elaborate on this scheme, leading to the publication of [8]; similar ideas have also been pursued by Dybjer [38] and others.

In the categorical approach to type structure so-called "unique extension properties" are used to characterise types as either the "initial" or "terminal" objects in a category. Hagino [45] proposed a method of type-definition based on this characterisation. Most researchers would concede that the two approaches are formally equivalent but would argue that in nature they are quite distinct, the intuitionistic approach being based on the natural-deduction style of proof development whereas the categorical approach is much more equational and often better suited to program development. On the other hand a major innovation of Martin-Löf's theory was the notion of dependent type, which notion does not seem to be so readily expressible within category theory.

Quite independently of the above work Bird and Meertens have been collaborating for many years on the development of an APL-like notation for functional programs which emphasises economy of expression and calculation. The importance of such economy to programming has been eloquently advocated by Meertens [68] and it would not do justice to his work to try to summarise the arguments here. A significant outcome, however, of this collaboration has been an impressive, albeit limited, calculus of program construction based around the notion of homomorphism on a list structure. The calculus has been used to reformulate existing solutions and to develop ingenious new solutions to many list-programming and other problems [16, 17, 20, 18, 21].

Some few years ago, research began with the aim of extending Bird and Meertens' work on lists to arbitrary, inductively-defined, data types. The conjecture we made at that time and which has since been amply confirmed was that the basic concepts and calculational techniques propounded by Bird and Meertens would be equally relevant and powerful in a more general type-theoretic setting. In the process of conducting this research we became more and more familiar with the categorical approach to type definition, and began to appreciate and further the application of unique extension properties. For accounts of this work refer to [4, 62, 64].

So much for the element of satisfaction. Now to the element of dissatisfaction.

## 1.2 Indeterminacy and Notational Issues

Although endowed with many mathematical niceties, there is, we believe, one overriding reason why purely-functional programming can only be a passing phase in the development of computing science: that is the lack of nondeterminism. Functions are by definition deterministic, but nondeterminism — the ability to postpone decisions, sometimes indefinitely, — has long been recognised as a vital component of any programming calculus. Indeed, the inclusion of nondeterminism is a major desideratum within calculi for imperative programming [36]. On the other hand, notions of type within imperative programming languages are grossly impoverished relative to the same notions in functional languages. Type theory has, until now, made the greatest advances within the functional programming paradigm.

In addition to our dissatisfaction with the determinism of functional programming and the type-poverty of imperative programming, we are becoming more and more distressed with what we perceive as a severe notational flaw that pervades the everyday practice of both imperative and functional programming, namely the ubiquitous use of bound variables. As a consequence formal manipulations become long and unwieldy and can indeed obscure rather than elucidate an argument. The minimisation of bound variables has, of course, long been advocated by category theory as well as being fundamental to the Bird-Meertens formalism. However, mathematical practice and programming practice lag far behind theoretical argument, and we continue to find scope for substantial economies in calculation. For more explanation and discussion of our viewpoint see [5].

So much for the element of dissatisfaction.

## 1.3 The Need For a Relational Framework

The relational calculus has been explored in the past as a framework for programming, for example in [12], [14], [33] and [84]. (This list is certainly by no means exhaustive.) Recently Hoare and He [52] have strongly advocated the view of specifications as relations and the programming process as that of refining a given relation into a (possibly functional) implementation. So far as we know, however, none of this research has combined the relational calculus with type theory.

The need to admit relations, rather than functions, in programming was also much in evidence at a summer school held in September, 1989. At this summer school de Moor lectured on his work on applying a relational calculus to various optimisation problems [74, 76] (such problems being by nature nondeterministic since unique optima are exceptional) and to program inversion [75] whilst Sheeran [86] and Jones [54] reported on the use of relations to describe butterfly circuits and the Fast Fourier Transform.

"Needs", "wishes" or "wouldn't-it-be-nice lists" are all very well, but the art of doing research is to recognise out of the great multitude of outstanding issues those few that can be resolved elegantly and effectively using current knowledge and techniques. The incentive for us to investigate a relational theory of types was the (re)discovery by de Bruin of the notion of "naturality" of polymorphism [26]. (As it turns out, this notion was already known to Reynolds [79] much earlier but its full relevance to program calculation does not seem to have been envisaged. De Bruin's and, more or less simultaneously, Wadler's [91] observation was that naturality of polymorphism explains and indeed predicts several of the most fundamental laws in the Bird-Meertens formalism.) In order to express the notion of "naturality" one is obliged to extend the definition of a type functor (a type constructor and corresponding "map" operator) to a mapping from *relations* to *relations*. In other words, relations are essential to meta-reasoning about polymorphic type constructors but there seems to be no reason why their use should be restricted to the meta-level. One is indeed encouraged to replace the categorical notion of "functor" by a (seemingly) stronger notion of "relator". The ideas underlying, the goals of, and preliminary justification for, a type-oriented theory of relational programming were discussed by Backhouse [1] at the above-mentioned summer school.

## 1.4 Relational Programming

The starting point for the present work is the (already-mentioned) notion of "relational programming" as put forward by Hoare and He [52]. In their view, *specifications* and *implementations* are binary relations on input and output values. An implementation $f$ *satisfies* specification $R$ if

$$f \subseteq R$$

(where a binary relation is regarded as a set of pairs). *Programming* is thus the process of calculating an implementation satisfying a given specification.

Which binary relations count as specifications is quite unrestricted: the whole of the language of mathematics may be used as specification language. Which binary relations count as implementations is fluid: the more we discover about what can and what cannot be efficiently automated the more "higher-level" our programming languages will become. Thus the two notions of specification and implementation are deliberately left vague in order to take account of future developments.

In spite of this vagueness there is still much that can be said about what might constitute a "healthy" theory of relational programming. *Monotonicity*, for example, of the operators in one's implementation language is desirable for "compositionality" of programming: if $\otimes$ is a binary operator, say, on relations monotonicity of $\otimes$ is the statement that

$$R \otimes S \ \subseteq\ U \otimes V \quad \Leftarrow \quad R \ \subseteq\ U \wedge S \subseteq V \ \ .$$

From a programming point of view this is the statement that a specification written in the form $U \otimes V$ can be implemented by finding an implementation $R$ of $U$ and — separately — an implementation $S$ of $V$, and then composing them to form $R \otimes S$.

Given the foregoing preamble, it will come as no surprise to the reader to learn that our principal "healthiness" criterion is that the theory should support a theory of types that encourages and facilitates the introduction of new type structures. Indeed, this whole monograph is devoted to the study of general mechanisms for defining polymorphic type constructors and their associated "catamorphisms" within an axiomatic theory of relations. The sort of type constructors that can be defined using such mechanisms are familiar constructors like *List* and *Tree*; in this sense the monograph offers no surprises. On the other hand, we do present a whole host of mathematical properties which, we argue, testify to the theory's healthiness both from a theoretical and a practical viewpoint. Moreover, we are particularly encouraged by the economy and clarity of our calculations, which is in our view of paramount importance.

`**** Structure of the book ******`

# Chapter 2

# Preliminaries

Every book must make certain assumptions about the knowledge and abilities of its readers, and this one is no exception. The basic assumptions we make are that you have a sound knowledge of elementary predicate calculus and set theory, and that you enjoy algebraic calculations.

It is possible that the notation and terminology we use differ from those that you are used to. The purpose of this chapter is to summarise our own notational preferences and thus avoid any misunderstandings that this may cause. In the first section we summarise our preferred notation for writing down predicates and name several laws that tend to occur frequently in our calculations. The next section is concerned with functions and some of their prominent properties. The section following that summarises the style we use for presenting calculations. The last section is concerned with the *pointwise* relational calculus. This calculus will provide a model of the pointfree calculus that we axiomatise in part 2.

## 2.1 Meta-language

The meta-language we use for conducting proofs is the predicate calculus. We assume the reader is familiar with the predicate calculus, so we content ourselves with a short description. A more extensive account can be found in [36]. For the benefit of those who have read [36], we do not use the *everywhere* operator, denoted by square brackets. We adopt the convention —unless stated otherwise— that the formulae we give are universally quantified over all free variables.

The predicate calculus, or the calculus of boolean structures, consists of two boolean scalars: *true* and *false*. The predicates can be seen as boolean-valued functions. In order to reason about the predicates, some operators are used.

The *equivalence* operator ($\equiv$) is used to denote boolean equality. It has the least binding power of all binary operators. The boolean scalar *true* is an identity for the equivalence.

Equivalence is both associative and transitive. This creates a dilemma as to how to parse expressions involving repeated equivalences such as $X \equiv Y \equiv Z$. Should one parse such an expression associatively – i.e. as $X \equiv (Y \equiv Z)$ or $(X \equiv Y) \equiv Z$ – or conjunctively – as $(X \equiv Y) \wedge (Y \equiv Z)$. Dijkstra and Scholten [36] argue convincingly for the former choice. Their arguments are expressed, however, in a context in which the predicate calculus itself is the object of study. In the present context, where we use the predicate calculus as meta-language and not as object language, it is more appropriate to adopt the conjunctive interpretation of such expressions, and this is what we shall do.

*Disjunction* ($\vee$) is used to model the boolean or, *conjunction* ($\wedge$) models the and. Both these binary operators are symmetric, associative and idempotent. The scalar *true* is a zero for the disjunction and an identity for the conjunction. The scalar *false* acts as an identity for the disjunction and as a zero for the conjunction.

The remaining two binary operators are *implication* ($\Rightarrow$) and *follows-from* ($\Leftarrow$). They have equal binding power, higher than equivalence but less than disjunction and conjunction. Implication and follows-from are formally indistinguishable, since $Y \Leftarrow X \;\equiv\; X \Rightarrow Y$ . Nevertheless it is vital to have both of them available for constructing proofs. In the expression $Y \Leftarrow X$ or $X \Rightarrow Y$ we refer to $X$ as the *antecedent* and to $Y$ as the *consequent*. From the truth of $Y \Leftarrow X \;\equiv\; (X \vee Y \;\equiv\; Y)$ , the reader can establish various properties of follows-from, and thus of implication.

Follows-from is not associative but, as for equivalence, one faces a choice when parsing expressions of the form $X \Leftarrow Y \Leftarrow Z$. Now there are three possibilities. One is to postulate that follows-from is right associative, so that the expression is parsed as $X \Leftarrow (Y \Leftarrow Z)$, the second is to postulate that it is left associative, so that the expression is parsed as $(X \Leftarrow Y) \Leftarrow Z$, and the third — motivated by the transitivity of follows-from — is to read the formula conjunctively as $(X \Leftarrow Y) \;\wedge\; (Y \Leftarrow Z)$. We choose to adopt the last of the three choices. (And the same goes for implication.) Note, however, that, because of the confusion that might occur, we avoid the use of repeated equivalences, implications and follow-froms in one-line expressions, reserving their use solely for multi-line proofs. (See the next section for further explanation.)

As a unary operator we have *negation* ($\neg$). It is written as a prefix operator. We adopt the convention that unary operators have a higher binding power than any binary operator, including function application/composition. Thus negation has the highest binding power. For negation we have the Law of the Excluded Middle, i.e. $X \vee \neg X$ for any predicate $X$. Of course we also have $false \equiv \neg true$ and $X \equiv \neg\neg X$ for any predicate $X$. When calculating with negation, the Laws of de Morgan come in handy: $\neg X \vee \neg Y \equiv \neg(X \wedge Y)$ and $\neg X \wedge \neg Y \equiv \neg(X \vee Y)$ .

Conjunction and disjunction are generalised in the usual way to *universal quantification* and *existential quantification*. We use $P.x$ to indicate that the predicate $P$ might depend on $x$. For predicates $P$ and $Q$, that might depend on $x$, universal quantification is written $\forall(x : P.x : Q.x)$ and read "for all $x$ such that $P.x$ holds, $Q.x$ holds". The existential quantification is written $\exists(x : P.x : Q.x)$ and read "there is an $x$ such that $P.x$ and $Q.x$". In such formulae we refer to $x$ as the dummy; it can be replaced by any other variable without changing the truth of the formulae if we replace its free occurrences in $P.x$ and $Q.x$ . We call $P.x$ the range and $Q.x$ the term. Perhaps redundantly, we mention that the predicates $P$ and $Q$ need not depend on $x$. The range $true$ will be omitted. For the universal quantification we have, among others, the following rules:

- $\forall(x : P.x : Q.x) \equiv \forall(x :: \neg P.x \vee Q.x)$ called *trading*,
- $\forall(x :: \forall(y :: P.x.y)) \equiv \forall(y :: \forall(x :: P.x.y))$ called *interchanging quantifications*,
- $\forall(x : \exists(y : P.y : Q.x.y) : R.x) \equiv \forall(y : P.y : \forall(x : Q.x.y : R.x))$ called *range disjunction*,
- $X \vee \forall(x :: P.x) \equiv \forall(x :: X \vee P.x)$ called $\vee$-$\wedge$ *distributivity*,
- $\forall(x :: P.x) \wedge \forall(x :: Q.x) \equiv \forall(x :: P.x \wedge Q.x)$ called $\forall$-$\wedge$ *distributivity*,
- $\forall(x :: true) \equiv true$ called *term true*,

- $\forall(x:\ \mathit{false}:\ P.x) \quad\equiv\quad \mathit{true}$ called *empty range*,
- $\forall(x:\ x=y:\ P.x) \quad\equiv\quad P.y$ called *one-point rule.*

Rules similar to these for existential quantification can be derived via de Morgan's law

- $\exists(x:\ P.x:\ Q.x) \equiv \neg\forall(x:\ P.x:\ \neg Q.x)$ .

In all formulae that we write the above meta-operators have lower precedence than operators of the object language.

## 2.2 Functions

`** Very drafty **`

As usual we indicate function application by the lower dot ".". The lower dot is right-associative and binds stronger than any other binary operator. If $x$ is an element of type $A$ and $f$ a function from $A$ —called the *domain*— to some other type, we denote the unique image element of $x$ by $f.x$ . To indicate that $f$ is a function to $B$ —called the *range*— from $A$ we write $f \in B \longleftarrow A$ . The choice for the unconventional direction of the arrow is based on the way we denote function application (and composition) of two functions. In case of function application, the argument of a function is placed on the right-hand side of the function. Writing the type information as we do, the domain of the function is placed on the right-hand side of the arrow.

On functions we can define a binary operator, the familiar *composition*. For $g \in C \longleftarrow B$ and $f \in B \longleftarrow A$ we define the composition $g \bullet f \in C \longleftarrow A$ , by

$$(f \bullet g).x \ = \ f.g.x$$

for all $x \in A$. The $\bullet$ is associative.

When working with functions and using them in proofs, the rule *Leibniz* is used frequently. I.e. for $x, y$ and $f$ of the appropriate type we have

$$x \ = \ y \quad\Rightarrow\quad f.x \ = \ f.y \ .$$

If both $A$ and $B$ are lattices with negation, one can define a unary operator on functions to $B$ from $A$ called the *conjugate*. If $f \in B \longleftarrow A$ then we define the conjugate $f^\diamond \in B \longleftarrow A$ by $f^\diamond.x = \neg(f.\neg x)$ , for all $x \in A$. Notice the way the latter expression is parenthesised: we adopt the convention that unary operators take precedence over binary operators.

For other properties of functions, like *injectivity* and *surjectivity* the reader is referred to section 2.4.

## 2.3 Proof Format

For the presentation of equational proofs we use the style introduced by W.H.J. Feijen in [35]. That is, we write

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

In the above proof $R$,$S$ 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 $\sqsubseteq$ . In such cases the connectives are used *conjunctively*; for example $R \sqsubseteq S \sqsubseteq T$ means $(R \sqsubseteq S)$ *and* $(S \sqsubseteq T)$.

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

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

$$
\begin{array}{ll}
\Leftarrow & \quad \{ \text{ hint } \} \\
& q \quad .
\end{array}
$$

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 not given as dictates beforehand, but they are collected in the hints. Thus the proof takes the form

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

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

$$q1 \wedge q2 \quad \Rightarrow \quad R \ = \ T \quad .$$

We draw the reader's attention to such hints by marking them with a bullet (the symbol "$\bullet$" used above).

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

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

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 like:

By construction of $x$:

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

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 keywords "by construction of" alert the reader to the fact that the variables that follow (in this case just $x$) will be assigned particular values during the course of the proof. These assignments are indicated by bullets in the hints. Most often they introduce fresh variables for which appropriate assignments have to be found also — such as $y$ in the above outline.

## 2.4 The Pointwise Relational Calculus

`** Extremely drafty **`

For the moment we take an interest in relations for granted. Since our objective is to study relational datatypes and the relational programming that comes with them, it doesn't hurt to pay a little attention to relations. In this section we briefly discuss the set theoretic notion of relation and the structure of the collection of relations on a given set (space) thereby introducing some notation.

A set theoretic relation between two sets $X$ and $Y$, in that order, is defined to be a subset of the cartesian product

$$X\times Y\ =\ \{(x,y)\,|\,x\in X\ \wedge\ y\in Y\}\ \ ,$$

or, equivalently, a boolean valued function (a predicate) on $X\times Y$.

For a relation $R$ between $X$ and $Y$ and $x\in X$, $y\in Y$ we mostly write $xRy$ instead of $(x,y)\in R$ or $R.(x,y)\ \equiv\ \mathit{true}$ (or $R.(x,y)$).

Some elementary examples of relations are: $\emptyset$, $X\times Y$ and $\{(x,y)\}$, and for $X$ and $Y$ equal the *diagonal* $I_X\ =\ \{(x,x)\,|\,x\in X\}$ . Moreover every function $f\in X\longleftarrow Y$ induces a relation between $X$ and $Y$ via its *graph*:

$$\mathsf{GR}.f\ =\ \{(x,y)\,|\,x\ =\ f.y\}\ \ .$$

As soon as functions are embedded in the relations (for example in the above way) a direction suggests itself: a relation $R$ between $X$ and $Y$ may be interpreted then as a mechanism to be fed with elements of the (right) domain $Y$ which returns elements of the left domain (range) $X$.

The collection of all relations between $X$ and $Y$ inherits the structure of the powerset of $X \times Y$, so we may consider union, intersection and negation (complement) of relations (provided they have the same domains).

In chapters to follow we shall axiomatise this structure via the concept of a lattice. Instead of the set theoretic notations like $\emptyset$, $\subseteq$, $\cup$ and $\cap$ we then use the lattice operations $\perp\!\!\perp$, $\sqsubseteq$, $\sqcup$, $\sqcap$ and denote the full relation $(X \times Y)$ by $\top\!\!\top$.

Like functions, relations may be composed if the corresponding domains match, so for $R \subseteq X \times Y$ and $S \subseteq Y \times Z$ define

$$x(R \circ S)z \quad \equiv \quad \exists(y :\ y \in Y :\ xRy \ \wedge \ ySz) \ \ .$$

The composition is associative and the diagonals serve as (partial) identities. For the collection of all relations on one space $X$ (so $I\!P(X \times X)$) this means that the composition and the diagonal make it into a monoid. This structure will be axiomatised as such.

Unlike functions, relations may be reversed: define and denote

$$y(R^{\cup})x \quad \equiv \quad xRy \ \ .$$

So $R \subseteq X \times Y$ iff $R^{\cup} \subseteq Y \times X$, and on $I\!P(X \times X)$ the reverse "$^{\cup}$" is an inversion that respects the set inclusion and "reverses" the composition. The stirr frying pan symbol "$^{\cup}$" is pronounced accordingly as *wok*, and it will be used in the axiomatisation too.

An interesting bonus is the following interface:

$$(P \ \cap \ R \circ Q^{\cup}) \circ (Q \ \cap \ P^{\cup} \circ R) \ \supseteq \ P \circ Q \ \cap \ R \ \ .$$

which is called the Dedekind rule (exercise: prove it).

Several standard properties of relations may be expressed in terms of the above structure, for example for $R \subseteq X \times Y$:

$$\begin{array}{ll} & R \text{ is total on } Y \\ \equiv & \qquad \{ \text{ definition of total } \} \\ & \forall(y :\ y \in Y :\ \exists(x :\ x \in X :\ xRy)) \\ \equiv & \qquad \{ \text{ definition of composition } \} \end{array}$$

$$
\begin{array}{ll}
 & \forall(y:\ y \in Y:\ y(R^{\cup} \circ R)y) \\
\equiv & \quad \{\ \text{definition of } I_Y\ \} \\
 & I_Y \subseteq R^{\cup} \circ R \quad .
\end{array}
$$

Similarly one may prove

- $R$ is functional iff $R \circ R^{\cup} \subseteq I_X$ ,
- $R$ is injective iff $R^{\cup} \circ R \subseteq I_Y$ ,
- $R$ is surjective iff $I_X \subseteq R \circ R^{\cup}$ ,

and

- $R$ is a function iff $R \circ R^{\cup} = I_X$ .

If $X = Y$, i.e. $R \subseteq X \times X$, we also have

- $R$ is reflexive iff $I_X \subseteq R$ ,
- $R$ is symmetric iff $R \subseteq R^{\cup}$ ,
- $R$ is anti-symmetric iff $R \cap R^{\cup} \subseteq I_X$ ,
- $R$ is transitive iff $R \circ R \subseteq R$ .

The description above look a lot cleaner than the usual ones where dummies and quantifications are all over the place. We therefore only seldomly refer to the set theoretic relations, though it is our main model, but mostly calculate in the axiomatised version. In case we do refer to the set theoretic interpretation we adopt the usual semantics notation to stress the fact that we interpret the (statement about the) relation in the set theoretic model so $[\![\text{Prop}.R]\!]$ is to be read as the set theoretic interpretation of property Prop with respect to the set theoretic interpretation $[\![R]\!]$ of $R$.

# Part I

# Lattice Theory (Elements of, Presented Calculationally)

To begin a book on a theory of datatypes with a substantial part on lattice theory is surely asking for trouble! The reader with little or no previous knowledge is likely to regard such an introduction as a formidable hurdle, and will question whether a textbook specifically devoted to the topic would not be a better place to begin; the reader with more knowledge will be confident that that is indeed the case and will be irritated by our presumption to think otherwise. Nevertheless we would encourage both sets of readers to spare some time reading carefully through the main sections of this part. To avoid the task's becoming a substantial hurdle we offer shortly some guidance on how to approach it dependent on one's prior knowledge.

The inclusion of such a substantial introduction to lattice theory is justified by the part's subtitle —"presented calculationally". A major driving force behind our work is to reduce substantial parts of the programming process to straightforward calculation. There are two challenges here, one being to reduce programming to calculation, the other to straightforward calculation. The latter, as opposed to the former, can only be achieved by utmost concern with the form and presentation of calculational rules. And, of course, that concern must begin at the very beginning — in our case with a calculational presentation of lattice theory and, later, of an axiomatisation of the calculus of relations.

The presentation of lattice theory here departs from that in all texts that we know of in the prominence given to the notion of a "Galois connection" introduced in chapter 5. A Galois connection is a rule connecting two functions to each other having a particularly simple and elegant shape. The recognition of a Galois connection between two functions considerably facilitates calculations with the functions. We shall encounter several such connections throughout the text, amply sufficient to justify presenting the abstract notion at a very early stage. Once mastered, the reader should have no difficulty in recognising many other instances in other application areas.

On the other hand, we do not presume to suggest that this text is a replacement for other texts on lattice theory. We use the qualifier "elements of" in the part's heading as a warning that there is much more to lattice theory than we have time, space or ability to discuss. The selection of topics is very much geared to our immediate needs and you may need to consult other texts if your needs are different from ours.

# Chapter 3

# Extremal Elements

## 3.1 Introduction

Let $\mathcal{A}$ be an arbitrary set. A binary relation $\sqsubseteq$ on $\mathcal{A}$ is said to be *reflexive* if $x \sqsubseteq x$ for all $x \in \mathcal{A}$. It is said to be *anti-symmetric* if $x = y \;\Leftarrow\; x \sqsubseteq y \wedge y \sqsubseteq x$, for all $x$ and $y$ in $\mathcal{A}$. Finally, it is said to be *transitive* if $x \sqsubseteq z \;\Leftarrow\; x \sqsubseteq y \wedge y \sqsubseteq z$ for all $x$, $y$ and $z$ in $\mathcal{A}$.

A *preorder* on $\mathcal{A}$ is a reflexive, transitive relation on $\mathcal{A}$; the pair $(\mathcal{A}, \sqsubseteq)$ is then called a *pre-ordered set.* A *partial order* on $\mathcal{A}$ is an anti-symmetric preorder on $\mathcal{A}$; the pair $(\mathcal{A}, \sqsubseteq)$ is then called a *partially-ordered set* or *poset* for short.

Actually, we assume that these definitions are already familiar to you and you can conjure up several examples of pre-ordered and partially-ordered sets if asked.

Often, lattices would now be introduced by considering an algebra having a binary "meet" operator and a binary "join" operator both of which are idempotent, symmetric and associative, and which collectively obey a certain absorption law. (See e.g. [24].) It is then observed that the carrier of the algebra (the set of values on which the operator is defined) can be ordered by a relation, defined in terms of meet, that is reflexive, anti-symmetric and transitive. Lattices are in this way shown to be partially-ordered sets.

We diverge from this approach. We take as our starting point partially-ordered sets, and consider the construction of a "meet" operator on sets rather than just pairs of elements. The "meet" of a set of elements is called its "infimum". A dual concept is that of "supremum". Both infima and suprema are what we call extremal elements. So too are greatest and least elements.

In this chapter we consider these concepts in some detail. Even if you are already familiar with them it may still be worthwhile reading the chapter in detail because it is here that we first illustrate our calculational style, and where we introduce some fundamental calculational techniques.

## 3.2 Infima

To begin: let $(\mathcal{A}, \sqsubseteq)$ be a partially-ordered set and let $S$ be a subset of $\mathcal{A}$. We say that element $y \in \mathcal{A}$ is a *lower bound* on $S$, or, more concisely, *$y$ is below $S$* if it is at most every element in $S$. That is,

$$(3.1) \quad y \text{ is below } S \quad \equiv \quad \forall(s:\ s \in S:\ y \sqsubseteq s) \quad .$$

Typically, for any given set $S$ there will be many elements below $S$. A *greatest lower bound* or *infimum* of $S$ is a solution of the equation

$$(3.2) \quad x :: \quad \forall(y :: \quad y \text{ is below } S \quad \equiv \quad y \sqsubseteq x) \quad .$$

Clearly, since $\sqsubseteq$ is reflexive, any infimum of $S$ is below $S$. I.e.

$$(3.3) \quad x \text{ solves } (3.2) \quad \Rightarrow \quad x \text{ is below } S \quad .$$

Clearly also, by weakening the equivalence in (3.2) to an implication we have, for all $x \in \mathcal{A}$,

$$(3.4) \quad x \text{ solves } (3.2) \quad \Rightarrow \quad \forall(y :: \quad y \text{ is below } S \quad \Rightarrow \quad y \sqsubseteq x) \quad .$$

The combination of (3.3) and (3.4) is the origin of the name "greatest lower bound" for a solution of (3.2); property (3.3) states that a solution is a lower bound and (3.4) states that a solution is greatest among such lower bounds. The converse of the conjunction of (3.3) and (3.4) is also clearly true: by the transitivity of $\sqsubseteq$ and elementary predicate calculus,

$$(3.5) \quad \forall(y :: \quad y \text{ is below } S \quad \Leftarrow \quad y \sqsubseteq x) \quad \Leftarrow \quad x \text{ is below } S \quad .$$

Hence,

$$(3.6) \qquad \begin{array}{ll} & x \text{ solves } (3.2) \\ \Leftarrow & x \text{ is below } S \quad \wedge \quad \forall(y ::\ y \text{ is below } S \Rightarrow y \sqsubseteq x) \quad . \end{array}$$

To summarise this preliminary discussion, there are two, completely equivalent, specifications of infimum, the first being a solution to (3.2) and the second a solution to

$$(3.7) \qquad x :: \qquad x \text{ is below } S \quad \wedge \quad \forall (y :: \; y \text{ is below } S \Rightarrow y \sqsubseteq x) \quad .$$

Equation (3.7) is the conventional definition of infimum and as explained gives rise to the terminology "greatest lower bound". We, however, prefer (3.2) to (3.7) because the former is more compact and easier to calculate with.

Equation (3.2) may not have a solution but we can assert that it has at most one solution. To see this we observe that

$$(3.8) \qquad u = v \quad \equiv \quad \forall (y :: \; y \sqsubseteq u \quad \equiv \quad y \sqsubseteq v)$$

— which rule we call the rule of *indirect equality*. Next we observe that the left side of the equivalence in (3.2) is totally independent of the dummy $x$. Thus, we can argue that

$$\begin{array}{ll} & y \sqsubseteq u \\ \equiv & \qquad \{u \text{ solves } (3.2) \;\} \\ & y \text{ is below } S \\ \equiv & \qquad \{v \text{ solves } (3.2) \;\} \\ & y \sqsubseteq v \quad . \end{array}$$

That is,

$$u \text{ and } v \text{ both solve } (3.2) \quad \Rightarrow \quad \forall (y :: \; y \sqsubseteq u \quad \equiv \quad y \sqsubseteq v) \quad .$$

In combination with the rule of indirect equality (3.8) this yields the desired uniqueness of a solution of (3.2):

$$u \text{ and } v \text{ both solve } (3.2) \quad \Rightarrow \quad u = v \quad .$$

(The rule of indirect equality is proved by elementary predicate calculus using the reflexivity and anti-symmetry of the ordering relation. Its simplicity beguiles its importance. We discuss the rule in more detail shortly in connection with its extension to proving inclusions.)

We denote the unique solution of (3.2) by $\sqcap.S$ . Also, instead of writing "is below" we silently lift the $\sqsubseteq$ relation to sets. That is, for $x \in \mathcal{A}$ and $S \subseteq \mathcal{A}$

we write $x \sqsubseteq S$ for $x$ is below $S$. Spelling out the definition of "is below" once more, that is to say

$$(3.9) \quad x \sqsubseteq S \quad \equiv \quad \forall(s:\ s \in S:\ x \sqsubseteq s) \quad .$$

(The device used here of "overloading" the $\sqsubseteq$ operator is a common one in mathematics but can lead to confusion if one does not clearly type all variables. Throughout this chapter we use the convention that small letters, like $x$ and $y$, denote elements and capitals, like $S$ and $T$, denote sets of elements. Do not be tempted to instantiate a variable denoting an element with an expression denoting a set, or vice-versa!)

Adopting this convention has the pleasant by-product that (3.2) takes on a particularly concise form. Specifically, if $\sqcap.S$ exists then, for all $y \in \mathcal{A}$,

$$(3.10) \quad y \sqsubseteq S \quad \equiv \quad y \sqsubseteq \sqcap.S \quad .$$

A *complete lattice* is a partially-ordered set $(\mathcal{A}, \sqsubseteq)$ in which $\sqcap.S$ exists for all subsets $S$ of $\mathcal{A}$. Throughout the rest of this section we assume that we are dealing with a complete lattice. The alternative is to tediously preface every statement involving $\sqcap.S$ for some $S$ with "assuming $\sqcap.S$ exists".

Note that the right side of (3.10) can be trivially made true by instantiating $y$ to $\sqcap.S$ . We obtain the simple but powerful property

$$(3.11) \quad \sqcap.S \sqsubseteq S \quad .$$

Equation (3.10) is an instance of a very important concept called a Galois connection that will be discussed later. For the moment it suffices to observe that (3.10) links the function $\sqcap$ with universal quantification (the universal quantification that is obtained by expanding the definition of $y \sqsubseteq S$). A consequence is that $\sqcap$ inherits certain basic properties of universal quantification. To see what these properties are we proceed in two steps. The first step is to explore the "is below" operator. Three elementary properties are

$$(3.12) \quad y \sqsubseteq \{x\} \quad \equiv \quad y \sqsubseteq x \qquad \text{the } \textit{one-point rule},$$

$$(3.13) \quad y \sqsubseteq S \cup T \quad \equiv \quad y \sqsubseteq S \ \wedge\ y \sqsubseteq T \qquad \text{the } \textit{range-disjunction rule},$$

$$(3.14) \quad y \sqsubseteq \emptyset \quad \equiv \quad \mathit{true} \qquad \text{the } \textit{empty range rule},$$

where $\emptyset$ denotes the empty set. In combination with (3.10) these three rules translate into properties of $\sqcap$. We give the rules the same names.

$$(3.15)\quad \sqcap.\{x\} \;=\; x \qquad\qquad \text{the } \textit{one-point rule},$$

$$(3.16)\quad \sqcap.(S \cup T) \;=\; (\sqcap.S) \sqcap (\sqcap.T) \qquad\qquad \text{the } \textit{range-disjunction rule},$$

$$(3.17)\quad y \sqsubseteq \sqcap.\emptyset \qquad\qquad \text{the } \textit{empty-range rule},$$

for all $y \in \mathcal{A}$. For convenience, we used in (3.16) the binary version of the supremum operator which is defined as

$$(3.18)\quad x \sqcap y \;=\; \sqcap.\{x,y\} \;\;.$$

The proofs of all these properties are very straightforward but it is nevertheless worthwhile discussing them because the techniques are very fundamental. Note that (3.15) and (3.16) are statements of equalities whereas the specification of $\sqcap.S$ (see e.g. (3.10)) involves only inclusions in which $\sqcap.S$ appears on the bigger side. Thus we cannot prove a statement of the form $x = \sqcap.S$ by proving both $x \sqsubseteq \sqcap.S$ and $\sqcap.S \sqsubseteq x$ since, at this point in time, we have no means of proving the latter inclusion. The trick is to use the rule of indirect equality (3.8)

$$(3.19)\quad x = \sqcap.S \;\equiv\; \forall(y :: \; y \sqsubseteq x \;\equiv\; y \sqsubseteq \sqcap.S) \;\;,$$

with $u$ instantiated to $x$ and $v$ instantiated to $\sqcap.S$ .

Let's see how this works in the case of (3.12) and (3.15). First, (3.12) follows because

$$\begin{array}{ll} & y \sqsubseteq \{x\} \\ \equiv & \qquad \{\; (3.9) \;\} \\ & \forall(z :\; z \in \{x\} :\; y \sqsubseteq z) \\ \equiv & \qquad \{\; \text{one-point rule of universal quantification} \;\} \\ & y \sqsubseteq x \quad . \end{array}$$

Now combining (3.10) with (3.12) we have, for all $y \in \mathcal{A}$,

$$\begin{array}{ll} & y \sqsubseteq \sqcap.\{x\} \\ \equiv & \qquad \{\; \text{characterisation: } (3.10) \;\} \\ & y \sqsubseteq \{x\} \\ \equiv & \qquad \{\; (3.12) \;\} \\ & y \sqsubseteq x \quad . \end{array}$$

Applying (3.19) we conclude that (3.15) is also true.

Now we consider (3.13) and (3.16). The former follows because, for all $y \in \mathcal{A}$,

$$\begin{array}{ll}
 & y \sqsubseteq S \cup T \\
\equiv & \qquad \{ \ (3.9) \ \} \\
 & \forall(z :\ z \in S \cup T :\ y \sqsubseteq z) \\
\equiv & \qquad \{ \ \text{range-disjunction rule for universal quantification} \ \} \\
 & \forall(z :\ z \in S :\ y \sqsubseteq z) \ \wedge \ \forall(z :\ z \in T :\ y \sqsubseteq z) \\
\equiv & \qquad \{ \ (3.9) \ \} \\
 & y \sqsubseteq S \ \wedge \ y \sqsubseteq T \quad .
\end{array}$$

Combining (3.13) with (3.10) we have, for all $y \in \mathcal{A}$,

$$\begin{array}{ll}
 & y \sqsubseteq \sqcap.(S \cup T) \\
\equiv & \qquad \{ \ \text{characterisation: (3.10)} \ \} \\
 & y \sqsubseteq S \cup T \\
\equiv & \qquad \{ \ (3.13) \ \} \\
 & y \sqsubseteq S \ \wedge \ y \sqsubseteq T \\
\equiv & \qquad \{ \ \text{characterisation: (3.10)} \ \} \\
 & y \sqsubseteq \sqcap.S \ \wedge \ y \sqsubseteq \sqcap.T \\
\equiv & \qquad \{ \ \text{range-disjunction and one-point rules} \\
 & \qquad\quad \text{for universal quantification} \ \} \\
 & \forall(z :\ z \in \{\sqcap.S, \sqcap.T\} :\ y \sqsubseteq z) \\
\equiv & \qquad \{ \ \text{(3.9) and (3.18)} \ \} \\
 & y \sqsubseteq (\sqcap.S) \sqcap (\sqcap.T) \quad .
\end{array}$$

Applying (3.19) we conclude that (3.16) is indeed true.

This completes the discussion of (3.13) and (3.16). It remains to verify (3.14) and (3.17). By now the strategy should be familiar. We have, for all $y \in \mathcal{A}$,

$$\begin{array}{ll}
 & y \sqsubseteq \emptyset \\
\equiv & \qquad \{ \ (3.9) \ \} \\
 & \forall(z :\ z \in \emptyset :\ y \sqsubseteq z) \\
\equiv & \qquad \{ \ \text{empty-range rule for universal quantification} \ \} \\
 & \mathit{true} \quad .
\end{array}$$

This is (3.14); its counterpart (3.17) follows from

$$
\begin{array}{ll}
 & y \sqsubseteq \sqcap.\emptyset \\
\equiv & \qquad \{ \text{ characterisation: (3.10) } \} \\
 & y \sqsubseteq \emptyset \\
\equiv & \qquad \{ \text{ (3.14) } \} \\
 & true \quad .
\end{array}
$$

Property (3.17) says that $\sqcap.\emptyset$ is the biggest element in the lattice. It is so special that it is worth giving it a special notation: we shall henceforth denote $\sqcap.\emptyset$ by $\top\!\!\top$ and call it *top*. The defining property of top is thus (3.17): for all $y \in \mathcal{A}$,

$$(3.20) \quad y \sqsubseteq \top\!\!\top \quad .$$

(A common convention is to use the symbol $\top$ for top. Whilst in printed documents $\top$ and $T$ are readily distinguishable they are not so in hand-written form. For that reason we choose to break with convention.)

An important proof technique was illustrated by the above calculations. Specifically, we established the equality of two poset elements $x$ and $z$ by establishing that, for arbitrary poset element $y$, $y \sqsubseteq x \quad \equiv \quad y \sqsubseteq z$. (See equation (3.8) and references to it.) This technique will be prevalent in the discussion of Galois connections in chapter 5. To reinforce its importance let us give it the status of a named theorem. At the same time let us generalise the technique to proving inclusions as well as equalities.

**Theorem 3.21 (Indirect Equality and Inclusion)** Let $x$ and $y$ be elements of a poset $(\mathcal{A}, \sqsubseteq)$ both satisfying predicate $p$. Then equivalent are

$$
\begin{array}{l}
x = y \quad , \\
\forall(z :\ p.z :\ z \sqsubseteq x \ \equiv\ z \sqsubseteq y) \quad , \\
\forall(z :\ p.z :\ x \sqsubseteq z \ \equiv\ y \sqsubseteq z) \quad .
\end{array}
$$

We call this *the rule of indirect equality*. Also equivalent are

$$
\begin{array}{l}
x \sqsubseteq y \quad , \\
\forall(z :\ p.z :\ z \sqsubseteq x \ \Rightarrow\ z \sqsubseteq y) \quad , \\
\forall(z :\ p.z :\ x \sqsubseteq z \ \Leftarrow\ y \sqsubseteq z) \quad .
\end{array}
$$

We call this *the rule of indirect inclusion*.

□

The proof of this theorem — a simple exercise in the predicate calculus — is left to the reader. In carrying out the exercise it is worth noting the minimum requirements on the ordering relation needed to establish the two parts individually. Together they add up to the requirement that $\sqsubseteq$ is reflexive, anti-symmetric and transitive. The converse is also true! That $\sqsubseteq$ is reflexive, anti-symmetric and transitive is *equivalent* to the conjunction of the two rules. (For a precise statement of this equivalence see exercise 3.28.) This is a significant observation because it means that resorting to proofs of equality or inclusion by means of indirect proof does not weaken one's possibilities.

Often in our use of the rules the predicate $p$ is identically true; in such cases we omit reference to the predicate. In some circumstances, however, it is advantageous to instantiate $p$ to a non-vacuous predicate. If that is the case we refer to $p$ as the *domain predicate*.

In the course of stating and establishing (3.16) the binary operator $\sqcap$ was introduced. From its definition (3.18) and spelling out (3.10) we obtain

(3.22) $z \sqsubseteq x \sqcap y \quad \equiv \quad z \sqsubseteq x \quad \wedge \quad z \sqsubseteq y \quad .$

Easy consequences of this equation are:

(3.23) $x \sqcap x \quad = \quad x$ $\qquad \sqcap$ is *idempotent*,

(3.24) $x \sqcap (y \sqcap z) \quad = \quad (x \sqcap y) \sqcap z$ $\qquad \sqcap$ is *associative*,

(3.25) $x \sqcap y \quad = \quad y \sqcap x$ $\qquad \sqcap$ is *symmetric*.

We also have the important relationship between the partial ordering $\sqsubseteq$ and $\sqcap$, namely:

(3.26) $x \sqsubseteq y \quad \equiv \quad x = x \sqcap y \quad .$

Let us prove (3.26) just to illustrate the generalisation to inclusions introduced in theorem 3.21.

$$
\begin{array}{ll}
 & x \sqcap y \;=\; x \\
\equiv & \qquad \{ \text{ indirect equality: 3.21 } \} \\
 & \forall(z :: \; z \sqsubseteq x \sqcap y \;\equiv\; z \sqsubseteq x) \\
\equiv & \qquad \{ \text{ (3.22) } \}
\end{array}
$$

$$
\begin{array}{ll}
 & \forall(z ::\ z \sqsubseteq x \ \wedge\ z \sqsubseteq y \quad \equiv \quad z \sqsubseteq x) \\
\equiv & \qquad \{\ \text{predicate calculus}\ \} \\
 & \forall(z ::\ z \sqsubseteq x \ \Rightarrow\ z \sqsubseteq y) \\
\equiv & \qquad \{\ \text{indirect inclusion: 3.21}\ \} \\
 & x \sqsubseteq y \quad .
\end{array}
$$

**Exercise 3.27** Other properties inherited by infima from universal quantification are

**a** $\sqcap.(x :\ x \in S :\ f.x)\ \sqcap\ \sqcap.(x :\ x \in S :\ g.x) \quad = \quad \sqcap.(x :\ x \in S :\ f.x \sqcap g.x)$ ,

**b** $S \neq \emptyset \quad \Rightarrow \quad (a\ \sqcap\ \sqcap.S \quad = \quad \sqcap.(x :\ x \in S :\ a \sqcap x))$ ,

**c** $\sqcap.(x :\ x \in S :\ \top\!\!\top) \quad = \quad \top\!\!\top$ .

Prove these properties, identifying clearly the corresponding rule for universal quantification.

□

**Exercise 3.28**

**a** Show that relation $R$ is reflexive and anti-symmetric implies

$$\forall(x,y ::\ x = y \quad \equiv \quad \forall(z ::\ zRx\ \equiv\ zRy)) \quad .$$

**b** Show that relation $R$ is reflexive and transitive equivales

$$\forall(x,y ::\ xRy \quad \equiv \quad \forall(z ::\ zRx\ \Rightarrow\ zRy)) \quad .$$

**c** Show that relation $R$ is reflexive, transitive and anti-symmetric equivales

$$
\begin{array}{ll}
 & \forall(x,y ::\ x = y \quad \equiv \quad \forall(z ::\ zRx\ \equiv\ zRy)) \\
\wedge & \forall(x,y ::\ xRy \quad \equiv \quad \forall(z ::\ zRx\ \Rightarrow\ zRy)) \quad .
\end{array}
$$

□

## 3.3 Suprema

We have introduced infima and examined some of their properties. We now want to introduce the dual concept — *supremum* or *least upper bound*. If $(\mathcal{A}, \sqsubseteq)$ is a poset then so is $(\mathcal{A}, \sqsupseteq)$ where $\sqsupseteq$ is the converse of $\sqsubseteq$, i.e.

$$x \sqsupseteq y \quad \equiv \quad y \sqsubseteq x$$

for all $x, y \in \mathcal{A}$. The supremum operator, denoted by $\sqcup$, in a poset $(\mathcal{A}, \sqsubseteq)$ is defined to be the infimum operator in the dual poset $(\mathcal{A}, \sqsupseteq)$. That is, for $x \in \mathcal{A}$ and $S \subseteq \mathcal{A}$, when $\sqcup.S$ exists it is unique and satisfies

(3.29) $x \sqsupseteq \sqcup.S \quad \equiv \quad x \sqsupseteq S$

where

(3.30) $x \sqsupseteq S \quad \equiv \quad \forall(y :\ y \in S :\ x \sqsupseteq y) \ .$

(It is suggested that you read $x \sqsupseteq S$ as $x$ "is above" $S$.)

This definition by duality is very powerful because we can claim at one stroke that all properties of infima in the previous section are dualisable to suprema by replacing $\sqcap$ by $\sqcup$ and $\sqsubseteq$ by $\sqsupseteq$. Here then are the principal rules:

(3.31) $\sqcup.S \sqsupseteq S \ ,$

(3.32) $\sqcup.\{x\} \ = \ x \ ,$

(3.33) $\sqcup.(S \cup T) \ = \ (\sqcup.S) \sqcup (\sqcup.T) \ ,$

(3.34) $y \sqsupseteq \sqcup.\emptyset \ ,$

(3.35) $z \sqsupseteq x \sqcup y \ \equiv \ z \sqsupseteq x \ \wedge \ z \sqsupseteq y \ ,$

(3.36) $x \sqcup y \ = \ \sqcup.\{x, y\} \ ,$

(3.37) $x \sqcup x \ = \ x \ ,$

(3.38) $(x \sqcup y) \sqcup z \ = \ (x \sqcup y) \sqcup z \ ,$

(3.39) $x \sqcup y \ = \ y \sqcup x \ ,$

(3.40) $x \sqsupseteq y \ \equiv \ x = x \sqcup y \ .$

The supremum of the empty set, like its infimum, is sufficiently special to deserve a special symbol. We use the symbol $\perp\!\!\perp$ and call it *bottom*. (More conventional is to use the symbol $\perp$, but see our remarks on the choice of the symbol $\top\!\!\top$

for a justification of this divergence from established practice.) Bottom has the defining property that for all $y \in \mathcal{A}$,

$$(3.41)\quad y \sqsupseteq \perp\!\!\perp \ .$$

There is one more rule that establishes a useful relationship between suprema and infima. It is that, for all subsets $S$ of $\mathcal{A}$, the supremum $\sqcup.S$ exists provided that the infimum $\sqcap.(y:\ y \sqsupseteq S:\ y)$ exists and, in that case, they are equal. That is to say,

$$(3.42)\quad \sqcup.S \quad = \quad \sqcap.(y:\ y \sqsupseteq S:\ y)$$

whenever the right side of the equation exists. (The converse also holds. See exercise 3.46.)

To show that this equation holds it suffices to assume that the right side exists and establish that it satisfies the specification (3.29) of $\sqcup.S$ .

Let $\hat{S}$ denote the set of all elements above $S$. I.e.

$$(3.43)\quad x \in \hat{S} \quad \equiv \quad x \sqsupseteq S \ .$$

The assumption is then that $\sqcap.\hat{S}$ exists and we have to show that it meets (3.29). So we have to prove, for all $x \in \mathcal{A}$,

$$x \sqsupseteq \sqcap.\hat{S} \quad \equiv \quad x \sqsupseteq S \ .$$

For the first time we are obliged to use a “ping-pong” argument — i.e. a proof of equivalence via mutual implication. The reason is that the characterising property of infima only allows us to relate infima to elements below themselves whereas the characterising property of suprema does the opposite. Because of the asymmetry in (3.42) there is an asymmetry in the “ping” and “pong” components. Follows-from is straightforward:

$$\begin{array}{ll} & x \sqsupseteq S \\ \equiv & \qquad \{\ (3.43)\ \} \\ & x \in \hat{S} \\ \Rightarrow & \qquad \{\ (3.11) \text{ with } S := \hat{S}\ \} \\ & x \sqsupseteq \sqcap.\hat{S} \quad . \end{array}$$

To prove implication we begin by simplifying the proof obligation:

$$
\begin{array}{ll}
 & x \sqsupseteq \sqcap.\hat{S} \quad \Rightarrow \quad x \sqsupseteq S \\
\equiv & \qquad \{ \ \text{(3.30), predicate calculus} \ \} \\
 & \forall(s : \ s \in S : \ x \sqsupseteq \sqcap.\hat{S} \quad \Rightarrow \quad x \sqsupseteq s) \\
\equiv & \qquad \{ \ \text{indirect inclusion: dual of theorem 3.21} \ \} \\
 & \forall(s : \ s \in S : \ \sqcap.\hat{S} \sqsupseteq s) \quad .
\end{array}
$$

To establish this universal quantification let us assume $s \in S$. Then

$$
\begin{array}{ll}
 & \sqcap.\hat{S} \sqsupseteq s \\
\equiv & \qquad \{ \ \text{characterisation: (3.10)} \ \} \\
 & \hat{S} \sqsupseteq s \\
\equiv & \qquad \{ \ \text{definition of ``is below'': (3.9)} \ \} \\
 & \forall(y : \ y \in \hat{S} : \ y \sqsupseteq s) \\
\equiv & \qquad \{ \ \text{definition of } \hat{S}\text{: (3.43)} \ \} \\
 & \forall(y : \ y \sqsupseteq S : \ y \sqsupseteq s) \\
\equiv & \qquad \{ \ \text{(3.30) and predicate calculus, } s \in S \ \} \\
 & \mathit{true} \quad .
\end{array}
$$

This completes the proof.

The dual of (3.42) also holds of course. We have

(3.44) $\sqcap.S \quad = \quad \sqcup.(y : \ y \sqsubseteq S : \ y)$

whenever the right side exists. The most important consequence of these two properties is that completeness of a lattice can be defined either in terms of infima, or of suprema, or both. Specifically:

**Theorem 3.45** The following are equivalent:

- Poset $(\mathcal{A}, \sqsubseteq)$ is a complete lattice.
- All infima exist in poset $(\mathcal{A}, \sqsubseteq)$ .
- All suprema exist in poset $(\mathcal{A}, \sqsubseteq)$ .
- All infima and suprema exist in poset $(\mathcal{A}, \sqsubseteq)$ .

□

**Exercise 3.46**

**a** Show that $x \sqcup (x \sqcap y) \;=\; x \;=\; x \sqcap (x \sqcup y)$ for all $x$ and $y$.

**b** Show that if the supremum $\sqcup.S$ exists then so does the infimum $\sqcap.(y : y \sqsupseteq S : y)$ .

□

## 3.4 Greatest and Least Elements

In this section we introduce some variations on the definitions of infimum and supremum that we have been working with until now. In particular we introduce local infima and suprema. Different notions of locality are possible. One such notion is captured by the definitions of least and greatest element of a set:

**Definition 3.47** For $Y \subseteq \mathcal{A}$, $x$ is called a *least* element of $Y$ iff $x \in Y$ and $x \sqsubseteq Y$ . Dually, $x$ is called a *greatest* element of $Y$ iff $x \in Y$ and $Y \sqsubseteq x$ .
□

Informally, $x \in Y$ is a least element if it is at most any other $y \in Y$ . We will denote a least element of a subset $Y$ by $\underline{\mathbf{min}}.Y$. The notation $\underline{\mathbf{min}}.(x : P.x : f.x)$ is also used instead of the more conventional $\underline{\mathbf{min}}.\{f.x \mid P.x\}$ . We will denote the greatest element of a subset $Y$ by $\underline{\mathbf{max}}.Y$. Occasionally we use $\underline{\mathbf{max}}.(x : P.x : f.x)$ instead of $\underline{\mathbf{max}}.\{f.x \mid P.x\}$ .

The existence of least or greatest elements is of course not guaranteed. But, where they exist, uniqueness is guaranteed and there is an obvious relationship to the infimum and supremum of the given set:

**Theorem 3.48** For all $Y \subseteq \mathcal{A}$ and $x \in \mathcal{A}$ we have the following:

**a** $x = \underline{\mathbf{min}}.Y \;\equiv\; x \in Y \wedge x = \sqcap.Y$ ,

**b** $x = \underline{\mathbf{max}}.Y \;\equiv\; x \in Y \wedge x = \sqcup.Y$ .
□

The definitions of infimum and supremum admit a slight generalisation whereby the bound of a set is not sought within the poset but in a superset of that set:

**Definition 3.49** For $Z \subseteq \mathcal{A}$ and $Y \subseteq Z$, we call $x$ the *infimum of* $Y$ *in* $Z$ iff $x \in Z$ and, for all $z \in Z$,

$$z \sqsubseteq x \;\equiv\; z \sqsubseteq Y \;.$$

The unique solution of this equation, if it exists, is denoted by $\sqcap_Z.Y$ .

□

Note that $\sqcap_Y.Y \;=\; \underline{\mathbf{min}}.Y$ and $\sqcap_{\mathcal{A}}.Y \;=\; \sqcap.Y$ . Furthermore we have the following easily verified property:

**Property 3.50** For $X \subseteq Y \subseteq Z \subseteq \mathcal{A}$ we have, provided $\sqcap_Y.X$ and $\sqcap_Z.X$ exist,

**a** $\sqcap_Y.X \sqsubseteq \sqcap_Z.X$ ,

**b** $\sqcap_Y.X = \sqcap_Z.X \quad\equiv\quad \sqcap_Z .X \in Y$ .

□

Property 3.50(**b**) is often used in a weaker form $\sqcap_Y.X \;=\; \sqcap_Z.X \;\Leftarrow\; \sqcap_Z.X \in Y$ . In this form it can be used to prove that a subset of a complete lattice is a complete lattice itself with the same infimum.

**Exercise 3.51** Show that if $\sqsubseteq$ is a total ordering then, for all non-empty, finite subsets $S$, $\sqcap.S$ exists iff $\underline{\mathbf{min}}.S$ exists.

□

**Exercise 3.52** Prove for $(\mathcal{A}, \sqsubseteq)$ a complete lattice and $S$ and $T$ subsets of $\mathcal{A}$ : $\sqcap.S \sqsubseteq \sqcap.T \quad\Leftarrow\quad S \supseteq T$ . What is the dual property?

□

# Chapter 4

# Junctivity and Continuity

In this chapter we look at functions on lattices and detail a hierarchy of desirable properties of such functions. The terminology and much of the presentation is borrowed, with appropriate adjustments, from Dijkstra and Scholten [36, chap. 6]. Indeed, several of the theorems presented here appear in their book, albeit in a different setting. Some of their theorems have been omitted because they rely on distributivity properties that are not generally true in a lattice, or because they are not relevant to our current goals.

## 4.1 Junctivity Types

The specific concern of this section is a classification of functions on lattices according to conditions under which they commute with the supremum and/or infimum operators. The classification is derived from a classification of indexed bags of lattice elements which we now define.

To increase the compactness of a number of theorems it is useful to extend function application silently from elements to sets. Specifically, if $f$ is a function and $S$ is a subset of its domain we write $f.S$ for $\{s : s \in S : f.s\}$ . (Naming conventions with regard to variables will always be clearly stated so that there is no doubt as to what is intended in a given formula.)

In the following definition we assume for the sake of simplicity that we are dealing with complete lattices. Later we discuss a revised definition relevant to the case that the posets are not complete.

**Definition 4.1 ($\sqcup$-Junctivity Types)** Let $(\mathcal{A}, \sqsubseteq)$ and $(\mathcal{B}, \sqsubseteq)$ be complete lattices and suppose $f \in \mathcal{A} \longleftarrow \mathcal{B}$. Let $S$ be an arbitrary subset of $\mathcal{B}$. Then we say that $f$ is $S$-$\sqcup$-*junctive* iff

$$(4.2) \quad f.\sqcup.S \;=\; \sqcup.f.S \quad .$$

Furthermore we say that $f$ is *universally* $\sqcup$-junctive if $f$ is $S$-$\sqcup$-junctive for *all* subsets $S$, *positively* $\sqcup$-junctive if $f$ is $S$-$\sqcup$-junctive for all *non-empty* subsets $S$, and *finitely* $\sqcup$-junctive if $f$ is $S$-$\sqcup$-junctive for all *finite* subsets $S$.
□

Other junctivity types (for example denumerable $\sqcup$-junctivity) can be added to this list in an obvious way. We reserve the shortest term — plain $\sqcup$-*junctive* — for the most frequently occurring junctivity type, namely finite, positive $\sqcup$-junctivity. With this understanding, it should be obvious that "finite" in the definition of $\sqcup$-junctivity may be replaced by "of size two". That is, $f$ is $\sqcup$-junctive if and only if for all $x, y \in \mathcal{B}$, $f.(x \sqcup y) \;=\; f.x \;\sqcup\; f.y$ . (Formally an inductive proof over the size of the set is needed to verify this claim.)

The definition of $\sqcap$-junctivity types is completely analogous and will be taken for granted.

Occasionally $\mathcal{A}$ and $\mathcal{B}$ are not complete lattices in which case equation (4.2) can be meaningless. The only case we consider in which this occurs is in chapter 5. There we shall use the term "existentially $\sqcup$-junctive" with the following meaning. Function $f \in \mathcal{A} \longleftarrow \mathcal{B}$ is *existentially* $\sqcup$-*junctive* iff for all $S \subseteq \mathcal{B}$, $f.\sqcup.S$ satisfies the specification of $\sqcup.f.S$ whenever $\sqcup.S$ exists.

In definition 4.1 the different types of junctivity are obtained by restricting the cardinality of the set. "Continuity" properties are obtained by another sort of restriction.

**Definition 4.3** Let $(\mathcal{A}, \sqsubseteq)$ be a partially-ordered set and let $S$ be a subset of $\mathcal{A}$. Then $S$ is said to be *totally ordered* or *a chain* iff $x \sqsubseteq y$ or $y \sqsubseteq x$ for all $x, y \in S$.
□

**Definition 4.4 (Continuity Types)** Let $(\mathcal{A}, \sqsubseteq)$ and $(\mathcal{B}, \sqsubseteq)$ be complete lattices and suppose $f \in \mathcal{A} \longleftarrow \mathcal{B}$. Then we say that $f$ is universally $\sqcup$-*continuous* iff $f$ is $S$-$\sqcup$-junctive for all chains $S$. The terms *positively* $\sqcup$-*continuous* and *finitely* $\sqcup$-*continuous* are defined as the corresponding junctivity types, namely by appropriately quantifying over the chains in the definition of $\sqcup$-continuous. Likewise, we define $\sqcap$-*continuous*, *universally* $\sqcap$-*continuous*, *positively* $\sqcap$-*continuous* and *finitely* $\sqcap$-*continuous.*
□

## 4.2 Monotonicity

It should be obvious from the definitions of the various $\sqcup$-junctivity and $\sqcup$-continuity types that they form a hierarchy. Each continuity property is weaker than its corresponding junctivity property; universal $\sqcup$-junctivity is the strongest property and finite, positive $\sqcup$-continuity is the weakest. These two extremes will be the most relevant in later chapters and only occasionally will we consider a junctivity or continuity type in between.

"Finite, positive $\sqcup$-continuity" is a bit of a mouthful, but it coincides with the notion of monotonicity (sometimes called *isotonicity*) as we now show.

**Definition 4.5 (Monotonicity)** Let $(\mathcal{A},\sqsubseteq)$ and $(\mathcal{B},\sqsubseteq)$ be two partially ordered sets. Function $f \in \mathcal{A} \longleftarrow \mathcal{B}$ is said to be *monotonic* iff

$$\forall(x,y :: \; f.x \sqsubseteq f.y \;\Leftarrow\; x \sqsubseteq y) \quad .$$

$\Box$

**Theorem 4.6** The following are all equivalent:

**a** $f$ is monotonic.

**b** $f$ is finitely, positively $\sqcup$-continuous.

**c** $f$ is finitely, positively $\sqcap$-continuous.

**Proof** We shall take for granted that "finite and positive" may be replaced by "of size two" as remarked earlier. Duality considerations permit us to restrict ourselves to a proof of the equivalence of **a** and **b**.

$$\begin{array}{ll}
 & f \text{ is finitely, positively } \sqcup\text{-continuous} \\
\equiv & \quad \{ \text{ definition, above remark } \} \\
 & \forall(x,y :: \; f.x \sqcup f.y \;=\; f.(x \sqcup y) \;\Leftarrow\; x \sqsubseteq y) \\
\equiv & \quad \{ \text{ (3.40) } \} \\
 & \forall(x,y :: \; f.x \sqcup f.y \;=\; f.(x \sqcup y) \;\Leftarrow\; x \sqcup y \;=\; y) \\
\equiv & \quad \{ \text{ calculus } \} \\
 & \forall(x,y :: \; f.x \sqcup f.y \;=\; f.y \;\Leftarrow\; x \sqcup y \;=\; y) \\
\equiv & \quad \{ \text{ (3.40) } \} \\
 & \forall(x,y :: \; f.x \sqsubseteq f.y \Leftarrow x \sqsubseteq y) \\
\equiv & \quad \{ \text{ definition } \} \\
 & f \text{ is monotonic} \quad .
\end{array}$$

□

One might ask why we have seen fit to introduce such a devious notion as "finitely, positively $\sqcup$- or $\sqcap$-continuous" when the notion can be defined so much more simply. One answer is that it is now clear that a function possessing any one of the above-mentioned junctivity or continuity types is automatically guaranteed to be monotonic. This, on its own, is a good enough justification for the deviousness. A second answer is that we intend shortly to present a couple of theorems that are true of all junctivity and continuity types, and thus also of monotonicity.

Very often monotonicity of a function is obvious. If that is the case, it helps to know that establishing $S$-$\sqcup$- or $S$-$\sqcap$-junctivity for some given $S$ (or class of subsets $S$) involves proving only one inclusion, the other being automatically valid. Specifically we have:

**Theorem 4.7** For all monotonic functions $f$ and all subsets $S$ of $\mathcal{A}$ for which $\sqcap.S$ and $\sqcap.f.S$ exist,

$$f.\sqcap.S \sqsubseteq \sqcap.f.S \quad .$$

Dually, for all subsets $S$ of $\mathcal{A}$ for which $\sqcup.S$ and $\sqcup.f.S$ exist,

$$f.\sqcup.S \sqsupseteq \sqcup.f.S \quad .$$

**Proof**

$$\begin{array}{ll}
 & f.\sqcap.S \sqsubseteq \sqcap.f.S \\
\equiv & \qquad \{ \text{ characterisation: (3.10) } \} \\
 & f.\sqcap.S \sqsubseteq f.S \\
\equiv & \qquad \{ \text{ definition of } f.S\text{, (3.9) } \} \\
 & \forall(s :\ s \in S :\ f.\sqcap.S \sqsubseteq f.s) \\
\Leftarrow & \qquad \{ \bullet \quad f \text{ is monotonic } \} \\
 & \forall(s :\ s \in S :\ \sqcap.S \sqsubseteq s) \\
\equiv & \qquad \{ \text{ (3.11) } \} \\
 & \mathit{true} \quad .
\end{array}$$

□

One class of functions that are simultaneously existentially $\sqcup$- and $\sqcap$-junctive are the *poset-isomorphisms*.

**Definition 4.8** If $\mathcal{A}$ and $\mathcal{B}$ are posets and $f \in \mathcal{A} \longleftarrow \mathcal{B}$ then $f$ is called a *poset-monomorphism* iff $f.x \sqsubseteq_{\mathcal{A}} f.y \equiv x \sqsubseteq_{\mathcal{B}} y$ . A function is called a *poset-isomorphism* iff it is a surjective poset-monomorphism.

□

**Theorem 4.9** If $\mathcal{A}$ and $\mathcal{B}$ are posets with $f \in \mathcal{A} \longleftarrow \mathcal{B}$ then

**a** if $f$ is a poset-monomorphism then $f$ is injective,

**b** if $f$ is a poset-isomorphism then $f$ is existentially $\sqcup_{\mathcal{A}\longleftarrow\mathcal{B}}$ junctive and existentially $\sqcap_{\mathcal{A}\longleftarrow\mathcal{B}}$ junctive.

**Proof** Part **a** is easily proven by using anti-symmetry. We prove **b** only. Let $X \subseteq \mathcal{B}$ be such that $\sqcup.X$ exists. We prove $f.\sqcup.X$ solves the defining equation for $\sqcup.f.X$ . For arbitrary $z \in \mathcal{A}$ we derive

$$
\begin{array}{ll}
 & f.\sqcup.X \sqsubseteq z \\
\equiv & \qquad \{ \ \bullet \ f.y = z \text{ since } f \text{ is surjective } \} \\
 & f.\sqcup.X \sqsubseteq f.y \\
\equiv & \qquad \{ \ f \text{ is a poset-monomorphism } \} \\
 & \sqcup.X \sqsubseteq y \\
\equiv & \qquad \{ \ \text{definition of supremum } \} \\
 & \forall(x : \ x \in X : \ x \sqsubseteq y) \\
\equiv & \qquad \{ \ \bullet \ f.y = z \ , \ f \text{ is a poset-monomorphism } \} \\
 & \forall(x : \ x \in X : \ f.x \sqsubseteq z) \qquad .
\end{array}
$$

□

**Exercise 4.10** Show that

$$f \text{ is monotonic} \equiv \ \forall(S : \underline{\mathbf{min}}.S \text{ exists} : \ f.\underline{\mathbf{min}}.S = \underline{\mathbf{min}}.f.S) \quad .$$

□

## 4.3 Composition of Functions

This section is devoted to just one theorem, a trivial theorem that is probably the most frequently used theorem of all that we present. (Because it is used so frequently we tend to take it for granted and rarely cite it explicitly.) Its proof is equally trivial.

(It is worth pausing to remark that the word "trivial" has two meanings: one meaning is "of little importance" and the other "commonplace". We shall often discuss "trivial" matters but by that we do not mean that they are unimportant. Rather the opposite — they are "commonplace", i.e. are used frequently, and hence are very important.)

**Theorem 4.11** Let $(\mathcal{A}, \sqsubseteq)$, $(\mathcal{B}, \sqsubseteq)$ and $(\mathcal{C}, \sqsubseteq)$ be partially-ordered sets. Suppose $f \in \mathcal{A} \longleftarrow \mathcal{B}$ and $g \in \mathcal{B} \longleftarrow \mathcal{C}$. Then $f \bullet g$ enjoys any junctivity or continuity type shared by $f$ and $g$.

**Proof** We may confine ourselves to monotonic $f$ and $g$, this being the weakest continuity type (see theorem 4.6).

Suppose $S$ is a subset of $\mathcal{C}$. Trivially, $g.S$ is a subset of $\mathcal{B}$ with the same or smaller cardinality than that of $S$. Since $g$ is, by assumption, monotonic it is also straightforward to see that $g.S$ is totally ordered if $S$ is totally ordered. Thus, with bound variables $S$ ranging over subsets of $\mathcal{C}$, and $T$ ranging over subsets of $\mathcal{B}$, both having some given junctivity type and being totally ordered in the case that $S$ is totally ordered, we have:

$$
\begin{array}{ll}
 & \forall(S :: \ f.g.\sqcup.S \ = \ \sqcup.f.g.S) \\
\Leftarrow & \qquad \{ \text{ calculus } \} \\
 & \forall(S :: \ f.g.\sqcup.S \ = \ f.\sqcup.g.S) \quad \wedge \quad \forall(S :: \ f.\sqcup.g.S \ = \ \sqcup.f.g.S) \\
\Leftarrow & \qquad \{ \text{ Leibniz's rule applied to the 1st conjunct,} \\
 & \qquad\quad T \ := \ g.S \text{ and predicate calculus to the 2nd (taking note} \\
 & \qquad\quad \text{of the above remarks regarding the type of } T) \ \} \\
 & \forall(S :: \ g.\sqcup.S \ = \ \sqcup.g.S) \quad \wedge \quad \forall(T :: f.\sqcup.T \ = \ \sqcup.f.T) \quad .
\end{array}
$$

□

## 4.4 Pointwise Orderings

In this section we show how to form (complete) lattices of functions. The basic insight is that functions on partially ordered sets can themselves be partially ordered.

**Definition 4.12 (Pointwise Ordering of Functions)** For functions $f$ and $g$ both having type $\mathcal{A} \longleftarrow \mathcal{B}$, where $(\mathcal{A}, \sqsubseteq)$ is a poset, we define

$$f \;\dot{\sqsubseteq}\; g \quad \equiv \quad \forall(x:\; x \in \mathcal{B}:\; f.x \sqsubseteq g.x) \quad .$$

□

In effect we "lift" the ordering on $\mathcal{A}$ to an ordering on functions with range $\mathcal{A}$.

The "point" above the inequality symbol is a reminder that the ordering is "point"wise defined. For pencil and paper calculations it soon becomes irritating to continually write it and you may choose not to do so if you are confident of what you are doing. We will always include the point for greater clarity and because some equations can look decidedly suspect if this type information is not present. (Once or twice we will even have to include two points!)

Together with this lifting, we also lift the structure present for $\mathcal{A}$.

**Theorem 4.13** Let $(\mathcal{A}, \sqsubseteq)$ be a complete lattice and $\mathcal{B}$ an arbitrary set. Then the set of functions of type $\mathcal{A} \longleftarrow \mathcal{B}$ forms a complete lattice under the pointwise ordering of functions. More concisely, $(\mathcal{A} \longleftarrow \mathcal{B}, \dot{\sqsubseteq})$ is a complete lattice.

**Proof** Let $F$ be a subset of $\mathcal{A} \longleftarrow \mathcal{B}$ . Our task is to exhibit a candidate value for the supremum or infimum of $F$. For no particular reason at all we choose to construct a candidate for the supremum of $F$. Then we have to show that the candidate fulfills the specification of the supremum, i.e. the exhibited candidate is a function of type $\mathcal{A} \longleftarrow \mathcal{B}$ and the candidate satisfies the equation

$$h :: \qquad \forall(g :: \; h \;\dot{\sqsubseteq}\; g \quad \equiv \quad \forall(f:\; f \in F:\; f \;\dot{\sqsubseteq}\; g))$$

where $g$ and $h$ are functions of type $\mathcal{A} \longleftarrow \mathcal{B}$ .

As candidate for the supremum we take the function $\overline{F} \in \mathcal{A} \longleftarrow \mathcal{B}$ defined by

$$\overline{F}.x \;=\; \sqcup_{\mathcal{A}}.(f:\; f \in F:\; f.x) \quad ,$$

there being no other reasonable choice. (The dual of the candidate for the supremum, i.e. $\underline{F}.x \;=\; \sqcap_{\mathcal{A}}.(f:\; f \in F:\; f.x)$ would be an adequate candidate for the infimum in $\mathcal{A} \longleftarrow \mathcal{B}$.) Now let us verify that $\overline{F}$ meets the specification of the supremum. Assume $g$ is a function in $\mathcal{A} \longleftarrow \mathcal{B}$. Then

$$
\begin{array}{ll}
 & \forall(f:\ f\in F:\ f \mathrel{\dot{\sqsubseteq}} g) \\
\equiv & \qquad \{\ \text{definition 4.12}\ \} \\
 & \forall(f:\ f\in F:\ \forall(x::\ f.x \sqsubseteq g.x)) \\
\equiv & \qquad \{\ \text{dummy interchange}\ \} \\
 & \forall(x::\ \forall(f:\ f\in F:\ f.x \sqsubseteq g.x)) \\
\equiv & \qquad \{\ \text{specification of supremum}\ \} \\
 & \forall(x::\ \sqcup.(f:\ f\in F:\ f.x) \sqsubseteq g.x) \\
\equiv & \qquad \{\ \text{definition of } \overline{F}\ \} \\
 & \forall(x::\ \overline{F}.x \sqsubseteq g.x) \\
\equiv & \qquad \{\ \text{definition: 4.12}\ \} \\
 & \overline{F} \mathrel{\dot{\sqsubseteq}} g \quad .
\end{array}
$$

□

In theorem 4.13 the set $\mathcal{B}$ is arbitrary. If we assume that it too is a complete lattice then each junctivity or continuity type identifies a complete sublattice of the lattice of functions of type $\mathcal{A} \longleftarrow \mathcal{B}$. This follows from the following simple argument.

Let $(\mathcal{A}, \sqsubseteq)$ and $(\mathcal{B}, \sqsubseteq)$ be complete lattices. Let $S$ be a subset of $\mathcal{B}$ and let $F$ be a subset of $\mathcal{A} \longleftarrow \mathcal{B}$. Define $\overline{F}$ as before by

$$\overline{F}.x \ = \ \sqcup_{\mathcal{A}}.(f:\ f\in F:\ f.x) \quad .$$

Then,

$$
\begin{array}{ll}
 & \overline{F}.\sqcup.S \quad = \quad \sqcup.\overline{F}.S \\
\equiv & \qquad \{\ \text{definition of } \overline{F}\ \} \\
 & \sqcup.(f:\ f\in F:\ f.\sqcup.(x:\ x\in S:\ x)) \\
 & \ = \quad \sqcup.(x:\ x\in S:\ \sqcup.(f:\ f\in F:\ f.x)) \\
\equiv & \qquad \{\ \text{dummy interchange}\ \} \\
 & \sqcup.(f:\ f\in F:\ f.\sqcup.(x:\ x\in S:\ x)) \\
 & \ = \quad \sqcup.(f:\ f\in F:\ \sqcup.(x:\ x\in S:\ f.x)) \\
\Leftarrow & \qquad \{\ \text{Leibniz}\ \} \\
 & \forall(f:\ f\in F:\ f.\sqcup.(x:\ x\in S:\ x) \quad = \quad \sqcup.(x:\ x\in S:\ f.x)) \\
\equiv & \qquad \{\ \text{definition}\ \} \\
 & \forall(f:\ f\in F:\ f.\sqcup.S \quad = \quad \sqcup.f.S) \quad .
\end{array}
$$

We conclude the following:

**Theorem 4.14** Let $(\mathcal{A}, \sqsubseteq)$ and $(\mathcal{B}, \sqsubseteq)$ be complete lattices. Let $T$ be some junctivity or continuity type, and let $\mathcal{F}$ be the subset of $\mathcal{A} \longleftarrow \mathcal{B}$ consisting of all functions of type $T$. Then $\mathcal{F}$ forms a complete lattice under the pointwise ordering of functions.

**Proof** Junctivity and continuity types come in two versions: the $\sqcup$-forms and $\sqcap$-forms.

Let $T$ be a $\sqcup$-junctive or $\sqcup$-continuity type. We know from theorem 4.13 that the set of *all* functions in $\mathcal{A} \longleftarrow \mathcal{B}$ forms a complete lattice under the pointwise ordering. By (3.50) it thus suffices to show that the supremum (in the latter lattice) of any subset $F$ of $\mathcal{F}$ is itself an element of $\mathcal{F}$. But this is evident from the above calculation. Just quantify over all $S$ having the given type $T$.

For $T$ a $\sqcap$-junctive or $\sqcap$-continuity type, a dual reasoning can be given based on the obvious candidate for the pointwise infima, the $\underline{F}$ in the proof of theorem 4.13

□

**Exercise 4.15** Let $(\mathcal{A}, \sqsubseteq)$ be a lattice and $\mathcal{B}$ be an arbitrary set. Suppose $f$ and $g$ are both functions of type $\mathcal{A} \longleftarrow \mathcal{B}$. Show that for all $X \subseteq \mathcal{B}$

$$\sqcap.f.X \ \dot{\sqsubseteq}\ \sqcap.g.X \quad \Leftarrow \quad f \ \dot{\sqsubseteq}\ g \quad .$$

State the dual property.

□

## 4.5 Sectioned Compositions

Earlier we briefly touched on function compositions. Here we explore their properties with respect to the pointwise ordering of functions.

We first observe that composition is monotonic with respect to its left argument.

(4.16) $f \bullet h \ \dot{\sqsubseteq}\ g \bullet h \quad \Leftarrow \quad f \ \dot{\sqsubseteq}\ g \quad .$

**Proof**

$$
\begin{array}{ll}
 & f \bullet h \ \dot{\sqsubseteq}\ g \bullet h \\
\equiv & \qquad \{ \text{ definition 4.12, definition } \bullet \ \} \\
 & \forall(x ::\ f.h.x \sqsubseteq g.h.x) \\
\Leftarrow & \qquad \{ \text{ predicate calculus, } y := h.x \ \} \\
 & \forall(y ::\ f.y \sqsubseteq g.y) \\
\equiv & \qquad \{ \text{ definition 4.12 } \} \\
 & f \ \dot{\sqsubseteq}\ g \quad .
\end{array}
$$

□

For the right argument, we have

(4.17) $\forall(f,g :: h \bullet f \ \dot{\sqsubseteq}\ h \bullet g \quad \Leftarrow \quad f \ \dot{\sqsubseteq}\ g) \quad \equiv \quad h$ is monotonic .

**Proof**

($\Rightarrow$) Instantiate $f, g$ to the constant functions $\hat{x}, \hat{y}$.

($\Leftarrow$)

$$
\begin{array}{ll}
 & h \bullet f \ \ \dot{\sqsubseteq}\ \ h \bullet g \\
\equiv & \qquad \{ \text{ definition 4.12 } \} \\
 & \forall(x ::\ h.f.x \sqsubseteq h.g.x) \\
\Leftarrow & \qquad \{ \ \bullet \quad h \text{ is monotonic } \} \\
 & \forall(y ::\ f.y \sqsubseteq g.y) \\
\equiv & \qquad \{ \text{ definition 4.12 } \} \\
 & f \ \dot{\sqsubseteq}\ g \ .
\end{array}
$$

□

Properties (4.16) and (4.17) exhibit an asymmetry in the left and the right arguments of function composition. For arbitrary $f$, let $(\bullet f)$ be defined as

$$(\bullet f).g \ = \ g \bullet f \quad .$$

So $(\bullet f)$ maps a function to a function. A similar definition can be made for $(f\bullet)$. In this setting (4.16) expresses the monotonicity of $(\bullet h)$ for *arbitrary* $h$, while (4.17) says that $(h\bullet)$ is monotonic iff $h$ is *monotonic*.

The definitions of $(\bullet f)$ and $(f\bullet)$ employ a device called *sectioning*. In general, given any binary function one can fix one of its arguments to some constant value to obtain a unary function. The device can also be employed on non-binary functions (thus unary functions, ternary functions, quaternary functions etc.) to obtain functions of lower arity. (Fixing the only argument of a unary function to some constant gives a constant.)

The function $(\bullet f)$ has one other property that will be useful later on.

**Theorem 4.18** If $(\mathcal{A}, \sqsubseteq)$ is a complete lattice and $f$ an endofunction on $\mathcal{A}$, then $(\bullet f)$ is universally $\dot{\sqcup}$-junctive.

**Proof** For $G \subseteq \mathcal{A} \longleftarrow \mathcal{A}$ and any $x \in \mathcal{A}$ we derive

$$
\begin{array}{ll}
& (\dot{\sqcup}((\bullet f).G)).x \\
= & \quad \{ \text{ definition } (\bullet f) \ \} \\
& (\dot{\sqcup}(G \bullet f)).x \\
= & \quad \{ \text{ definition } \dot{\sqcup} \ \} \\
& \sqcup(h : \ h \in G \bullet f : \ h.x) \\
= & \quad \{ \text{ dummy change, definition } \bullet \ \} \\
& \sqcup(g : \ g \in G : \ g.f.x) \\
= & \quad \{ \text{ definition of } \dot{\sqcup} \ \} \\
& (\dot{\sqcup}G).f.x \\
= & \quad \{ \text{ calculus, definition of } (\bullet f) \ \} \\
& ((\bullet f).(\dot{\sqcup}G)).x \quad .
\end{array}
$$

□

The function $(f\bullet)$ is in general not universally $\dot{\sqcup}$-junctive, since in general it is not even monotonic (see (4.17)).

**Exercise 4.19** Let $(\mathcal{A}, \sqsubseteq)$ be a complete lattice. Prove that for $f$ an endofunction on $\mathcal{A}$:

$$f \text{ is universally } \sqcup-\text{junctive} \quad \equiv \quad (f\bullet) \text{ is universally } \dot{\sqcup}-\text{junctive} \quad .$$

□

# Chapter 5

# Galois Connections

## 5.1 Introduction

This chapter forms the climax to our skirmish with lattice theory. In it we define and explore the notion of a Galois connection between two functions. In later chapters we apply the acquired knowledge in unfamiliar ways to familiar areas of lattice theory. For instance, in one of the following chapter we derive the well-known Knaster-Tarski theorem on fixed points as a simple corollary of a property of Galois connections.

Although such applications of Galois connections are unfamiliar, the notion itself has a very long history beginning, one might argue, with the mathematicians of ancient Greece. The ancient Greeks were concerned with constructibility problems such as: using ruler and compass alone, is it possible to trisect an angle or, construct a regular polygon with $n$ sides for given $n > 2$? Another related constructibility problem is that of solving a polynomial equation using only "radicals" — rational operations and the extraction of roots. The solutions to the general quadratic equation $ax^2 + bx + c = 0$, nowadays a compulsory element of secondary school mathematics, were known to the Babylonian scholars around 900 A.D. The Italian mathematicians Scipio Ferro and Niccolo Fontana (nicknamed Tartaglia because he stammered) solved the general cubic equation, their results being unscrupulously plagiarised by Girolamo Cardan in his *Ars Magna* published in 1545. Ferrari, a pupil of Cardan, was the first to solve the general quartic equation, but almost 300 years were to pass before Abel demonstrated in 1828 the *unsolvability* of the general quintic equation.

Évariste Galois, who died in 1832 at the age of 21 and whose work was published fourteen years after his death, established necessary and sufficient conditions for a polynomial equation to be solvable by radicals. He did so by relating every extension of a field to a group and then studying the properties of field extensions by studying the properties of the related groups. (In the case of solving polynomial equations using radicals the field is formed by the rationals and the extensions are formed by adding the extraction of roots.) The revolutionary methods Galois introduced led to the development of the modern theories of groups and fields, the relationship he established between field extensions and groups now being known as *the Galois correspondence*.

We shall have nothing further to say about *the* Galois correspondence, our attention being devoted to connections between functions in a much broader setting. (Those wishing to know more about the Galois correspondence are referred to [41] — from which together with [90] the above history has been culled.) The essential idea to be retained from this discussion, however, and the reason that Galois connections bear the illustrious mathematician's name, is to explore the properties of one function by relating it to a second and then exploring that function's properties.

Now, what is a Galois connection? Let's present a first definition. A Galois connection involves two ordered sets $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$, a function $F \in \mathcal{A} \longleftarrow \mathcal{B}$ and a function in the opposite direction $G \in \mathcal{B} \longleftarrow \mathcal{A}$. We will say $(F, G)$ *is a Galois connection* iff for all $x \in \mathcal{B}$ and $y \in \mathcal{A}$ the following holds

$$F.x \ \sqsubseteq_{\mathcal{A}} \ y \ \equiv \ x \ \sqsubseteq_{\mathcal{B}} \ G.y \ .$$

This compact definition of a Galois connection was first introduced in [85]. We refer to $F$ as the *lower adjoint* and to $G$ as the *upper adjoint*.

As might be anticipated from the names given to $F$ and $G$, Galois connections are related to the categorical notion of an adjunction[1]. When considering a set with an order as a category, Galois connections and adjunctions coincide. So we can study Galois connections by studying adjunctions. Adjunctions have been extensively studied, one of the most comprehensive accounts of adjunctions being [59], so why bother to study Galois connections separately? There are several reasons why Galois connections are interesting in their own right.

The notion of an adjunction is an order of magnitude more complex than the notion of a Galois connection. At best an adjunction involves two categories,

[1] All remarks referring to category theory can be skipped if you're not familiar with it.

two functors and two functions between the hom-sets. Other definitions require two natural transformations instead of the functions between hom-sets, or one natural transformation and a universality property. To call a Galois connection an adjunction is just mathematical overkill!

Galois connections have excellent calculational properties due in no small measure to the simplicity and elegance of the definition. Calculations with adjunctions are much harder. Moreover, there are properties of Galois connections that are not valid for adjunctions in general.

Conditions a function has to satisfy in order to ensure existence of an (upper or lower) adjoint are easily stated for Galois connections. So one can readily specify a function by stating that it is the Galois adjoint of a known function and derive properties of the specified function without the need to give a closed formula. This approach can be very fruitful since often the closed formula turns out to be complicated or clumsy to work with.

This is not to say that the categorical notion of an adjunction does not have its place. The class of functions that can be defined by the categorical notion is much broader and includes many functions of daily use in computing science that cannot be defined via a Galois connection. Nevertheless the class of functions that can be so defined is sufficiently large to be interesting and worthy of study. Moreover, it is our view that a proper understanding of the categorical notion of adjunction is best gained by viewing it as a "constructive version" of the notion of a Galois connection. (We will explain this assertion in more detail later.) An indispensable prerequisite for a study of the categorical notion of adjunction is thus a thorough understanding of the notion of a Galois connection.

## 5.2 Elementary Examples

Galois connections occur in various parts of mathematics and computing science, but they are not often recognised as such. Even where the existence of a Galois connection is recognised that fact is rarely exploited. As a consequence, proofs we have encountered are either complicated or unnecessarily long, whereas exploitation of the Galois connection can immediately suggest compact and straightforward proofs. In this section we give some elementary examples of Galois connections and their use in constructing elegant calculations.

The examples in this section give a first impression of how to calculate with

Galois connections. They also give a first hint as to what properties are common to Galois connections. They have been chosen for the appeal of familiarity and are not used further in the text. In later sections we give an overview of the properties of Galois connections. We make no claim to originality but we do try to establish each property in a convincing, calculational style.

### 5.2.1 Floor and ceiling

Our first example of a Galois connection is in the realm of number theory. In most mathematical texts the function *floor* from reals to integers is defined as follows: for all real $x$ we take $\lfloor x \rfloor$ to be the greatest integer at most $x$. Likewise the *ceiling*, denoted $\lceil x \rceil$, is defined for all real $x$ as the least integer at least $x$.

With these definitions various properties of the two functions can be verified, but it is difficult to actually calculate with them. A possible way to improve this is to give the definition as a Galois connection. Let's first consider the floor-function.

$\lfloor x \rfloor$ is defined as the greatest integer satisfying some property. To be precise, it has the property that it is at most $x$. Hence if we have another integer $n$ that satisfies the same property —i.e. $n \leq x$—, $n$ cannot be greater than $\lfloor x \rfloor$, since $\lfloor x \rfloor$ is defined to be the greatest such integer. This gives the following Galois connection as definition:

**Definition 5.1** For all real $x$, $\lfloor x \rfloor$ is an integer such that for all integers $n$

$$n \leq \lfloor x \rfloor \;\equiv\; n \leq x \;\; .$$

□

In a similar way we find a Galois connection for the ceiling-function.

**Definition 5.2** For all real $x$, $\lceil x \rceil$ is an integer such that for all integers $n$

$$\lceil x \rceil \leq n \;\equiv\; x \leq n \;\; .$$

□

One might complain that definitions 5.1 and 5.2 are not genuine Galois connections, since they involve only one function, namely the floor, respectively ceiling, function from reals to integers. A Galois connection should involve two functions in opposite directions. But in both specifications there is a second, invisible, function present that maps integers to reals. In this case that is a very trivial function, since integers can be embedded in a straightforward way in the reals. If we take $U$ to be that embedding, we can reformulate definition 5.1 as

$$(5.3) \qquad n \leq \lfloor x \rfloor \ \equiv \ U.n \leq x \ .$$

A similar rewriting can be done for definition 5.2. As long as we note that $n$ is an integer, we can safely omit the $U$.

Embedding functions offer a good example of "trivial" meaning "commonplace" rather than "of little importance". Some of the central results to follow are obtained by observing that an embedding function or some equally "trivial" function has an adjoint.

The specifications of floor and ceiling have been given in the shape of a Galois connection elsewhere, for example in [44] and [82], but that shape is not used in any calculation. Even worse, in [44] the authors don't consider it useful at all to recognise a Galois connection since they have difficulty remembering it! In order to show the usefulness of the given Galois connection, let's calculate some properties.

A complaint that might, with some justification, be made about 5.1 and 5.2 is that it is not immediately evident that, viewed as equations in $\lfloor x \rfloor$ and $\lceil x \rceil$, respectively, they do indeed have solutions. To see that this is so we will conduct a small calculation.

Replacing $\lceil x \rceil$ in (5.2) by the dummy $m$ (ranging over integers) we wish to show that the equation

$$(5.4) \qquad m :: \qquad \forall(n :: \ m \leq n \ \ \equiv \ \ x \leq n)$$

has exactly one solution. (Since $m$ is an integer, one can not use indirect equality to conclude $x = m$ from (5.4).)

It is evident that it has at most one solution — since the right side of the equivalence in (5.4) does not depend on $m$ — so it suffices to show that it has at least one solution. We do this by eliminating the universal quantification as follows:

$$\begin{array}{ll}
 & \forall(n :: \ m \leq n \equiv x \leq n) \\
\equiv & \qquad \{ \text{ predicate calculus } \} \\
 & \forall(n :: \ m \leq n \Rightarrow x \leq n) \quad \wedge \quad \forall(n :: \ m \leq n \Leftarrow x \leq n) \\
\Leftarrow & \qquad \{ \text{ transitivity of at-most, integer arithmetic } \} \\
 & x \leq m \quad \wedge \quad \forall(n :: \ m{-}1 \ < \ n \ \Leftarrow \ x \leq n)
\end{array}$$

$$
\begin{array}{ll}
\Leftarrow & \{ \text{ arithmetic } \} \\
& x \le m \quad \wedge \quad m-1 < x \\
\equiv & \{ \text{ arithmetic } \} \\
& x \le m < x+1 \quad .
\end{array}
$$

Since there is exactly one integer that is at least $x$ and smaller than $x+1$ this shows that $\lceil x \rceil$ is well defined.

For our second calculation we shall establish the following property mentioned in [44].

$$
(5.5) \qquad \left\lfloor \sqrt{\lfloor x \rfloor} \right\rfloor \;=\; \left\lfloor \sqrt{x} \right\rfloor
$$

for all $x$, $0 \le x$.

**Proof** For any integer $n$ we derive

$$
\begin{array}{ll}
& n \;\le\; \left\lfloor \sqrt{\lfloor x \rfloor} \right\rfloor \\
\equiv & \{ \; n \text{ is an integer, definition 5.1 } \} \\
& n \;\le\; \sqrt{\lfloor x \rfloor} \\
\equiv & \{ \text{ arithmetic } \} \\
& n^2 \;\le\; \lfloor x \rfloor \quad \vee \quad n \;<\; 0 \\
\equiv & \{ \; n^2 \text{ is an integer, definition 5.1 } \} \\
& n^2 \;\le\; x \quad \vee \quad n \;<\; 0 \\
\equiv & \{ \text{ arithmetic } \} \\
& n \;\le\; \sqrt{x} \\
\equiv & \{ \; n \text{ is an integer, definition 5.1 } \} \\
& n \;\le\; \lfloor \sqrt{x} \rfloor \quad .
\end{array}
$$

The property now follows by the rule of indirect equality

□

Note that the decision on how to prove the theorem, i.e. the introduction of the integer $n$, is entirely inspired by the shape of definition 5.1. The only way we can calculate something about the floor-function is to use its specification. That specification allows one to rewrite the floor-function only when it is in some special shape. In this case: it is on the *greater* side of the $\le$ and on the *smaller* side there is an integer. So the only way one can hope to be able to prove something from its specification is to manipulate the demonstrandum in such a way that the specification can be used.

You are cordially invited to compare the above proof of (5.5) with the proof given in [44]. You may also wish to prove $\lfloor x+m \rfloor = \lfloor x \rfloor + m$, and $\lfloor x \rfloor \ \underline{\mathbf{min}}\ \lfloor y \rfloor = \lfloor x\, \underline{\mathbf{min}}\, y \rfloor$ for all real $x$ and $y$, and integer $m$, or $\lfloor x/m \rfloor = \lfloor \lfloor x \rfloor /m \rfloor$ for all real $x$ and positive integer $m$ in the same calculational style. You should observe a pattern and be able to formulate it as a general theorem.

### 5.2.2 Sums and Differentials

If two functions are inverses of each other then they are Galois connected. Suppose the inverse functions are $\theta$ and $\phi$. Then we have, for all $x$ in the domain of $\theta$, and $y$ in the domain of $\phi$,

$$(5.6) \qquad \theta.x = y \quad \equiv \quad x = \phi.y \quad .$$

Just like giving Galois connections as examples of adjunctions it would normally be pure overkill to give inverse functions as examples of Galois connections! The two poset orderings needed to establish the connection are the trivial orderings whereby the only ordered elements are equal elements, and little can be gained by instantiating general theorems about Galois connections that is not predicted by much simpler, direct calculations using the fact that a composition of the one function followed by the other is an identity function. The main benefit that is gained from the observation is that it can suggest properties that one might investigate of Galois-connected functions. For example, inverse functions have "inverse" algebraic properties. The exponential function, for instance, has as its inverse the logarithmic function, and

$$\exp(-x) = \frac{1}{\exp x} \qquad \text{and} \qquad \exp(x+y) = \exp x \cdot \exp y$$

whereas

$$-\ln x = \ln(\frac{1}{x}) \qquad \text{and} \qquad \ln x + \ln y = \ln(x \cdot y) \quad .$$

In general, if $\theta$ and $\phi$ are inverse functions then, for any functions $f$ and $g$ of appropriate type,

$$\forall(x :: \ \theta.f.x = g.\theta.x) \quad \equiv \quad \forall(y :: \ f.\phi.y = \phi.g.y) \quad .$$

More generally, and expressed at function level, if $(\theta_0, \phi_0)$ and $(\theta_1, \phi_1)$ are pairs of inverse functions, then for all functions $f$ and $g$ of appropriate type,

$$(5.7) \qquad \theta_0 \bullet f = g \bullet \theta_1 \quad \equiv \quad f \bullet \phi_1 = \phi_0 \bullet g \quad .$$

(You are invited to discover instances of this theorem. A suggested starting point is the identity $sin^2x = 1 - cos^2x$.) Knowing this, one is encouraged to investigate whether Galois-connected functions have similar "inverse" algebraic properties, but one would be foolhardy to believe that any investigation of Galois connections would uncover new facts about inverse functions. Nevertheless, the characterisation (5.6) of inversality can sometimes be useful. In this section we consider an example to do with summing polynomial functions. Like the example of the ceiling function this application was suggested to us by reading the book Concrete Mathematics by Graham, Knuth and Patashnik [44], in particular the section on "finite calculus".

Let $f$ and $g$ denote functions from naturals to reals. Assume that $f.0 = 0$. Define the operators $\Delta$ and $\Sigma$ by

$$\begin{array}{rcl}(\Delta f).x & = & f.(x+1) - f.x \\ (\Sigma g).x & = & \Sigma(y:\ 0 \leq y < x : g.y)\end{array}$$

for all numbers $x$. Then we have the Galois connection:

$$(5.8) \quad f = \Sigma g \quad \equiv \quad \Delta f = g \quad .$$

The proof of this identity involves very elementary quantifier calculus and is therefore omitted.

Let us suppose our goal is to develop a body of rules that enable one to find efficient ways of evaluating finite sums $\Sigma g$ for given function $g$. This goal may be approached by tackling the easier problem of developing a body of rules to compute differentials $\Delta f$ and then using the Galois connection (5.8) to convert the rules to rules about $\Sigma$.

To illustrate this idea let us restrict $g$ to the class of polynomial functions. Our goal is thus to develop a little theory that will enable us to compute finite sums of polynomials such as $\Sigma(y:\ 0 \leq y < x:\ y^2 + 3y + 1)$.

We begin our theory development by exploring the *differentials* of polynomials. Since a polynomial function of $x$ is either a constant function, the identity function, the sum of two polynomial functions or the product of two polynomial functions, table 5.1 suffices to rewrite $(\Delta f).x$ as a polynomial in $x$ for any given polynomial $f.x$ satisfying the assumption $f.0 = 0$. (In the table $c$ denotes an arbitrary constant. Verification of all four statements is straightforward.) We observe that a table of differentials in the finite calculus looks like a table of differentials in the infinite calculus. In particular taking derivatives reduces the degree of a polynomial by exactly one.

| $f.x$ | $(\Delta f).x$ |
|---|---|
| $0$ | $0$ |
| $cx$ | $c$ |
| $f.x + g.x$ | $(\Delta f).x + (\Delta g).x$ |
| $f.x \times g.x$ | $f.x \times (\Delta g).x + (\Delta f).x \times g.(x+1)$ |

Table 5.1: Table of Differentials

Ideally we would now like to construct a similar table for $\Sigma$. Four entries would be required, one for constants, one for the identity function, one for a sum and one for a product of two polynomials. The unfortunate occurrence of "+1" in the $\Delta$ entry for products frustrates this particular goal but nevertheless an algorithm for expressing the sum of a polynomial function as a polynomial function can be derived that exploits the above table of differentials. We illustrate the algorithm by considering the $\Sigma$ entry for the identity function.

Since taking derivatives reduces the degree of a polynomial by one we conjecture that the sum of the identity function is a quadratic polynomial. The coefficients of that polynomial are calculated as follows:

$$
\begin{array}{ll}
\multicolumn{2}{l}{\text{By construction of } a \text{ and } b\text{:}} \\
 & \forall(x ::\ ax + bx^2 \ = \ \Sigma(y :\ 0 \le y < x :\ y)) \\
\equiv & \qquad \{ \text{ Galois connection: (5.8) with } g \text{ the identity } \} \\
 & \forall(y ::\ \Delta(x \mapsto ax + bx^2).y \ = \ y) \\
\equiv & \qquad \{ \text{ differential calculus: table 5.1 } \} \\
 & \forall(y ::\ a + by + b(y+1) \ = \ y) \\
\equiv & \qquad \{ \text{ arithmetic } \} \\
 & a + b \ = \ 0 \ \ \wedge \ \ 2b \ = \ 1 \\
\equiv & \qquad \{ \text{ arithmetic } \} \\
 & a \ = \ -\tfrac{1}{2} \ \ \wedge \ \ b \ = \ \tfrac{1}{2} \quad .
\end{array}
$$

We have thus established the identity

$$\Sigma(y :\ 0 \le y < x :\ y) \quad = \quad -\tfrac{1}{2}x + \tfrac{1}{2}x^2 \quad .$$

Extrapolating from this four step calculation one can easily see that it embodies an algorithm to express $\Sigma g$ as a polynomial function for any given polynomial function $g$. The steps in the algorithm are: postulate that $\Sigma g$ is a polynomial function $f$ with degree one higher than $g$. Compute (symbolically) the coefficients of $\Delta f$ using the table of differentials. Equate the expressions obtained for the coefficients of $f$ to the corresponding given coefficients of $g$. In this way one obtains a system of simultaneous equations which is then solved to obtain the coefficients of $f$. Try it out for yourself on the squaring function.

The point of this little example is to show how one can predict the behaviour of a relatively complicated operator — in this case $\Sigma$ — by studying the behaviour of its inverse — in this case $\Delta$.

### 5.2.3 A short bibliography

We will see several additional examples of Galois connections later in the text but for the moment the ones we have given will have to suffice. It is time to take a more formal approach.

The theory to be presented is not new. Just like the proverbial wheel the notion of a Galois connection has been discovered and rediscovered in various fields, it has a variety of guises and is known under a variety of different names. One of the earliest theoretical contributions (that we are aware of) was made by G. Birkhoff with the introduction of so-called "polarities" [24]. C.J. Everett subsequently proved that every Galois connection between powersets arises from a polarity [40]. The actual generalisation to the Galois connections as we use them here was done by O. Ore [78]. J. Schmidt introduced a concise formula for describing a Galois connection [85], that formula being the one used here as the definition of a Galois connection in preference to the one proposed by Ore.

The importance of the notion was recognised at a very early stage in mathematically-oriented computing science literature. As long ago as 1964 Hartmanis and Stearns [47] developed an alternative, but entirely equivalent, formulation of Galois connections called "pair algebras" which they applied to a data-refinement problem – the state assignment problem in sequential machines. (Although they did not use the term in the original paper describing their theory Hartmanis and Stearns briefly acknowledge the relevance of Galois connections in a footnote in their textbook [48] in which they said: "For related mathematical concepts see the discussion of Galois connections between partially ordered sets in [23]." Simons [87] formally establishes the equivalence between Galois connections and pair algebras.) Seven years later, Conway [30] published a book on finite-state machines in which a very important (but sadly almost totally ignored) element was the chapter on so-called "factor theory" and its subsequent application to the construction and analysis of so-called "biregulators". Conway did not refer to the work of Hartmanis and Stearns, nor to Galois connections, but there are clearly recognisable, formally establishable, parallels between his "L-R factorisations" of a regular language and Hartmanis and Stearns' "m-M decompositions" of a finite-state machine.

More recent references to computing science applications of Galois connections are [49, 71, 67]. In [49] there are four kinds of "Galois connections" introduced, ranging from polarities to a restricted form of adjunctions. A comprehensive overview of the theory of Galois connections can be found in [42]. At the end of this chapter we review some of the earlier applications of Galois connections and some applications that may appeal to computing scientists.

## 5.3 Abstract properties

In what follows we take $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ to be partially-ordered sets. We let $F$ be a function to $\mathcal{A}$ from $\mathcal{B}$ and $G$ a function in the opposite direction, so $F \in \mathcal{A} \longleftarrow \mathcal{B}$ and $G \in \mathcal{B} \longleftarrow \mathcal{A}$. For such an $F$ and $G$ we recall the following definition.

**Definition 5.9 (Galois Connection)** $(F, G)$ is a Galois connection iff for all $x \in \mathcal{B}$ and $y \in \mathcal{A}$

$$F.x \ \sqsubseteq_{\mathcal{A}} \ y \ \equiv \ x \ \sqsubseteq_{\mathcal{B}} \ G.y \ .$$

$\Box$

In order to make the formulae more readable, we will drop the subscripts from the orderings. This will not lead to confusion, since it can always be deduced which ordering is meant from type considerations. On occasion, when expressing the junctivity type of a function, we will tag the supremum and infimum operator with the typing of the involved function, in order to assist the reader in keeping the type deduction process manageble. Hence when we call $F$ universally $\sqcup_{\mathcal{A} \leftarrow \mathcal{B}}$ junctive, this means that $F$ preserves all suprema form $\mathcal{B}$ to $\mathcal{A}$ .

Recall also that $F$ is referred to as the lower adjoint, since it is on the *lower* side of an ordering, and $G$ as the upper adjoint, since it is on the *upper* side of an ordering.

In category theory the names left and right adjoint are more common, but we find it difficult to remember which is which, and often mix them up. On the other hand the names lower and upper adjoint are also easily mixed up, since the lower adjoint is the upper adjoint in the dual ordering! Formally:

**Theorem 5.10** $(F, G)$ is a Galois connection iff $(G, F)$ is a Galois connection, where the orderings of $\mathcal{A}$ and $\mathcal{B}$ are reversed.

**Proof** We have for any $x \in \mathcal{A}$ and $y \in \mathcal{B}$:

$$\begin{array}{ll}
 & G.y \sqsupseteq x \\
\equiv & \qquad \{ \text{ dual order } \} \\
 & x \sqsubseteq G.y \\
\equiv & \qquad \{ \ (F, G) \text{ is a Galois connection } \} \\
 & F.x \sqsubseteq y \\
\equiv & \qquad \{ \text{ dual order } \} \\
 & y \sqsupseteq F.x \quad .
\end{array}$$

□

A result of this is that all statements about one of the adjoints of a Galois connection have a dual statement for the other adjoint. That is, any theorem concerning a lower adjoint gives rise to a theorem about the upper adjoint, since that one is the lower adjoint when we reverse the ordering. So with one proof, we get two theorems. In general we state the dual of a theorem, but we don't prove it.

An overview of the following subsections is as follows. We will first derive some so-called "cancellation laws". These are simple calculational rules that enable one to "cancel" (i.e. eliminate), or vice-versa introduce, the functions in a Galois connection under certain circumstances. Next we formulate a number of equivalent definitions of a Galois connection. Knowing that a concept can be defined in several different ways is an indicator of its importance as well as helping one to recognise it in other applications. Then we consider the uniqueness of adjoints, and necessary and sufficient conditions for their existence. Typically the existence conditions are hedged with assumptions about the existence of infima and/or suprema so in the final subsection we consider the properties of Galois connections given that the posets in question form complete lattices.

### 5.3.1 Cancellation laws

In this section we consider some direct and elementary consequences of the definition of a Galois connection. Apart from the defining equation, the first theorem is probably the law that is most frequently used when calculating with Galois connections, as will be seen throughout this chapter. Thus, although we refer to most of the theorems in this section collectively as "cancellation laws", this one is "the" cancellation law.

**Theorem 5.11** **(cancellation)** If $(F, G)$ is a Galois connection, then we have

**a** $x \sqsubseteq G.F.x$ for all $x \in \mathcal{B}$,

**b** $F.G.y \sqsubseteq y$ for all $y \in \mathcal{A}$.

**Proof** Since **a** and **b** are dual, only **a** is proven.

$$
\begin{array}{ll}
 & x \sqsubseteq G.F.x \\
\equiv & \qquad \{ \ (F, G) \text{ is a Galois connection } \} \\
 & F.x \sqsubseteq F.x \\
\equiv & \qquad \{ \ \text{reflexivity } \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

With this theorem it is straightforward to prove the following:

**Corollary 5.12** If $(F, G)$ is a Galois connection, then both $F$ and $G$ are monotonic.

**Proof** For monotonicity of $F$ we observe

$$
\begin{array}{ll}
 & F.x \sqsubseteq F.z \\
\equiv & \qquad \{ \ (F, G) \text{ is a Galois connection } \} \\
 & x \sqsubseteq G.F.z \\
\Leftarrow & \qquad \{ \ \text{cancellation, transitivity } \} \\
 & x \sqsubseteq z \quad .
\end{array}
$$

Monotonicity of $G$ follows by duality.

□

What is particularly attractive about the form of the definition of a Galois connection is that it expresses an equivalence between two predicates. Sometimes in calculations, however, its form is inappropriate, preventing its being used directly. For greater flexibility one would like to have equivalences between a broader class of expressions. That is the content of the next few theorems.

**Theorem 5.13** If $(F, G)$ is a Galois connection then the following are equivalent:

**a** $x \sqsubseteq G.y$ ,

**b** $F.x \sqsubseteq F.G.y$ ,

**c** $F.x \sqsubseteq y$ ,

**d** $G.F.x \sqsubseteq G.y$ .

**Proof** The proof is by cyclic implication.

$$\begin{array}{ll}
 & x \sqsubseteq G.y \\
\Rightarrow & \quad \{ \ F \text{ is monotonic } \} \\
 & F.x \sqsubseteq F.G.y \\
\Rightarrow & \quad \{ \ \text{cancellation, transitivity } \} \\
 & F.x \sqsubseteq y \\
\Rightarrow & \quad \{ \ G \text{ is monotonic } \} \\
 & G.F.x \sqsubseteq G.y \\
\Rightarrow & \quad \{ \ \text{cancellation, transitivity } \} \\
 & x \sqsubseteq G.y \quad .
\end{array}$$

□

Observe from the proof of theorem 5.13 that the cancellation laws and monotonicity suffice to prove the existence of a Galois connection.

By instantiating $x := G.x$ in theorem 5.13, and abandoning part **d**, we obtain:

**Corollary 5.14** If $(F, G)$ is a Galois connection then the following are equivalent:

**a** $G.x \sqsubseteq G.y$ ,

**b** $F.G.x \sqsubseteq F.G.y$ ,

**c** $F.G.x \sqsubseteq y$ .

□

Dualising corollary 5.14 leads to:

**Corollary 5.15** If $(F, G)$ is a Galois connection then the following are equivalent:

**a** $F.x \sqsubseteq F.y$ ,

**b** $G.F.x \sqsubseteq G.F.y$ ,

**c** $x \sqsubseteq G.F.y$ .

□

These last two corollaries tell us something about the two adjoints when they are restricted to $F.\mathcal{B}$ and $G.\mathcal{A}$ . In particular:

**Theorem 5.16** If $(F, G)$ is a Galois connection then

**a** $F \in \mathcal{A} \longleftarrow G.\mathcal{A}$ is a poset-monomorphism,

**b** $G \in \mathcal{B} \longleftarrow F.\mathcal{B}$ is a poset-monomorphism.

**Proof** Assume $(F, G)$ is a Galois connection. For part **a** we have to prove that $F.u \sqsubseteq F.v \;\equiv\; u \sqsubseteq v$ for all $u, v \in G.\mathcal{A}$ . This follows directly from the equivalence of **b** and **a** in corollary 5.14.

Part **b** is the dual of part **a**.

□

We shall shortly strengthen this result (see theorem 5.22).

So far no use has been made of the anti-symmetry of the given ordering relations. We might just as well have restricted our attention to preorders rather than to posets. Taking anti-symmetry into account permits one to deduce equivalences between genuine equalities. For instance, by using the symmetry present in the first two clauses of corollary 5.14 and corollary 5.15 together with anti-symmetry of the ordering relations, we deduce:

**Corollary 5.17** If $(F, G)$ is a Galois connection then $F$ and $G$ are injective on the images of $G$ respectively $F$, i.e.

**a** $G.x = G.y \;\equiv\; F.G.x = F.G.y$ ,

**b** $F.x = F.y \;\equiv\; G.F.x = G.F.y$ .

□

The functions of a Galois connection are not only each other's duals, but they are also in a way inverse to each other. Sometimes, this property is referred to by calling $F$ and $G$ each other's *semi-inverse* or *quasi-inverse*. We adopt the former name.

**Theorem 5.18 (semi-inverse)** If $(F,G)$ is a Galois connection then

**a** $F \;=\; F \bullet G \bullet F$ ,

**b** $G \;=\; G \bullet F \bullet G$ .

**Proof** We only prove **a**, the statement **b** being the dual.

$$
\begin{array}{cl}
 & F.x \;=\; F.G.F.x \\
\equiv & \quad \{ \text{ anti-symmetry } \} \\
 & F.x \sqsubseteq F.G.F.x \quad \wedge \quad F.G.F.x \sqsubseteq F.x \\
\Leftarrow & \quad \{ \text{ cancellation with } y := F.x \ \} \\
 & F.x \sqsubseteq F.G.F.x \\
\equiv & \quad \{ \text{ 5.13(}\mathbf{b}\text{) and } \mathbf{d} \text{ with } y \ := \ F.x \ \} \\
 & G.F.x \sqsubseteq G.F.x \\
\Leftarrow & \quad \{ \text{ reflexivity } \} \\
 & true \quad .
\end{array}
$$

□

**Corollary 5.19** If $(F,G)$ is a Galois connection then $F \bullet G$ and $G \bullet F$ are idempotent.

**Proof** Follows directly from semi-inverse and the use of Leibniz with $F$, respectively $G$ .
□

We now work towards a strengthening of 5.16.

**Theorem 5.20** If $(F,G)$ is a Galois connection then

**a** $G.F.x = x \;\equiv\; x \in G.\mathcal{A}$ ,

**b** $F.G.y = y \;\equiv\; y \in F.\mathcal{B}$ .

**Proof** Again only **a** is proven, since **b** is its dual. We prove **a** by mutual implication
For the $\Rightarrow$: this is trivial, since $F.x \in \mathcal{A}$.
For the $\Leftarrow$: since $x \in G.\mathcal{A}$ we have $x = G.y$ for some $y \in \mathcal{A}$.

$$
\begin{array}{ll}
 & x = G.F.x \\
\equiv & \quad \{ \ x = G.y \ \} \\
 & G.y = G.F.G.y \\
\equiv & \quad \{ \ \text{calculus, semi-inverse} \ \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

Theorem 5.20 states that the fixed points of $G \bullet F$ are exactly the elements of $G.\mathcal{A}$ (a fixed point of an endofunction $f$ being, by definition, an element $x$ such that $f.x = x$). But the theorem is mostly used in the opposite direction. It provides an alternative expression for an element of $G.\mathcal{A}$, respectively $F.\mathcal{B}$ , that lends itself better for calculations. For an element of $G.\mathcal{A}$ we can freely introduce or remove an application of $G \bullet F$ . A dual property holds for the elements of $F.\mathcal{B}$ . These properties can also be viewed as cancellation properties.

For a Galois connection we have the cancellation laws $x \sqsubseteq G.F.x$ for any $x \in \mathcal{B}$ and $F.G.y \sqsubseteq y$ for any $y \in \mathcal{A}$ . Using this we obtain from theorem 5.20 the following

**Corollary 5.21** If $(F, G)$ is a Galois connection then

**a** $G.F.x \sqsubseteq x \equiv x \in G.\mathcal{A}$ ,

**b** $y \sqsubseteq F.G.y \equiv y \in F.\mathcal{B}$ .

□

Corollary 5.21 is more useful than theorem 5.20 when the equivalences are used as left-to-right implications.

Now we can strengthen 5.16 as promised.

**Theorem 5.22** If $(F, G)$ is a Galois connection then

**a** $F \in F.\mathcal{B} \longleftarrow G.\mathcal{A}$ is a poset-isomorphism,

**b** $G \in G.\mathcal{A} \longleftarrow F.\mathcal{B}$ is a poset-isomorphism.

Hence $G.\mathcal{A}$ and $F.\mathcal{B}$ are isomorphic posets.

**Proof** (Part **a** only.) By the definition of a poset isomorphism (a surjective poset monomorphism) we have only to supplement 5.16 by a proof that $F \in F.\mathcal{B} \longleftarrow G.\mathcal{A}$ is surjective. I.e. for each $y \in F.\mathcal{B}$ we have to exhibit an $x \in G.\mathcal{A}$

such that $y = F.x$ . Since $y = F.G.y$, by theorem 5.20(**b**), and $y \in \mathcal{A}$, $x = G.y$ is a solution.
□

As an immediate corollary we have (see property 4.9(**b**))

**Corollary 5.23** If $(F, G)$ is a Galois connection then

**a** $F$ is existentially $\sqcup_{F.\mathcal{B} \leftarrow G.\mathcal{A}}$ junctive and existentially $\sqcap_{F.\mathcal{B} \leftarrow G.\mathcal{A}}$ junctive,

**b** $G$ is existentially $\sqcup_{G.\mathcal{A} \leftarrow F.\mathcal{B}}$ junctive and existentially $\sqcap_{G.\mathcal{A} \leftarrow F.\mathcal{B}}$ junctive.
□

### 5.3.2 Alternative definitions

With the tools we now have, let us look at some equivalent formulations of a Galois connection.

The earliest definition of a Galois connection is the one introduced by O. Ore in [78]. (He called them Galois "connexions" but his peculiar spelling of the word "connection" never caught on.) Slightly differently formulated Ore's definition is captured by the next theorem.

**Theorem 5.24** $(F, G)$ is a Galois connection iff the following two clauses hold:

**a** $x \sqsubseteq G.F.x$ and $F.G.y \sqsubseteq y$ .

**b** $F$ and $G$ are monotonic.

**Proof** The proof is by mutual implication.
The $\Rightarrow$ part follows immediately from theorem 5.11 (cancellation) and corollary 5.12.
The $\Leftarrow$ part has already been proven, see the remark following theorem 5.13.
□

Definition 5.9, proposed by J. Schmidt [85], and Ore's definition, contained in theorem 5.24, both have their merits. Schmidt's is easy to remember since it contains only one clause, and lends itself to compact calculation. It is a form of "shunting rule": the game that one plays with it is to shunt occurrences of function $F$ in an expression out of the way in order to expose the function's argument. After performing some manipulations on the argument $F$ is shunted back into the picture. (Or, of course, the other way around: function $G$ is shunted temporarily out of the way.) It's an attractive strategy, requiring little creativity, that is particularly useful in inductive proofs. We will see plenty of examples later.

Ore's definition is most useful when expressed at function level. Eliminating the dummies $x$ and $y$ in 5.24(**a**) we obtain

$$(5.25) \quad I_{\mathcal{B}} \ \dot{\sqsubseteq}\ G \bullet F \qquad \text{and} \qquad F \bullet G \ \dot{\sqsubseteq}\ I_{\mathcal{A}} \ .$$

In an order-enriched category monotonic arrows $F$ and $G$ satisfying (5.25) are sometimes called "maps" and "co-maps", respectively.

Schmidt's definition can also be lifted to function level and, in combination with (5.25), can be used to construct elegant theorems. Specifically, we have:

**Theorem 5.26** $(F,G)$ is a Galois connection iff, for all functions $h$ and $k$ with the same domain and range respectively $\mathcal{B}$ and $\mathcal{A}$,

$$F \bullet h \ \dot{\sqsubseteq}\ k \quad \equiv \quad h \ \dot{\sqsubseteq}\ G \bullet k \ .$$

□

The proof is so straightforward that we choose to omit it.

An example of a calculation most neatly expressed using these forms of the definition is as follows. Suppose, for $i = 0,1$, $(\mathcal{A}_i, \sqsubseteq_{\mathcal{A}_i})$ and $(\mathcal{B}_i, \sqsubseteq_{\mathcal{B}_i})$ are posets and $(F_i \in \mathcal{A}_i \leftarrow \mathcal{B}_i,\ G_i \in \mathcal{B}_i \leftarrow \mathcal{A}_i)$ are Galois-connected pairs of functions. Thus $F_0$, $F_1$, $G_0$ and $G_1$ are all monotonic and, for $i = 0,1$,

$$(5.27) \quad I_{\mathcal{B}_i} \ \dot{\sqsubseteq}\ G_i \bullet F_i \qquad \text{and} \qquad F_i \bullet G_i \ \dot{\sqsubseteq}\ I_{\mathcal{A}_i} \ .$$

Let $h \in \mathcal{B}_0 \leftarrow \mathcal{B}_1$ and $k \in \mathcal{A}_0 \leftarrow \mathcal{A}_1$ be arbitrary functions. Then

$$(5.28) \quad F_0 \bullet h \ \dot{\sqsubseteq}\ k \bullet F_1 \quad \equiv \quad h \bullet G_1 \ \dot{\sqsubseteq}\ G_0 \bullet k \ .$$

(On a first reading of the theorem and its proof you are recommended to ignore the subscripts. The theorem generalises property (5.7) of inverse functions mentioned in section 5.2.2. The extra complication of the subscripts has been introduced because we want to kill several birds with one stone: in particular, in section 5.4 we return to this theorem and use it to observe a central property of adjoint formation.)

The proof is by mutual implication but only one implication is given since the other is entirely dual.

$$\begin{array}{ll} & F_0 \bullet h \ \ \dot{\sqsubseteq} \ \ k \bullet F_1 \\ \equiv & \qquad \{ \text{ theorem 5.26 } \} \\ & h \ \ \dot{\sqsubseteq} \ \ G_0 \bullet k \bullet F_1 \\ \Rightarrow & \qquad \{ \text{ monotonicity: (4.16) } \} \\ & h \bullet G_1 \ \ \dot{\sqsubseteq} \ \ G_0 \bullet k \bullet F_1 \bullet G_1 \\ \Rightarrow & \qquad \{ \text{ (5.25), monotonicity: (4.16), and transitivity } \} \\ & h \bullet G_1 \ \ \dot{\sqsubseteq} \ \ G_0 \bullet k \quad . \end{array}$$

For the moment we continue with pointwise calculations. The reader may wish to explore what some of our calculations would look like if they were expressed in point-free form.

There is also a sort of mixed form of definition 5.9 and theorem 5.24 that defines a Galois connection.

**Theorem 5.29** $(F, G)$ is a Galois connection iff the following three clauses hold

- $F$ is monotonic,
- $F.G.y \sqsubseteq y$ ,
- $F.x \sqsubseteq y \Rightarrow x \sqsubseteq G.y$ .

**Proof** The proof is by mutual implication.
The $\Rightarrow$ part is a direct result of theorem 5.11(**b**), corollary 5.12 and the definition of a Galois connection.
For the $\Leftarrow$ part we prove that $F$ and $G$ satisfy definition 5.9. We only have to prove $F.x \sqsubseteq y \Leftarrow x \sqsubseteq G.y$ .

$$\begin{array}{ll} & x \sqsubseteq G.y \\ \Rightarrow & \qquad \{ \ F \text{ is monotonic } \} \\ & F.x \sqsubseteq F.G.y \\ \Rightarrow & \qquad \{ \ F.G.y \sqsubseteq y \ \} \\ & F.x \sqsubseteq y \quad . \end{array}$$

□
And its dual

**Theorem 5.30** $(F, G)$ is a Galois connection iff the following three clauses hold

**a** $\quad G$ is monotonic,

**b** $\quad x \sqsubseteq G.F.x$ ,

**c** $\quad F.x \sqsubseteq y \Leftarrow x \sqsubseteq G.y$ .

□

The interest in 5.29 and 5.30 is that they are the definitions most suited to a verbal summary. Theorem 5.29, for example, states that $F$ and $G$ are Galois connected iff $F$ is monotonic and, for each $y$, $G.y$ is the greatest element $x$ such that $F.x \sqsubseteq y$. For this reason they are often favoured — they correspond to the definition in category theory of an adjunction via a so-called (co-)universal mapping property [59, pages 55–59,80–82] — even though for calculational purposes they are the least suitable of all the definitions.

### 5.3.3 Uniqueness and Existence

In this section we explore necessary and sufficient conditions for the existence of an upper or lower adjoint of a known function. First, we note that if $(F_0, G_0)$ and $(F_1, G_1)$ are Galois connections between the same posets, then $F_0 = F_1 \equiv G_0 = G_1$ . This follows from (5.28) by instantiating $h$ and $k$ to the identity functions and using the symmetry in the subscripts together with the anti-symmetry of the ordering relation. Thus, we have:

**Theorem 5.31** Each adjoint in a Galois connection uniquely determines the other adjoint.

□

From this theorem one might anticipate that each adjoint is expressible in terms of the other. That will be the concern of the current section.

Let's give a first formulation of a Galois connection in which one adjoint is expressed in terms of the other.

**Theorem 5.32** The following are equivalent:

**a** $\quad (F, G)$ is a Galois connection,

**b** $\quad F$ is monotonic and $G.y = \underline{\mathbf{max}}.(x : F.x \sqsubseteq y : x)$ ,

**c** $\quad G$ is monotonic and $F.x = \underline{\mathbf{min}}.(y : x \sqsubseteq G.y : y)$ .

**Proof** The fact that **a** equivales **b** is just a reformulation of theorem 5.29. For **a** equivales **c** use theorem 5.30.
□
Since being a least element is a stronger property than being an infimum, we obtain the following

**Corollary 5.33** If $(F, G)$ is a Galois connection then

**a** $G.y = \sqcup.(x : F.x \sqsubseteq y : x)$ ,

**b** $F.x = \sqcap.(y : x \sqsubseteq G.y : y)$ .
□
From theorem 5.32 one can extract necessary and sufficient conditions for a function to have an upper, respectively lower, adjoint.

**Theorem 5.34** Function $F \in \mathcal{A} \longleftarrow \mathcal{B}$ has an upper adjoint iff $F$ is monotonic and for every $y \in \mathcal{A}$ the equation $x :: F.x \sqsubseteq y$ has a greatest solution.

**Proof** The proof is by mutual implication.
The $\Rightarrow$ part follows directly from theorem 5.32; the greatest solution of $x :: F.x \sqsubseteq y$ is given by $G.y$ for every $y \in \mathcal{A}$ .
For the $\Leftarrow$ part define $G.y$, for every $y \in \mathcal{A}$, to be the greatest solution of $x :: F.x \sqsubseteq y$ , i.e. $G.y = \underline{\mathbf{max}}.(x : F.x \sqsubseteq y : x)$ . Since $F$ is monotonic, the result follows from theorem 5.32.
□
As a dual we have:

**Theorem 5.35** Function $G \in \mathcal{B} \longleftarrow \mathcal{A}$ has a lower adjoint iff $G$ is monotonic and for every $x \in \mathcal{B}$ the equation $y :: x \sqsubseteq G.y$ has a least solution.
□

These theorems provide one answer to the question of when a function has a lower, respectively upper adjoint. But requiring that a subset of a poset has a least or greatest element is quite a strong requirement. In fact, if we require that every non-empty subset of a poset has a least (or greatest) element, it means that the poset is totally ordered — a requirement that is much too strong. We look instead for characterisations in terms of infima and suprema rather than least and greatest elements.

From corollary 5.33 we know that an upper adjoint can be expressed as a supremum of a set. In order to extract some kind of existence theorem using infima and suprema, we first observe the following. From theorem 5.32 we see that a function has an upper adjoint if it is monotonic and some particular set has a greatest element. If a function is monotonic, it preserves greatest elements. And conversely, if a function preserves greatest elements then it is monotonic.

If we want to give an existence theorem for an upper adjoint, using suprema, it might be worthwhile to first focus on the preservation of suprema by the function . In other words, we want to establish the junctivity type of the functions involved in a Galois connection. From corollary 5.23 we know something about the type of junctivity with respect to $G.\mathcal{A}$ and $F.\mathcal{B}$ , but this says nothing about the elements outside those sets. We need something stronger for that.

**Lemma 5.36** If $(F, G)$ is a Galois connection then

**a** $F$ is existentially $\sqcup_{\mathcal{A}\longleftarrow\mathcal{B}}$ junctive,

**b** $G$ is existentially $\sqcap_{\mathcal{B}\longleftarrow\mathcal{A}}$ junctive.

**Proof** We only prove **a**, since **b** is its dual. Take any $X \subseteq \mathcal{B}$ and assume $\sqcup_{\mathcal{B}}.X$ exists. We have to show that $F.\sqcup_{\mathcal{B}}.X$ solves the defining equation of $\sqcup_{\mathcal{A}}.(F.X)$ . For any $y \in \mathcal{A}$ we derive:

$$
\begin{array}{ll}
 & F.\sqcup_{\mathcal{B}}.X \sqsubseteq y \\
\equiv & \qquad \{ \ (F,G) \text{ is a Galois connection } \} \\
 & \sqcup_{\mathcal{B}}.X \sqsubseteq G.y \\
\equiv & \qquad \{ \ G.y \in \mathcal{B} \text{ , definition supremum } \} \\
 & \forall(x : x \in X : x \sqsubseteq G.y) \\
\equiv & \qquad \{ \ (F,G) \text{ is a Galois connection } \} \\
 & \forall(x : x \in X : F.x \sqsubseteq y) \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \forall(z : z \in F.X : z \sqsubseteq y) \quad .
\end{array}
$$

□

Now we are in a position to express a Galois connection in terms of suprema and infima.

**Theorem 5.37** The following three are equivalent:

**a** $(F,G)$ is a Galois connection,

**b** $F$ is existentially $\sqcup_{\mathcal{A}\longleftarrow\mathcal{B}}$ junctive and $G.y = \sqcup.(x:\ F.x \sqsubseteq y:\ x)$ ,

**c** $G$ is existentially $\sqcap_{\mathcal{B}\longleftarrow\mathcal{A}}$ junctive and $F.x = \sqcap.(y:\ x \sqsubseteq G.y:\ y)$ .

**Proof** We only prove **a** equivales **b**. The equivalence of **a** and **c** follows by duality. The proof is by mutual implication.
**a**$\Rightarrow$**b**: This is the conjunction of corollary 5.33(**a**) and lemma 5.36(**a**).
**a**$\Leftarrow$**b**: From $G.y = \sqcup.(x:\ F.x \sqsubseteq y:\ x)$ we deduce that the supremum of $\{x \mid F.x \sqsubseteq y\}$ for every $y \in \mathcal{A}$ exists. We prove $F.z \sqsubseteq y \equiv z \sqsubseteq G.y$ by a ping-pong argument.

$$
\begin{array}{ll}
 & F.z \sqsubseteq y \\
\Rightarrow & \qquad \{\ S \sqsubseteq \sqcup.S \text{ for all sets } S\ \} \\
 & z \sqsubseteq \sqcup.(x:\ F.x \sqsubseteq y:\ x) \\
\Rightarrow & \qquad \{\ F \text{ is monotonic }\ \} \\
 & F.z \sqsubseteq F.\sqcup.(x:\ F.x \sqsubseteq y:\ x) \\
\equiv & \qquad \{\ F \text{ is existentially } \sqcup_{\mathcal{A}\longleftarrow\mathcal{B}} \text{ junctive }\ \} \\
 & F.z \sqsubseteq \sqcup.(x:\ F.x \sqsubseteq y:\ F.x) \\
\Rightarrow & \qquad \{\ \sqcup.(x:\ F.x \sqsubseteq y:\ F.x) \sqsubseteq y, \sqsubseteq \text{ is transitive }\ \} \\
 & F.z \sqsubseteq y \quad .
\end{array}
$$

□

This enables us to formulate an alternative existence theorem for a lower, respectively upper, adjoint.

**Theorem 5.38** A function $F \in \mathcal{A} \longleftarrow \mathcal{B}$ has an upper adjoint iff $F$ is existentially $\sqcup_{\mathcal{A}\longleftarrow\mathcal{B}}$ junctive and the set $\{x \mid F.x \sqsubseteq y\}$ has a supremum for every $y \in \mathcal{A}$.

**Proof** The proof is by mutual implication.
The $\Rightarrow$ part follows directly from theorem 5.37.
For the $\Leftarrow$ part: define for all $y \in \mathcal{A}$ , $G.y$ as $\sqcup.(x:\ F.x \sqsubseteq y:\ x)$ . This supremum is well defined, by assumption. The rest follows from theorem 5.37.
□
As a dual we have

**Theorem 5.39** A function $G \in \mathcal{B} \longleftarrow \mathcal{A}$ has a lower adjoint iff $G$ is existentially $\sqcap_{\mathcal{B}\longleftarrow\mathcal{A}}$ junctive and the set $\{y \mid x \sqsubseteq G.y\}$ has an infimum for every $x \in \mathcal{B}$.

□

The theorems in this section are used to establish the existence of an adjoint, and thus a Galois connection, without giving an explicit formula for the adjoint. Yet it is possible to give an expression for the adjoint, in terms of an extremal element. In general that expression is not amenable to manipulation, so it is hardly ever used.

**Exercise 5.40 (Perfect Connections)** Suppose $(F, G)$ is a Galois connection. It is possible that all elements of $\mathcal{B}$ are fixed points of $G \bullet F$ . Ore would say: the Galois connection is *perfect* in $\mathcal{B}$. In [71] this is called a *Galois insertion from* $\mathcal{B}$ *to* $\mathcal{A}$. There are several ways to express this property:

**a** $\forall(x :\ x \in \mathcal{B} :\ F.x = \underline{\mathbf{min}}.(y :\ x = G.y :\ y))$ ,

**b** $\forall(x :\ x \in \mathcal{B} :\ G.F.x = x)$ ,

**c** $G$ is surjective,

**d** $F$ is a poset-monomorphism,

**e** $F$ is injective.

Prove that all these expressions are equivalent. Further, prove that any one of the above implies

- $\forall(x :\ x \in \mathcal{B} :\ F.x = \sqcap.(y :\ x = G.y :\ y))$ .

What is the dual of this theorem?

□

So much for Galois connections for partial orders. The theorems encountered so far form a substantial part of the known, or rather documented, theorems about Galois connections. In particular we have introduced most of the theorems that are useful for calculational purposes.

Some of the theorems depend on the existence of suprema or infima. If we have a structure where the existence of those extremal elements is trivial, one might be able to improve some of the results stated in this section.

### 5.3.4 Complete lattices

There are two orderings that play a rôle in a Galois connection. If we want to adhere to the symmetry between these orderings and the theorems, it would be advantageous to take both orderings to be complete lattices. However, that is quite a strong requirement. We will only assume —unless stated otherwise — that just one of the orderings is a complete lattice. When we give the dual of a theorem, we will have to require that the other ordering is a complete lattice.

In this section we merely improve on some of the theorems already mentioned.

Assume that $\mathcal{B}$ is a complete lattice. We can now characterise the functions that have an upper adjoint in the following concise way.

**Theorem 5.41** A function $F \in \mathcal{A} \longleftarrow \mathcal{B}$ has an upper adjoint iff $F$ is universally $\sqcup_{\mathcal{A}\longleftarrow\mathcal{B}}$ junctive.

**Proof** Since $\mathcal{B}$ is a complete lattice, we know the set $\{x \mid F.x \sqsubseteq y\}$ has a supremum in $\mathcal{B}$ for every $y \in \mathcal{A}$ . From theorem 5.38 we deduce that $F$ has an upper adjoint iff $F$ is existentially $\sqcup_{\mathcal{A}\longleftarrow\mathcal{B}}$ junctive. With $\mathcal{B}$ being a complete lattice, this is equivalent to $F$ being universally $\sqcup_{\mathcal{A}\longleftarrow\mathcal{B}}$ junctive which completes the proof.
□

For the dual, assume $\mathcal{A}$ is a complete lattice. We then obtain the following:

**Theorem 5.42** A function $G \in \mathcal{B} \longleftarrow \mathcal{A}$ has a lower adjoint iff $G$ is universally $\sqcap_{\mathcal{B}\longleftarrow\mathcal{A}}$ junctive.

□

The previous two theorems can be used in two different ways. If one wants to prove that a function is universally $\sqcup$-junctive, one only has to prove that the function has an upper adjoint. On the other hand, if one wants to establish that a function has an upper adjoint, it is sufficient to prove that the function is universally $\sqcup$-junctive. This gives a nice existence theorem which will be exploited extensively later.

We now focus our attention on the image sets of $F$ and $G$. We already know that they are isomorphic posets. With $\mathcal{B}$ being a complete lattice, we can do better.

**Theorem 5.43** If $\mathcal{B}$ is a complete lattice, the poset $G.\mathcal{A}$ is a complete lattice. Moreover the infima in $G.\mathcal{A}$ coincide with the infima of $\mathcal{B}$.

**Proof** To show that $G.\mathcal{A}$ is a complete lattice, it is sufficient to show all infima exist. Take any $X \subseteq G.\mathcal{A}$. We have to show that $\sqcap_{G.\mathcal{A}}.X$ exist. We do that by demonstrating it is equal to $\sqcap_{\mathcal{B}}.X$ , which exists. By property 3.50(**b**) it is sufficient to show that $\sqcap_{\mathcal{B}}.X \in G.\mathcal{A}$ .

$$\begin{array}{ll}
& \sqcap_{\mathcal{B}}.X \in G.\mathcal{A} \\
\equiv & \qquad \{ \text{ corollary 5.21}(\mathbf{a}) \ \} \\
& G.F.\sqcap_{\mathcal{B}}.X \sqsubseteq \sqcap_{\mathcal{B}}.X \\
\Leftarrow & \qquad \{ \ G.F \text{ is monotonic, hence } G.F.\sqcap_{\mathcal{B}}.X \sqsubseteq \sqcap_{\mathcal{B}}.G.F.X \ \} \\
& \sqcap_{\mathcal{B}}.G.F.X \sqsubseteq \sqcap_{\mathcal{B}}.X \\
\equiv & \qquad \{ \ X \subseteq G.\mathcal{A} \text{ theorem 5.20}(\mathbf{a}) \ \} \\
& \sqcap_{\mathcal{B}}.X \sqsubseteq \sqcap_{\mathcal{B}}.X \\
\equiv & \qquad \{ \text{ calculus } \} \\
& \mathit{true} \quad .
\end{array}$$

□

Given the fact that $\mathcal{B}$ is a complete lattice, we now know that $G.\mathcal{A}$ is a complete lattice and the infima in $G.\mathcal{A}$ coincide with the infima in $\mathcal{B}$ . We also know the suprema in $G.\mathcal{A}$ always exist. Alas, the suprema in $G.\mathcal{A}$ do not, in general, coincide with the suprema of $\mathcal{B}$.

**Theorem 5.44** If $\mathcal{B}$ is a complete lattice, then for any $X \subseteq G.\mathcal{A}$ the supremum in $G.\mathcal{A}$, $\sqcup_{G.\mathcal{A}}.X$, is $G.F.\sqcup_{\mathcal{B}}.X$ .

**Proof** Take any $X \subseteq G.\mathcal{A}$ . We prove that $G.F.\sqcup_{\mathcal{B}}.X = \sqcup_{G.\mathcal{A}}.X$ by mutual containment.

$$\begin{array}{ll}
& \sqcup_{G.\mathcal{A}}.X \\
\sqsubseteq & \qquad \{ \ X = G.F.X \sqsubseteq G.F.\sqcup_{\mathcal{B}}.X \text{ since } G.F \text{ monotonic } \} \\
& G.F.\sqcup_{\mathcal{B}}.X \\
\sqsubseteq & \qquad \{ \ G.F \text{ monotonic, } \sqcup_{\mathcal{B}}.X \sqsubseteq \sqcup_{G.\mathcal{A}}.X \ \} \\
& G.F.\sqcup_{G.\mathcal{A}}.X \\
= & \qquad \{ \ \sqcup_{G.\mathcal{A}}.X \in G.\mathcal{A} \text{ theorem 5.20}(\mathbf{a}) \ \} \\
& \sqcup_{G.\mathcal{A}}.X \quad .
\end{array}$$

□

As a dual to theorem 5.43 and theorem 5.44 we have the following

**Theorem 5.45** If $\mathcal{A}$ is a complete lattice, the poset $F.\mathcal{B}$ is a complete lattice. The suprema in $F.\mathcal{B}$ coincide with the suprema in $\mathcal{A}$, and for any $Y \subseteq F.\mathcal{B}$ the infimum in $F.\mathcal{B}$, i.e. $\sqcap_{F.\mathcal{B}}.Y$, is given by $F.G.\sqcap_{\mathcal{A}}.Y$ .
$\square$

So far we have proved that if $\mathcal{B}$ is a complete lattice then so is $G.\mathcal{A}$ . A dual result holds for $\mathcal{A}$ and $F.\mathcal{B}$ . Now is the time to claim that when $\mathcal{B}$ and $\mathcal{A}$ are complete lattices then so are $F.\mathcal{B}$ and $G.\mathcal{A}$. In fact they are isomorphic complete lattices, since $F \in F.\mathcal{B} \longleftarrow G.\mathcal{A}$ and $G \in G.\mathcal{A} \longleftarrow F.\mathcal{B}$ are both poset-isomorphisms; see also corollary 5.23.

But we can do better. For $F.\mathcal{B}$ being a complete lattice it is not necessary that $\mathcal{A}$ is a complete lattice. By using corollary 5.23, we can construct suprema and infima of $F.\mathcal{B}$ even when $\mathcal{A}$ is not a complete lattice.

**Theorem 5.46** If $\mathcal{B}$ is a complete lattice, then $F.\mathcal{B}$ is a complete lattice. The supremum and infimum operators in $F.\mathcal{B}$ are given by:

**a** $\quad \sqcup_{F.\mathcal{B}}.Y \;=\; F.\sqcup_{G.\mathcal{A}}.G.Y \;=\; F.\sqcup_{\mathcal{B}}.G.Y$ ,

**b** $\quad \sqcap_{F.\mathcal{B}}.Y \;=\; F.\sqcap_{G.\mathcal{A}}.G.Y \;=\; F.\sqcap_{\mathcal{B}}.G.Y$ .

**Proof** For any $Y \subseteq F.\mathcal{B}$ we observe

$$
\begin{array}{ll}
 & F.\sqcup_{\mathcal{B}}.G.Y \\
= & \quad \{ \text{ semi-inverse } \} \\
 & F.G.F.\sqcup_{\mathcal{B}}.G.Y \\
= & \quad \{ \text{ theorem 5.44 } \} \\
 & F.\sqcup_{G.\mathcal{A}}.G.Y \\
= & \quad \{ \text{ corollary 5.23}(\mathbf{a}) \text{ } \} \\
 & \sqcup_{F.\mathcal{B}}.F.G.Y \\
= & \quad \{ \; Y \subseteq F.\mathcal{B}, \text{ theorem 5.20}(\mathbf{a}) \text{ } \} \\
 & \sqcup_{F.\mathcal{B}}.Y \quad .
\end{array}
$$

And

$$
\begin{array}{ll}
 & F.\sqcap_{\mathcal{B}}.G.Y \\
= & \quad \{ \text{ theorem 5.43 } \} \\
 & F.\sqcap_{G.\mathcal{A}}.G.Y \\
= & \quad \{ \text{ corollary 5.23}(\mathbf{a}) \text{ } \} \\
 & \sqcap_{F.\mathcal{B}}.F.G.Y \\
= & \quad \{ \; Y \subseteq F.\mathcal{B}, \text{ theorem 5.20}(\mathbf{a}) \text{ } \} \\
 & \sqcap_{F.\mathcal{B}}.Y \quad .
\end{array}
$$

□

As a dual we have

**Theorem 5.47** If $\mathcal{A}$ is a complete lattice, then $G.\mathcal{A}$ is a complete lattice. The supremum and infimum operators in $G.\mathcal{A}$ are given by:

**a** $\sqcup_{G.\mathcal{A}}.X \;=\; G.\sqcup_{F.\mathcal{B}}.F.X \;=\; G.\sqcup_{\mathcal{A}}.F.X$ ,

**b** $\sqcap_{G.\mathcal{A}}.X \;=\; G.\sqcap_{F.\mathcal{B}}.F.X \;=\; G.\sqcap_{\mathcal{A}}.F.X$ .

□

So we have the following result.

**Theorem 5.48** If $\mathcal{A}$ or $\mathcal{B}$ is a complete lattice then $F.\mathcal{B}$ and $G.\mathcal{A}$ are isomorphic complete lattices.

□

This theorem is rarely cited in the literature. Only [72] mentions this result.

## 5.4 Sharp and Flat

We now know a great deal about Galois connections. In particular we know that for complete lattices $\mathcal{A}$ and $\mathcal{B}$ there is a (1-1) correspondence between universally $\sqcup$-junctive functions in $\mathcal{A} \leftarrow \mathcal{B}$ and universally $\sqcap$-junctive functions in $\mathcal{B} \leftarrow \mathcal{A}$. Since we also know that these two sets of functions form complete lattices (see (4.13)) a natural question to ask is whether the two lattices are isomorphic. Indeed they are as we will now show.

Let $F \in \mathcal{A} \leftarrow \mathcal{B}$ be universally $\sqcup$-junctive. Denote its upper adjoint by $F^\sharp$. Let $G \in \mathcal{B} \leftarrow \mathcal{A}$ be universally $\sqcap$-junctive. Denote its lower adjoint by $G^\flat$. (Note: we do *not* assume that the pair $(F,G)$ forms a Galois connection.) Then, by definition,

(5.49) $F.x \sqsubseteq y \;\;\equiv\;\; x \sqsubseteq F^\sharp.y$ ,

and

(5.50) $G^\flat.x \sqsubseteq y \;\;\equiv\;\; x \sqsubseteq G.y$ .

Moreover, $F^\sharp$ is universally $\sqcap$-junctive and $G^\flat$ is universally $\sqcup$-junctive.

*Remark* You may wish to pronounce $F^\sharp$ as "$F$ upper" and $G^\flat$ as "$G$ lower". We, ourselves, tend to pronounce operators according to the name of the symbol used to denote them. So we pronounce $F^\sharp$ as "$F$ sharp" and $G^\flat$ as "$G$ flat". This has the advantage that when calculating with the operators we oblige ourselves to consult their algebraic properties rather being (mis)guided by any intuition we have about the "meaning" of the operators. *(End of Remark)*

The functions $^\sharp$ and $^\flat$ form the (1-1) correspondence mentioned above since, by making the substitutions $G \;:=\; F^\sharp$ in (5.50) and $F \;:=\; G^\flat$ in (5.49),

$$(5.51)\quad F^{\sharp\flat}.x \sqsubseteq y \;\;\equiv\;\; x \sqsubseteq F^\sharp.y \;\;,$$

and

$$(5.52)\quad G^\flat.x \sqsubseteq y \;\;\equiv\;\; x \sqsubseteq G^{\flat\sharp}.y \;\;.$$

(These substitutions are permitted because of the junctivity properties of $F^\sharp$ and $G^\flat$.) So, by the unicity of adjoints,

$$(5.53)\qquad F^{\sharp\flat} \;=\; F \qquad \text{and} \qquad G \;=\; G^{\flat\sharp} \;\;.$$

With this notation the cancellation laws are now expressed by two pairs of inclusions

$$(5.54)\qquad F \bullet F^\sharp \;\dot{\sqsubseteq}\; I_{\mathcal{A}} \qquad \text{and} \qquad I_{\mathcal{B}} \;\dot{\sqsubseteq}\; F^\sharp \bullet F \;\;,$$

$$(5.55)\qquad G^\flat \bullet G \;\dot{\sqsubseteq}\; I_{\mathcal{A}} \qquad \text{and} \qquad I_{\mathcal{B}} \;\dot{\sqsubseteq}\; G \bullet G^\flat \;\;.$$

It is now straightforward to show that $^\sharp$ and $^\flat$ form an order isomorphism. We first observe that they are themselves adjoints in a "perfect" Galois connection (see exercise 5.40):

$$(5.56)\quad F^\sharp \;\dot{\sqsubseteq}\; G \;\;\equiv\;\; F \;\dot{\sqsupseteq}\; G^\flat \;\;.$$

(Note the reversal of the orderings.) This follows immediately from (5.28) by making the substitutions $F_0 \;:=\; G^\flat$, $G_0 \;:=\; G$, $F_1 \;:=\; F$, $G_1 \;:=\; F^\sharp$, $h \;:=\; I_{\mathcal{B}}$ and $k \;:=\; I_{\mathcal{A}}$ . The fact that it is a perfect connection follows from (5.53), which expresses that both $^\flat$ and $^\sharp$ are surjective.

Having the surjectivity present, it remains to prove that $^\flat$ or $^\sharp$ is a poset monomorphism. Combining (5.56) with (5.53) we obtain, for all universally $\sqcup$-junctive functions $F_0$ and $F_1$,

$$\begin{array}{ll}
 & F_0^\sharp \;\dot{\sqsubseteq}\; F_1^\sharp \\
\equiv & \qquad \{\ (5.56)\ \} \\
 & F_0 \;\dot{\sqsupseteq}\; F_1^{\sharp\flat} \\
\equiv & \qquad \{\ (5.53)\ \} \\
 & F_0 \;\dot{\sqsupseteq}\; F_1 \quad ,
\end{array}$$

which establishes the claimed (contravariant) poset isomorphism.

## 5.5 Historical Examples

In this final section we present several examples of Galois connections drawn from the computing science literature. The approach taken here is conventional so that the examples can easily be recognised. We return to several of the examples later in the text but when we do we approach them differently. No further use will be made of the examples here, so that, apart from their historical interest, they may safely be omitted.

### 5.5.1 Relations and Set-Valued Functions

As a preliminary to our first two examples we record first two well-known bijections between binary relations and set-valued functions.

**Definition 5.57** For $R \subseteq \mathcal{X} \times \mathcal{Y}$, a function to $I\!P(\mathcal{X})$ from $\mathcal{Y}$ is defined by taking for every $y \in \mathcal{Y}$ :

$$R.y \;=\; \{x \mid xRy\} \;\; .$$

□

By elementary set calculus, we observe that $x \in R.y \equiv xRy$ .
In the same vein we make the following definition.

**Definition 5.58** For $R \subseteq \mathcal{X} \times \mathcal{Y}$ we define a function to $I\!P(\mathcal{Y})$ from $\mathcal{X}$ by defining for every $x \in \mathcal{X}$ :

$$x.R \;=\; \{y \mid xRy\} \;\; .$$

□

Note that a relation is fully determined by either one of these functions. Furthermore we observe the following connection

$$x.R \ni y \quad \equiv \quad x \in R.y \;\; ,$$

for every $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. When we view $x.R$ as a predicate on $\mathcal{Y}$ and $R.y$ as a predicate on $\mathcal{X}$, this connection translates into

$$xRy \quad = \quad (x.R).y \quad = \quad x.(R.y) \;\; ,$$

for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

A similar description, but with different notation, can also be found in [85].

### 5.5.2 Polarities

Our first example is by now a classic. It was first introduced by G. Birkhoff in 1940 and can be viewed as a starting point for the interest in Galois connections. The description given here is based upon the description given in [24].

Let $R$ be a relation between sets $\mathcal{X}$ and $\mathcal{Y}$ . The functions $x.R$ and $R.y$ can be *lifted* to functions to $\mathbb{P}(\mathcal{Y})$ from $\mathbb{P}(\mathcal{X})$, respectively to $\mathbb{P}(\mathcal{X})$ from $\mathbb{P}(\mathcal{Y})$. These functions are called *polars* in [24].

**Definition 5.59** For every $X \in \mathbb{P}(\mathcal{X})$ define the *right polar* as

$$\langle\!| X |\!\rangle R \;=\; \cap.(x:\; x \in X:\; x.R) \;\; .$$

For every $Y \in \mathbb{P}(\mathcal{Y})$ define the *left polar* as

$$R\langle\!| Y |\!\rangle \;=\; \cap.(y:\; y \in Y:\; R.y) \;\; .$$

□

If we take $x \in \mathcal{X}$ then $\langle\!|\{x\}|\!\rangle R = x.R$ . A similar property holds for the left polar. Hence a relation is fully determined by either one of its polars.

These two polars are connected. Indeed, they are Galois connected. For $R \subseteq \mathcal{X} \times \mathcal{Y}$ , $X \in \mathbb{P}(\mathcal{X})$ and $Y \in \mathbb{P}(\mathcal{Y})$ we have

**Theorem 5.60** $\langle\!| X |\!\rangle R \supseteq Y \quad\equiv\quad X \times Y \subseteq R \quad\equiv\quad X \subseteq R\langle\!| Y |\!\rangle$ .

**Proof**

$$
\begin{array}{ll}
 & X \subseteq R\langle\!| Y |\!\rangle \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \forall(x:\; x \in X:\; x \in R\langle\!| Y |\!\rangle) \\
\equiv & \qquad \{ \text{ definition 5.59 } \} \\
 & \forall(x:\; x \in X:\; \forall(y:\; y \in Y:\; xRy)) \\
\equiv & \qquad \{ \text{ definition } \times \} \\
 & X \times Y \subseteq R \\
\equiv & \qquad \{ \text{ definition } \times \} \\
 & \forall(y:\; y \in Y:\; \forall(x:\; x \in X:\; xRy)) \\
\equiv & \qquad \{ \text{ definition 5.59 } \} \\
 & \forall(y:\; y \in Y:\; \langle\!| X |\!\rangle R \ni y) \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \langle\!| X |\!\rangle R \supseteq Y \quad .
\end{array}
$$

□

Most of the formal properties of polars can easily be deduced by instantiating the general properties of the Galois connections, see section 5.3. For more properties, especially the applicability of polars to other examples, in the field of geometry, theory of rings and groups, the reader is referred to [24].

Recently, the polars have acquired a new jacket. They popped up in the field of *formal concept analysis* [31]. Let's give a brief description to see the connection with polars.

A *context* is a triple $(\mathsf{G}, \mathsf{M}, R)$, where $\mathsf{G}$, called the objects, and $\mathsf{M}$, called the attributes, are sets and $R \subseteq \mathsf{G} \times \mathsf{M}$. Hence $R$ relates objects and attributes. For $G \subseteq \mathsf{G}$ , $\{G\}R$ is the set of attributes common to all objects in $G$ . Similarly for $M \subseteq \mathsf{M}$ , $R\{M\}$ is the set of objects possessing all the attributes in $M$.

In this context, a *concept* is a pair $(G, M)$ with $G = R\{M\}$ and $M = \{G\}R$ . The set of all concepts in a context $(\mathsf{G}, \mathsf{M}, R)$ is denoted $\mathcal{B}(\mathsf{G}, \mathsf{M}, R)$ . On this set one can define an ordering $\leq$ as follows:

$$(g_0, m_0) \leq (g_1, m_1) \ \equiv \ g_0 \subseteq g_1 \ .$$

It is easy to show that $g_0 \subseteq g_1$ is equivalent to $m_1 \subseteq m_0$ . With this ordering, the set $\mathcal{B}(\mathsf{G}, \mathsf{M}, R)$ forms a complete lattice: the concept lattice. Without going into further details, all the properties of *formal concepts* are easily proven by using general properties of Galois connections, specifically by the properties of the polars. This was also noted in [31].

The polars arise in another important disguise. They form a pointwise basis for *factors*. Factors will be discussed after the next example, the weakest liberal precondition.

### 5.5.3 The weakest liberal precondition

There is another way of lifting a relation $S$ on $\mathcal{X} \times \mathcal{Y}$ into a function to $I\!P(\mathcal{Y})$ from $I\!P(\mathcal{X})$, or to $I\!P(\mathcal{X})$ from $I\!P(\mathcal{Y})$. Like the previous example, the functions $x.S$ and $S.y$ can be lifted but in a different way. For lack of standard nomenclature, these functions will be called *image* functions.

**Definition 5.61** For $X \in I\!P(\mathcal{X})$ define the *right image* of the relation $S \subseteq \mathcal{X} \times \mathcal{Y}$ by

$$[X]S \ = \ \cup.(x : \ x \in X : \ x.S) \ ,$$

□

**Definition 5.62** For $Y \in I\!P(\mathcal{Y})$ define the *left image* of the relation $S \subseteq \mathcal{X} \times \mathcal{Y}$ by

$$S[Y] \;=\; \cup.(y:\; y \in Y:\; S.y) \;\;.$$

□

For functions, the notion of an image is well known. When we consider a function as a relation, the left image of that relation is the image of the function. That is the reason for the name "image function".

As is the case for polar functions, any one of these image functions fully determines the relation, since for any $x \in \mathcal{X}$ : $[\{x\}]S = x.S$ . A dual equality holds for the left image function.

Knowing the Galois connection for polars, one might anticipate a similar result for the image functions. There is indeed a connection between the image functions, even a Galois connection.

**Theorem 5.63** For all $X \in I\!P(\mathcal{X})$ and $Y \in I\!P(\mathcal{Y})$:

$$[X]S \subseteq Y \quad\equiv\quad X \subseteq S^{\diamond}[Y] \;\;,$$

where $S^{\diamond}[\_]$ denotes the *conjugate* of $S[\_]$.

**Proof** Since $[X]S \subseteq Y$ is equivalent to $\forall(x:\; x \in X:\; x.S \subseteq Y)$, it is sufficient to prove $x.S \subseteq Y \equiv x \in S^{\diamond}[Y]$ . We derive for any $x \in X$

$$\begin{array}{ll}
 & x \in S^{\diamond}[Y] \\
\equiv & \qquad \{ \text{ definition left image, calculus } \} \\
 & x \notin \cup.(y:\; y \notin Y:\; S.y) \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \forall(y:\; y \notin Y:\; x \notin S.y) \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \forall(y:\; y \notin Y:\; \neg(x.S.y)) \\
\equiv & \qquad \{ \text{ trading } \} \\
 & \forall(y:\; x.S.y:\; y \in Y) \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & x.S \subseteq Y \quad .
\end{array}$$

□

By instantiating $X, Y := \neg X, \neg Y$ in theorem 5.63 and some simplifications of the resulting expression, we find

**Corollary 5.64** $\quad X \supseteq S[Y] \quad \equiv \quad [X]S^{\diamond} \supseteq Y$ .

□

The function $S^{\diamond}[\_]$ is well studied in computing science, albeit in a somewhat narrower setting and, of course, under a different name.

Let's call $\mathcal{X}$ the statespace. Take $\perp \notin \mathcal{X}$ and define for any $X \in I\!P(\mathcal{X})$ : $X_{\perp} = X \cup \{\perp\}$. A program $S$ can be modelled by a relation $S \subseteq \mathcal{X} \times \mathcal{X}_{\perp}$. The $\perp$ is used to represent a nonterminating computation. Hence a program *maps* states from $\mathcal{X}$ onto states in $\mathcal{X}_{\perp}$. This means it either terminates in some state of $\mathcal{X}$, or it doesn't terminate, which is modelled by $\perp$. Note that, since $S$ is a relation and not necessarily a function, non-determinancy can easily be dealt with in this framework.

A well-established method in showing the correctness of programs is by way of so-called Hoare-triples. For $P, Q \subseteq \mathcal{X}$ this means showing the validity of $\{P\}\, S\, \{Q\}$; i.e. show that the program $S$, when started in a state belonging to $P$, either terminates in a state belonging to $Q$ or doesn't terminate at all. In the relational setting this amounts to the validity of $[P]S \subseteq Q_{\perp}$ . Using the Galois connection theorem 5.63 this is equivalent to $P \subseteq S^{\diamond}[Q_{\perp}]$ .

In computing science one writes $\mathsf{wlp}.S.Q$ instead of $S^{\diamond}[Q_{\perp}]$ . So the connection between the weakest liberal precondition and Hoare triples

$$P \subseteq \mathsf{wlp}.S.Q \quad \equiv \quad \{P\}\, S\, \{Q\}$$

is a Galois connection. It was noted in [36] that $\mathsf{wlp}.S.Q$ is an extremal solution of an equation involving a Hoare triple. To be precise, it is the greatest —or in the terminology used in [36]; the weakest— solution of $X :: \{X\}\, S\, \{Q\}$ .

### 5.5.4 Factors

One very fine example of a Galois connection is provided by the factors. They first appeared in [37] under the name *residuals*. They are used under the same name in [24]. The name *factor* is used by [30] in the context of regular languages and finite machines. They also, more recently, were used in program specification [52] under the names *weakest pre-* and *postspecifications*. We also make much use of them later.

The approach to factors given in this section is based on the polarities. This means that we discuss the factors in a specific model. The factors will be defined in terms of their elements. It is very well possible to define factors without using elements, as is done in [52] and later in this monograph. The advantage of using polarities to define factors, and hence the reason for introducing the notion in this way, is that it may help in recognising a factor in a pointwise definition. One cannot recognise a factor in an expression involving elements if one is unaware of the pointwise definition of factors.

The notation used in this section is that introduced in section 5.5.1. All the relations used here are subsets of $\mathcal{X} \times \mathcal{X}$.

**Definition 5.65** For $R$ and $S$ relations, define the *right factor* $R\backslash S$ by

$$x.(R\backslash S) \;=\; \langle\!| R.x |\!\rangle S \;,$$

for all $x \in \mathcal{X}$. For relations $T$ and $S$ the *left factor* $S/T$ is defined by

$$(S/T).y \;=\; S\langle\!| y.T |\!\rangle \;,$$

for all $y \in \mathcal{X}$.
□

Since there is a Galois connection between the polars, one might anticipate a Galois connection for the factors. Indeed, one can prove:

**Theorem 5.66** $R \subseteq S/T \quad\equiv\quad R\backslash S \supseteq T \;.$

**Proof**

$$\begin{array}{ll}
 & R \subseteq S/T \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \forall(z :: \; R.z \subseteq (S/T).z) \\
\equiv & \qquad \{ \text{ definition 5.65 } \} \\
 & \forall(z :: \; R.z \subseteq S\langle\!| z.T |\!\rangle) \\
\equiv & \qquad \{ \text{ Galois connection for polars, theorem 5.60 } \} \\
 & \forall(z :: \; \langle\!| R.z |\!\rangle S \supseteq z.T) \\
\equiv & \qquad \{ \text{ definition 5.65, calculus } \} \\
 & R\backslash S \supseteq T \quad .
\end{array}$$

□

This Galois connection is not the only one that can be given for factors. There is a more interesting one. Before embarking on that one, let's first give a lemma that is interesting in its own right.

**Lemma 5.67** For $R$, $S$ and $T$ relations:

**a** $x(R\backslash S)y \quad \equiv \quad R.x \subseteq S.y$ ,

**b** $x(S/T)y \quad \equiv \quad x.S \supseteq y.T$ .

**Proof** Only **a** is proven; **b** can be proven likewise.

$$
\begin{array}{ll}
 & x(R\backslash S)y \\
\equiv & \qquad \{ \text{ definition 5.65 } \} \\
 & \llbracket R.x \rrbracket S \ni y \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \llbracket R.x \rrbracket S \supseteq \{y\} \\
\equiv & \qquad \{ \text{ Galois connection for polars, theorem 5.60 } \} \\
 & R.x \subseteq S\llbracket \{y\} \rrbracket \\
\equiv & \qquad \{ \text{ definition polar } \} \\
 & R.x \subseteq S.y \quad .
\end{array}
$$

□

In [52] it is observed that for factors the following holds

$$x(R\backslash S)y \quad \equiv \quad \forall(z :: \ zRx \Rightarrow zSy) \ .$$

Our notation eliminates the dummy $z$.

Now let's give a more interesting Galois connection for factors.

**Theorem 5.68** For $R$, $S$ and $T$ relations

$$R \circ S \subseteq T \quad \equiv \quad R \subseteq T/S \ .$$

**Proof**

$$
\begin{array}{ll}
 & R \circ S \subseteq T \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \forall(x :: \ x.(R \circ S) \subseteq x.T)
\end{array}
$$

$$\begin{array}{ll} \equiv & \{ \text{ calculus } \} \\ & \forall(x,y:\ x.R.y:\ y.S \subseteq x.T) \\ \equiv & \{ \text{ lemma 5.67}(\mathbf{b}) \ \} \\ & \forall(x,y:\ x.R.y:\ x.(T/S).y) \\ \equiv & \{ \text{ calculus } \} \\ & R \subseteq T/S \quad . \end{array}$$

□

Combining theorem 5.66 and theorem 5.68 leads to:

**Theorem 5.69** For $R$, $S$ and $T$ relations

$$R \circ S \subseteq T \ \equiv \ S \subseteq R\backslash T \ .$$

□

Theorem 5.68 is used in [52] as the definition of the weakest prespecification although there they write $P\backslash R$ instead of $R/P$. The reason for preferring the latter is that factors —like any component of a Galois connection— enjoy a *cancellation* property. Taking $R := T/S$ in theorem 5.68 gives the cancellation property:

$$(T/S) \circ S \subseteq T \ .$$

In this expression the two adjacent occurrences of $S$ cancel one another. Using the notation suggested in [52], this property would read as:

$$(S\backslash T) \circ S \subseteq T \ ,$$

in which the two occurrences of $S$ that cancel are not adjacent. This makes it more difficult to remember the property or to see that cancellation is applicable, especially when $S$ or $T$ is a long formula. The choice for the notation of the factors is based upon economy of calculation. A similar argument can be given against the notation used in [24].

So much for the factors in a relational setting. The factors also appear in another setting: the theory of regular languages. In [30] they were introduced as a tool for expressing a regular expression $E$ as a regular function of $F_1$, $F_2$, ... with the $F_i$ to be determined. We will define factors for regular languages in the same vein as is done for relations. Notice the analogy with factors in a relational setting. The reader is urged to translate the theorems about factors for regular languages into a relational setting.

It is tacitly assumed that the reader has some knowledge of the theory of regular languages. The symbol $\cdot$ will be used to denote the *concatenation-operator*, $\cup$ stands for set union and the $\leq$ is the complete lattice ordering on regular languages, to be precise: $F \ \leq \ E \quad \equiv \quad E \ = \ E \cup F$ . We will not distinguish between a one-element language and a word, as is common practice.

One of the standard operations for regular languages is the *derivative*.

**Definition 5.70** For a regular language $E$ and a word $w$, the *word-derivative* $E_w$ is defined by $E_w \ = \ \{v \mid w \cdot v \ \in \ E\}$ .

□

Observe that for a regular language $E$ and a word $w$ we have

(5.71) $\neg E_w \ = \ (\neg E)_w$ ,

since for any word $v$ we derive

$$
\begin{array}{ll}
 & v \ \in \ (\neg E)_w \\
\equiv & \qquad \{ \text{ definition 5.70 } \} \\
 & w \cdot v \ \in \ \neg E \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & w \cdot v \ \notin \ E \\
\equiv & \qquad \{ \text{ definition 5.70 } \} \\
 & v \ \notin \ E_w \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & v \ \in \ \neg(E_w) \quad .
\end{array}
$$

Another frequently appearing notation for the word-derivative $E_w$ is $\partial_w E$. The word-derivative can be seen as a function which maps a word onto a regular language. This function can be generalised in order to obtain a function that maps a regular language onto a regular language.

**Definition 5.72** For regular languages $E$ and $F$ we define the *derivative* by $\partial_F E \ = \ \cup.(w : \ w \ \in \ F : \ E_w)$ .

□

By taking $F = \{w\}$ one easily sees that the derivative is indeed a generalisation of the word-derivative. A more familiar, but completely equivalent way of defining the derivative is by

$$(5.73) \quad \partial_F E \;=\; \{v \mid F{\cdot}v \cap E \neq \emptyset\} \quad .$$

Although the notation for the derivative suggests that it is a function of $E$, it can also be seen as a function of $F$. These functions are very useful for regular languages. They are for example used to efficiently construct a finite machine that accepts a given regular language.

The word-derivative can be generalised in another way.

**Definition 5.74** For regular languages $E$ and $F$, the *right factor* is defined by $F\backslash E \;=\; \cap.(w :\; w \in F :\; E_w)$ .

□

Even for those who are familiar with regular languages, it might be the first time they have come across factors.

As was the case for relations, there is a Galois connection for the right factor.

**Theorem 5.75** For $E, F$ and $G$ regular languages we have $F \cdot G \leq E \equiv G \leq F\backslash E$ .

**Proof**

$$
\begin{array}{cl}
 & G \leq F\backslash E \\
\equiv & \qquad \{ \text{ definition 5.74 } \} \\
 & G \leq \cap.(w :\; w \in F :\; E_w) \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \forall(w :\; w \in F :\; G \leq E_w) \\
\equiv & \qquad \{ \text{ definition 5.70 } \} \\
 & \forall(w :\; w \in F :\; w \cdot G \leq E) \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & F \cdot G \leq E \quad .
\end{array}
$$

□

There is an intimate relation between the right factor and the derivative. The derivative and the right factor are each other's conjugates.

**Theorem 5.76** For regular languages $E$ and $F$:

$$(\partial_F)^\diamond E \;=\; F\backslash E \;.$$

**Proof**

$$
\begin{array}{ll}
 & \neg(F\backslash E) \\
\equiv & \qquad \{ \text{ definition 5.74, de Morgan } \} \\
 & \cup.(w:\ w \in F:\ \neg(E_w)) \\
\equiv & \qquad \{ \text{ (5.71) and definition 5.72 } \} \\
 & \partial_F(\neg E) \quad .
\end{array}
$$

□

Hence right factors and derivatives are in a one to one correspondence. The properties of the one can immediately be transcribed into properties of the other. So, formally it doesn't matter which one of the two is studied. In most of the literature concerning regular languages the derivative is defined as is done above, i.e. by explicitly stating its elements. The factor is most easily expressed in the form of the Galois connection, see definition 5.75. This means that proofs involving the derivative will be in terms of elements, while proofs involving factors will be element-free —although it can be done using elements, see the beginning of this section—.

There is also a left factor. The conjugate of the left factor is called the *antiderivative*. The properties of that one are dual to the properties of the derivative, since properties of the left factor are dual to the properties of the right factor.

# Chapter 6

# More Structure in Lattices

The manipulative elegance, exemplified by the Galois connections, and the inherent higher-order possibilities of lattice theory make lattices extremely well suited for a first description of the prominent domain of our interest: the set theoretical relations. A necessary condition is that those relations may be characterised in terms of (properties of) lattices. A first hurdle is the characterisation of powersets.
In this chapter we concentrate on three important lattice properties satisfied by powersets and we present several lattice theoretic characterisations of powersets. The main properties to be discussed are: distributivity, complementation and atomicity. As in the former chapters the chosen treatment is strongly influenced by manipulativity requirements, but the results are completely standard; it is not a study on the frontiers of lattice theory, but merely a rendering of those parts that we foresee to be important in later chapters, in a way that we deem fit for the applications to be expected.

The three mentioned properties are studied but not automatically assumed to hold for the lattices in the sequel. We intend to admit other models than the standard relations, so we want
- to profit from a certain degree of generality
- to pinpoint exactly the reasons why certain rules are valid, useful or necessary.
In particular, we try to avoid complementation as much as possible (in the categorical theory of datatypes it occurs only under heavy topoi assumptions) and we try to refrain from atomicity in order to see to what extent point-free manipulation is possible and useful. We do, however, assume some distributiv-

ity of the lattices in the following chapters (especially universal distributivity) because of its elementary and indispensable manipulative power.

## 6.1 Distributivity

Suprema and infima are dual notions and, with the exception of (3.42) they are independent. In many lattices (finite) suprema and infima are linked by distributivity, for example the predicate calculus in section 2.1.

**Definition 6.1** A lattice $(\mathcal{A}, \sqsubseteq)$ is said to be *distributive* if for all $x, y, z \in \mathcal{A}$

$$x \sqcap (y \sqcup z) \;=\; (x \sqcap y) \sqcup (x \sqcap z) \quad .$$

□

Or, to put it differently and with less dummies, a lattice is called distributive if for every $x \in \mathcal{A}$ the section $(x\sqcap)$ is $\sqcup$-junctive. One might wonder about a "dual" notion of distributivity. In fact the formula in definition 6.1 is equivalent to its dual (see exercise 6.9). Hence in a distributive lattice we have for all $x, y, z \in \mathcal{A}$

$$(6.2) \qquad x \sqcup (y \sqcap z) \;=\; (x \sqcup y) \sqcap (x \sqcup z) \quad .$$

From exercise 3.27(**b**) we know that $(x\sqcap)$ is positively $\sqcap$-junctive for every $x \in \mathcal{A}$. Together with its dual this establishes the following alternative formulation of distribitivity.

**Theorem 6.3** A lattice $(\mathcal{A}, \sqsubseteq)$ is called distributive iff $(x\sqcap)$ and $(x\sqcup)$ are *both* $\sqcap$-junctive and $\sqcup$-junctive for all $x \in \mathcal{A}$.

□

There are many characterisations of distributivity: exercises 6.8 and 6.9 provide a few of them. The definition of distributivity via properties of sectioned suprema and/or infima as presented in theorem 6.3 leads to some immediate generalisations.

**Definition 6.4** A lattice $(\mathcal{A}, \sqsubseteq)$ is called *chain distributive* iff $(x\sqcap)$ and $(x\sqcup)$ are both positively $\sqcup$- and $\sqcap$-continuous for all $x \in \mathcal{A}$.

□

**Definition 6.5** A lattice $(\mathcal{A}, \sqsubseteq)$ is called *universally distributive* iff $(x\sqcap)$ is universally $\sqcup$-junctive and $(x\sqcup)$ is universally $\sqcap$-junctive for all $x \in \mathcal{A}$.

□

As noted earlier, the sections $(x\sqcup)$ and $(x\sqcap)$ are positively $\sqcup$- respectively $\sqcap$-junctive for every $x \in \mathcal{A}$. Since $(x\sqcap)$ is bottom strict and $(x\sqcup)$ is top strict it follows that 6.5 may be given in the same vein as 6.4 and 6.3.

**Theorem 6.6** A lattice $(\mathcal{A}, \sqsubseteq)$ is universally distributive iff $(x\sqcap)$ and $(x\sqcup)$ are both positively $\sqcup$- and $\sqcap$-junctive for all $x \in \mathcal{A}$.

□

A last form of distributivity — that we 'sort of' consider — cannot be given in terms of properties of sectioned $\sqcup$ and $\sqcap$ .

**Definition 6.7** A lattice $(\mathcal{A}, \sqsubseteq)$ is called *completely distributive* iff for all sets $J$ and $K$ and all functions $f \in \mathcal{A} \longleftarrow J \times K$ the following equality and its dual hold

$$\begin{aligned} &\sqcap.(j : \ j \in J : \ \sqcup.(k : \ k \in K : \ f.(j,k))) \ = \\ &\quad \sqcup.(g : \ g \in K \longleftarrow J : \ \sqcap.(j : \ j \in J : \ f.(j, g.j))) \ \ . \end{aligned}$$

□

The definitions given thus far do not assume a complete lattice, the formula in definition 6.7 should be read as: "if the left hand side exists, then the right hand side exists and they are equal".

Several variations on the above definitions are possible, depending on the requirements on the chains, sets and subsets (mostly related to cardinality). For our purposes universal distributivity suffices, which implies distributivity, but is weaker than complete distributivity (for a counter example consider the regular open algebra of the open unit interval. For more details, the interested reader is referred to [46]). Moreover, our domain of interest is the complete lattices, so existence of $\sqcap$ and $\sqcup$ for arbitrary sets is guaranteed.

**Exercise 6.8** Show that the following properties of a lattice are equivalent

**a** $x \sqcap (y \sqcup z) \sqsubseteq (x \sqcap y) \sqcup z$ ,

**b** $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$ ,

**c** $x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap (x \sqcup z)$ ,

**d** $(x \sqcap y) \sqcup (y \sqcap z) \sqcup (z \sqcap x) = (x \sqcup y) \sqcap (y \sqcup z) \sqcap (z \sqcup x)$ .

□

**Exercise 6.9** Show that a lattice is distributive iff for every $x, y$ and $z$ the implication $x \sqcap z \sqsubseteq y \wedge x \sqsubseteq z \sqcup y \Rightarrow x \sqsubseteq y$ holds.
□

**Exercise 6.10** Assume $(\mathcal{A}, \sqsubseteq)$ is a distributive and complete lattice. Show, under assumption of the axiom of choice, that chain distributivity and universal distributivity coincide
*Hint*: The axiom of choice allows a well-ordering of any subset $\mathcal{S}$ of $\mathcal{A}$, say $\mathcal{S} = \{S_\alpha \mid \alpha \text{ ordinal}, \alpha < \nu\}$ . Define $\Sigma_\beta = \sqcup.(\alpha : \alpha < \beta : S_\alpha)$, then $\{\Sigma_\beta \mid \beta \leq \nu\}$ is a chain. Transfinite induction does the job.
□

## 6.2 Complements

From the chapter on Galois connections, it may be clear that definition 6.5 of universal distributivity has promising consequences. By using theorem 5.41 and 5.42 we can define universal distributivity in another way.

**Theorem 6.11** A lattice $(\mathcal{A}, \sqsubseteq)$ is universally distributive iff $(x\sqcap)^\sharp$ and $(x\sqcup)^\flat$ exist for all $x \in \mathcal{A}$
□
Thus for $(\mathcal{A}, \sqsubseteq)$ a universally distributive lattice we have for all $x, y, z \in \mathcal{A}$

$$\begin{array}{rcl} x \sqcap y \sqsubseteq z & \equiv & y \sqsubseteq (x\sqcap)^\sharp.z \\ (x\sqcup)^\flat.y \sqsubseteq z & \equiv & y \sqsubseteq x \sqcup z \end{array}$$

These equations are generalisations of the equations in the definition of pseudo-complements and pseudo-supplements below.

**Definition 6.12** In a lattice $(\mathcal{A}, \sqsubseteq)$ we say $x$ has *pseudo-complement* $pc.x$ iff for all $y \in \mathcal{A}$ we have

(6.13) $x \sqcap y = \perp\!\!\perp \equiv y \sqsubseteq pc.x$ .

Dually, we call $ps.x$ the *pseudo-supplement* of $x$ iff for all $y \in \mathcal{A}$

(6.14) $ps.x \sqsubseteq z \equiv \top\!\!\top = x \sqcup z$ ,

holds.

□

Clearly, by indirect equality, pseudo-complements and pseudo-supplements are unique if they exist. From theorem 6.11 it follows that they do indeed exist (uniquely) in the case of a universally distributive lattice. To be precise

$$pc.x \;=\; (x\sqcap)^\sharp.\bot\!\!\bot \quad \text{and} \quad ps.x \;=\; (x\sqcup)^\flat.\top\!\!\top$$

As an example calculation, we show that a pseudo-complement is at most the pseudo-supplement in a universally distributive lattice. I.e. we show for every $x \in \mathcal{A}$

(6.15) $(x\sqcap)^\sharp.\bot\!\!\bot \;\sqsubseteq\; (x\sqcup)^\flat.\top\!\!\top$ .

$$\begin{array}{ll}
 & (x\sqcap)^\sharp.\bot\!\!\bot \\
= & \qquad \{ \text{ from (6.14): } \top\!\!\top \;=\; x \sqcup (x\sqcup)^\flat.\top\!\!\top \;\} \\
 & ((x\sqcup)^\flat.\top\!\!\top \sqcup x) \sqcap (x\sqcap)^\sharp.\bot\!\!\bot \\
\sqsubseteq & \qquad \{ \text{ distribution, calculus } \} \\
 & (x\sqcup)^\flat.\top\!\!\top \sqcup (x \sqcap (x\sqcap)^\sharp.\bot\!\!\bot) \\
= & \qquad \{ \text{ form (6.13): } x \sqcap (x\sqcap)^\sharp.\bot\!\!\bot \;=\; \bot\!\!\bot \;\} \\
 & (x\sqcup)^\flat.\top\!\!\top \quad .
\end{array}$$

If the pseudo-complement and the pseudo-supplement of $x$ coincide, then $x$ has a complement. Complements can be defined in any bounded lattice, i.e. a lattice having a $\bot\!\!\bot$ and $\top\!\!\top$.

**Definition 6.16** For a bounded lattice $(\mathcal{A}, \sqsubseteq)$ we call $x'$ *a complement* of $x$ if

$$x \sqcap x' \;=\; \bot\!\!\bot \quad \text{and} \quad x \sqcup x' \;=\; \top\!\!\top \quad .$$

A lattice $(\mathcal{A}, \sqsubseteq)$ is said to be *complemented* if every $x \in \mathcal{A}$ has a complement.

□

Unlike the situation with pseudo-complements one cannot deduce the unicity of complements from their definition; however, in a distributive lattice they are unique (see exercise 6.30).

From now on distributivity of the lattices under consideration will be assumed, hence complements — when they exist — are unique.

Whenever the complement of $x$ exists, it will be denoted, as usual, by $\neg x$ ; in that case the complement of $\neg x$ exists too and $\neg\neg x \;=\; x$ .

If the (distributive) lattice is complemented, the complements are just the pseudo-complements and the pseudo-supplements. In such a lattice, a stronger type of distributivity holds .

**Theorem 6.17** A complemented distributive lattice is universally distributive. In particular $(x\sqcap)^\sharp.\perp\!\!\perp \;=\; ((\neg x)\sqcup)$ and $(x\sqcup)^\flat.\top\!\!\top \;=\; ((\neg x)\sqcap)$ .

**Proof** By theorem 6.11 it is sufficient to show that $x \sqcap y \sqsubseteq z \;\equiv\; y \sqsubseteq \neg x \sqcup z$ for all $x, y$ and $z$, together with its dual.

$$
\begin{array}{ll}
 & x \sqcap y \sqsubseteq z \\
\Rightarrow & \quad \{\ ((\neg x)\sqcup) \text{ is monotonic } \} \\
 & \neg x \sqcup (x \sqcap y) \sqsubseteq \neg x \sqcup z \\
\equiv & \quad \{\ \text{distribution } \} \\
 & (\neg x \sqcup x) \sqcap (\neg x \sqcup y) \sqsubseteq \neg x \sqcup z \\
\equiv & \quad \{\ \neg x \sqcup x \;=\; \top\!\!\top \ \} \\
 & \neg x \sqcup y \sqsubseteq \neg x \sqcup z \\
\equiv & \quad \{\ \text{suprema } \} \\
 & y \sqsubseteq \neg x \sqcup z \\
\Rightarrow & \quad \{\ (x\sqcap) \text{ is monotonic } \} \\
 & x \sqcap y \sqsubseteq \neg x \sqcap (x \sqcup z) \\
\equiv & \quad \{\ \text{similar to the above steps } \} \\
 & x \sqcap y \sqsubseteq z \quad .
\end{array}
$$

The proof of the dual is left to the reader.
□

**Corollary 6.18** A universally distributive lattice $(\mathcal{A}, \sqsubseteq)$ is complemented iff $(x\sqcup)^\flat.\top\!\!\top \;\sqsubseteq\; (x\sqcap)^\sharp.\perp\!\!\perp$ for every $x \in \mathcal{A}$ .

**Proof** If the lattice is complemented then, by theorem 6.17, $(x\sqcap)^\sharp.\perp\!\!\perp \;=\; \neg x \sqcup \perp\!\!\perp \;=\; \neg x \;=\; \neg x \sqcap \top\!\!\top \;=\; (x\sqcup)^\flat.\top\!\!\top$ .

Conversely, assume $(x\sqcup)^\flat.\top\!\!\top \;\sqsubseteq\; (x\sqcap)^\sharp.\perp\!\!\perp$ for every $x \in \mathcal{A}$. Then, by (6.15), the two are equal, say $x' \;=\; (x\sqcap)^\sharp.\perp\!\!\perp \;=\; (x\sqcup)^\flat.\top\!\!\top$ , and

$$
\begin{array}{l}
x \sqcap x' \;=\; x \sqcap (x\sqcap)^\sharp.\perp\!\!\perp \;\sqsubseteq\; \perp\!\!\perp \quad , \\
x \sqcup x' \;=\; x \sqcup (x\sqcup)^\flat.\top\!\!\top \;\sqsupseteq\; \perp\!\!\perp \quad .
\end{array}
$$

So $x' \;=\; \neg x$ .

□

In fact, the proof of theorem 6.17 gives the so-called *shunting rule*, the major manipulative tool in complemented distributive lattices,

(6.19) $x \sqcap y \sqsubseteq z \quad \equiv \quad y \sqsubseteq \neg x \sqcup z \quad ,$

and, since $\neg\neg x \;=\; x$ , this equivales

(6.20) $\neg x \sqcap y \sqsubseteq z \quad \equiv \quad y \sqsubseteq \; x \sqcup z \quad .$

Each of the two forms of the shunting rule will occur in calculations where complementation is essential.

As mentioned, $\neg$ is its own inverse; but shunting shows even more:

(6.21) $\neg x \sqsubseteq y \quad \equiv \quad x \sqsupseteq \; \neg y \quad .$

So $\neg$ is its own adjoint in a Galois connection with one ordering reversed. Since the supremum is the infimum in the reversed ordering and vice-versa, this together with the junctivity type for upper and lower adjoints immediately establishes the "De Morgan" laws

**Theorem 6.22** For a complemented distributive lattice $(\mathcal{A}, \sqsubseteq)$, we have for all $S \subseteq \mathcal{A}$

$$\neg(\sqcap.S) \;=\; \sqcup.\neg S \quad \text{and} \quad \neg(\sqcup.S) \;=\; \sqcap.\neg S \quad .$$

□

Note that $\neg$ is monotonic from $\sqsubseteq$ to $\sqsupseteq$ (also called anti monotonic). The order reversal leads to a second 1-1 correspondence between $\sqcap$-junctive and $\sqcup$-junctive endofunctions after the Galois connection: the *conjugate*. Define for any endofunction $f$ on a complemented lattice its conjugate $f^\diamond$ by

(6.23) $f^\diamond.x \;=\; \neg(f.\neg x) \quad$ or $\quad f^\diamond \;=\; \neg \bullet f \bullet \neg$

A few properties of the conjugate are

(6.24) $\quad f^{\diamond\diamond} \;=\; f$

(6.25) $\quad f^\diamond \;\dot{\sqsubseteq}\; g \quad \equiv \quad f \;\dot{\sqsupseteq}\; g^\diamond$

(6.26) $\quad f$ is $S$-$\sqcup$-junctive $\quad \equiv \quad f^\diamond$ is $(\neg S)$-$\sqcap$-junctive

It may be seen from (6.26) that the negation gives a 1-1 correspondence for functions of any type of $\sqcap$- or $\sqcup$-junctivity, and not only for functions that are universally $\sqcap$- or $\sqcup$-junctivity (as is the case for Galois connections).

The link between conjugates and Galois adjoints is given by

(6.27) $\quad f^{\diamond\sharp} \;=\; f^{\flat\diamond} \qquad$ for universally $\sqcap$-junctive functions $f$ ,

(6.28) $\quad f^{\sharp\diamond} \;=\; f^{\diamond\flat} \qquad$ for universally $\sqcup$-junctive functions $f$ .

We only prove (6.27). First note that for universally $\sqcap$-junctive $f$, $f^{\diamond}$ is universally $\sqcup$-junctive so $f^{\flat}$ and $f^{\diamond\sharp}$ exist. The equality $f^{\diamond\sharp} \;=\; f^{\flat\diamond}$ follows by unicity of adjoints from the following derivation.

$$
\begin{array}{ll}
 & f^{\diamond}.x \sqsubseteq y \\
\equiv & \qquad \{ \text{ (6.23) and (6.21) } \} \\
 & f.\neg x \sqsupseteq \neg y \\
\equiv & \qquad \{ \text{ } f \text{ is universally } \sqcap\text{-junctive } \} \\
 & \neg x \sqsupseteq f^{\flat}.\neg y \\
\equiv & \qquad \{ \text{ (6.21) and (6.23) } \} \\
 & x \sqsubseteq f^{\flat\diamond}.y \quad .
\end{array}
$$

Which proves that $f^{\flat\diamond}$ is the upper adjoint of $f^{\diamond}$ .

If $(\mathcal{A}, \sqsubseteq)$ is a complete complemented lattices, the adjoints of the $\sqcap$ and $\sqcup$ sections are conjugates. I.e. for every $x, y \in \mathcal{A}$ we have

(6.29) $\quad (x\sqcap)^{\sharp} \;=\; (x\sqcap)^{\diamond} \quad$ and $\quad (x\sqcup)^{\flat} \;=\; (x\sqcup)^{\diamond}$

**Some Examples** A powerset is a complete, completely distributive and complemented lattice.

The equivalence relations on a set form a complete lattice whose infima are just the infima in the powerset lattice of the square of the set; the suprema however differ (in general) from the suprema in the lattice of all relations. We will denote the suprema and infima in the lattice of equivalence relations by $\vee$ , respectively $\wedge$ .

The complete lattice of all equivalence relations on $\mathcal{X}$ is not distributive: Let $A$ be a proper subset of $\mathcal{X}$ (which means that $A$ is non-empty and differs from $\mathcal{X}$). Take two distinct elements of $A$, say $a_0$ and $a_1$, and two distinct elements outside the set $A$, say $b_0$ and $b_1$ . Then define

$$
\begin{aligned}
E &= A \times A \sqcup \neg A \times \neg A \quad , \\
F &= I \sqcup \{(a_0, b_0),\, (b_0, a_0)\} \quad , \\
G &= I \sqcup \{(a_1, b_1),\, (b_1, a_1)\} \quad .
\end{aligned}
$$

Hence, we have $E \;\vee\; F \;=\; \mathcal{X} \times \mathcal{X} \;=\; E \;\vee\; G$ and $F \;\wedge\; G \;=\; I$ . And

$$
\begin{array}{ll}
 & E \ \vee \ (F \ \wedge \ G) \\
= & \quad \{ \ F \ \wedge \ G = I \ \} \\
 & E \ \vee \ I \\
= & \quad \{ \ \text{definition of } E \ \} \\
 & E \quad ,
\end{array}
$$

while $(E \ \vee \ F) \ \wedge \ (E \ \vee \ G) \ = \ \mathcal{X} \times \mathcal{X} \neq \ E$ .

Since the suprema of chains are the suprema in the lattice of all relations, on half of the chain distributivity is satisfied: for any equivalence relation $E$ on $\mathcal{X}$ and any chain $\mathcal{C}$ we have $E \sqcap (\vee \mathcal{C}) \ = \ \vee \ (E \ \wedge \ \mathcal{C})$ . The other half fails with a counter example in the same vein as above.

Every finite lattice is complete and chain distributive, but there are non-distributive finite complemented lattices.

Every complete chain is universally distributive but there are non-complemented complete chains (see also exercise 6.31).

Let $L$ be a complete chain such that $\top\!\!\top \ = \ \sqcup.(\mathcal{L}\backslash\{\top\!\!\top\})$ and let $k \notin \mathcal{L}$ . Define $\mathcal{K} \ = \ \mathcal{L} \cup \{k\}$ and $\sqsubseteq_{\mathcal{K}} \ = \ \sqsubseteq_{\mathcal{L}} \cup \{(\bot\!\!\bot, k), \, (k, \top\!\!\top), \, (k, k)\}$ . Then $\mathcal{K}$ is distributive, but not chain distributive for $k \sqcap \sqcup.(\mathcal{L}\backslash\{\top\!\!\top\}) \ = \ k$ and $\sqcup.(k \sqcap (\mathcal{L}\backslash\{\top\!\!\top\})) \ = \ \bot\!\!\bot$ .

**Exercise 6.30** Show that complements in a distributive lattice are unique.
□

**Exercise 6.31** Show that $I\!N \cup \{\infty\}$ with the usual ordering is pseudo-complemented and pseudo-supplemented, but not complemented.
□

**Exercise 6.32** Show that conjugation dualises the junctivity type, i.e. prove (6.26).
□

**Exercise 6.33** Let $(\mathcal{A}, \sqsubseteq)$ be a universally distributive lattice. Prove for all $x, y, z \in \mathcal{A}$

**a** $(x\sqcup)^\flat.y \ \sqsubseteq \ y \ \sqsubseteq \ (x\sqcap)^\sharp.y$ ,

**b** $(x\sqcap)^\sharp.y \ = \ y \ \sqcup \ (x\sqcup)^\flat.(x\sqcap)^\sharp.y$ ,

**c** $(x\sqcap)^\sharp.y \ \sqcap \ (z\sqcap)^\sharp.y \ = \ ((x \ \sqcup \ z)\sqcap)^\sharp.y$ .

(*Hint for part* **b**: use part **a** to generalise the proof of (6.15).)
□

## 6.3 Atoms

The prominent model in this study is the lattice of all relations on some universe. It is to be expected that eventually all properties of that lattice will be needed. To that end a complete characterisation of powerset lattices is required, not only to be able to use that full structure, but also to pinpoint exactly the reason why. The only feature of powerset lattices that has not yet been discussed is the fact that the lattice members are built up from points (or elements). The lattice theoretic concept that corresponds to a point is the notion of an atom.

**Definition 6.34** For a lattice $(\mathcal{A}, \sqsubseteq)$ we call $a \in \mathcal{A}$ an *atom* if it has no proper subelement, i.e. for every $x \in \mathcal{A}$

$$x \sqsubseteq a \quad \equiv \quad x = a \ \vee \ x = \perp\!\!\perp \ .$$

A lattice is called *atomic* if every proper lattice element contains a proper atom.
□

Atoms may or may not exist in bounded lattices. Since we are mainly (and for the calculational model only) interested in complete universally distributive lattices, we discuss atoms only in that realm. Complementation is not required, but it will pop up!

For the remainder of this section, let $(\mathcal{A}, \sqsubseteq)$ be a complete and universally distributive lattice. The atoms will be denoted by lowercase letters from the beginning of the alphabet. The set of atoms is denoted by $I\!\!A$.

Clearly $\perp\!\!\perp \in I\!\!A$, but that may be the only one, see for instance $(I\!\!N \cup \{\infty\}, \geq)$. This trivial atom is often excluded from the atoms, but we won't. Instead we refer to non-trivial atoms or proper atoms to exclude $\perp\!\!\perp$ from our considerations.

The two most prominent properties of atoms are that they are *disjoint*

(6.35) $\quad a = b \ \vee \ a \sqcap b = \perp\!\!\perp \ ,$

and *irreducible*

(6.36) $\quad a \sqsubseteq x \sqcup y \quad \equiv \quad a \sqsubseteq x \ \vee \ a \sqsubseteq y \ .$

Irreducible elements of a lattice do not need to be atoms, e.g. every element of $(I\!\!N \cup \{\infty\}, \geq)$ is irreducible. However, if a lattice is complemented, atoms and irreducible elements coincide, see exercise 6.46.

We can express atomicity of a lattice in another way.

**Theorem 6.37** A lattice $(\mathcal{A}, \sqsubseteq)$ is atomic iff for every $x \in \mathcal{A}$

$$\sqcup.\mathbb{A} \sqcap x = \perp\!\!\perp \quad \equiv \quad x = \perp\!\!\perp \ .$$

**Proof** First observe that

$$\exists.(a: \ a \neq \perp\!\!\perp : \ a \sqsubseteq x) \ \equiv \ \sqcup.(a: \ a \sqsubseteq x: \ a) \neq \perp\!\!\perp \ ,$$

and

$$\begin{array}{ll} & \sqcup.(a: \ a \sqsubseteq x: \ a) \\ = & \qquad \{ \ a \sqcap x = a \ \vee \ a \sqcap x = \perp\!\!\perp \ \} \\ & \sqcup.(a :: \ a \sqcap x) \\ = & \qquad \{ \ \text{the lattice is universally distributive} \ \} \\ & \sqcup.\mathbb{A} \sqcap x \quad . \end{array}$$

So

$$\begin{array}{ll} & x \neq \perp\!\!\perp \Rightarrow \exists.(a: \ q \neq \perp\!\!\perp : \ a \sqsubseteq x) \\ \equiv & \qquad \{ \ \text{above} \ \} \\ & x \neq \perp\!\!\perp \Rightarrow \sqcup.\mathbb{A} \sqcap x \neq \perp\!\!\perp \\ \equiv & \qquad \{ \ \text{calculus} \ \} \\ & x = \perp\!\!\perp \quad \equiv \quad \sqcup.\mathbb{A} \sqcap x = \perp\!\!\perp \quad . \end{array}$$

□

This second "definition" of atomicity may be rephrased, suppressing the dummy $x$, in terms of adjoints as follows:

**Corollary 6.38** A lattice is atomic iff $(\sqcup.\mathbb{A}\sqcap)^\sharp$ is bottom-strict.

**Proof** First, note that the assumptions on the lattice guarantee the existence of $(\sqcup.\mathbb{A}\sqcap)^\sharp$ .

$$\begin{array}{ll} & \sqcup.\mathbb{A} \sqcap x = \perp\!\!\perp \quad \equiv \quad x = \perp\!\!\perp \\ \equiv & \qquad \{ \ (6.13), \text{calculus} \ \} \\ & x \sqsubseteq (\sqcup.\mathbb{A}\sqcap)^\sharp.\perp\!\!\perp \quad \equiv \quad x \sqsubseteq \perp\!\!\perp \\ \equiv & \qquad \{ \ \text{indirect equality} \ \} \\ & (\sqcup.\mathbb{A}\sqcap)^\sharp.\perp\!\!\perp = \perp\!\!\perp \quad . \end{array}$$

□

Atomicity of a lattice does not sufficiently capture a powerset-like behaviour with respect to "points", e.g. $(\mathbb{N} \cup \{\infty\}, \leq)$ is atomic with atoms $\{0, 1\}$. For a powerset structure it would be desirable if every lattice element is built up by the atoms it contains.

**Definition 6.39** A lattice $(\mathcal{A}, \sqsubseteq)$ is called *saturated* iff for every $x \in \mathcal{A}$ $x \;=\; \sqcup.(a :\; a \sqsubseteq x :\; a)$ .

□

Again we can give an equivalent formulation.

**Theorem 6.40** A lattice $(\mathcal{A}, \sqsubseteq)$ is saturated iff for every $x \in \mathcal{A}$ we have $x \;=\; \sqcup.\mathbb{A} \sqcap x$ .

**Proof** For any $x \in \mathcal{A}$ we derive

$$\begin{array}{ll} & \sqcup.\mathbb{A} \sqcap x \\ = & \quad \{\ \mathcal{A} \text{ is universally distributive }\ \} \\ & \sqcup.(a ::\; a \sqcap x) \\ = & \quad \{\ a \sqcap x \;=\; a \;\vee\; a \sqcap x \;=\; \perp\!\!\perp\ \} \\ & \sqcup.(a :\; a \sqsubseteq x :\; a) \quad . \end{array}$$

□

Using indirect equality (6.40) may be reformulated as

$$\sqcup.\mathbb{A} \sqcap x \sqsubseteq y \quad \equiv \quad x \sqsubseteq y \quad ,$$

or in terms of adjoints

(6.41) $(\sqcup.\mathbb{A}\sqcap)^\sharp \;=\; I_\mathcal{A}$

There are various equivalent formulations of saturation, some of them are given in the next lemma (the proof can be found as exercise 6.47).

**Lemma 6.42** Equivalent are

**a** $\mathcal{A}$ is saturated,

**b** $\sqcup.\mathbb{A} \;=\; \top\!\!\top$ ,

**c** $x \sqsubseteq y \quad \equiv \quad \forall(a :\; a \sqsubseteq x :\; a \sqsubseteq y)$ .

□

In some treatments atomicity is defined as "our" saturation. In case the lattice is complemented, there is no difference (see theorem 6.43); but without complementation there are examples of non-saturated atomic lattices (e.g. $(I\!N \cup \{\infty\}, \leq)$). The full power of saturation may be seen from

**Theorem 6.43** For $\mathcal{A}$ a complete universally distributive lattice, the following are equivalent

**a** $\mathcal{A}$ is saturated,

**b** $\mathcal{A}$ is atomic and complemented,

**c** $\mathcal{A}$ is isomorphic to a powerset.

**Proof**
**a** $\Rightarrow$ **c**: Define $\varphi \in I\!P I\!A \longleftarrow \mathcal{A}$ by

$$\varphi.x \;=\; \{a \mid a \sqsubseteq x\} \quad .$$

Then $\varphi$ is surjective, for $\mathcal{B} \sqsubseteq I\!A$ we observe

$$\begin{array}{ll}
 & \varphi.\sqcup.\mathcal{B} \\
= & \qquad \{ \text{ definition } \varphi \ \} \\
 & \{a \mid a \sqsubseteq \sqcup.\mathcal{B}\} \\
= & \qquad \{ \ (6.35) \ \} \\
 & \mathcal{B} \quad .
\end{array}$$

With the use of lemma 6.42(**c**) we conclude that $\varphi$ is an order isomorphism.
**c** $\Rightarrow$ **b**: Immediate.
**b** $\Rightarrow$ **a**:

$$\begin{array}{ll}
 & \sqcup.I\!A \;=\; \top\!\!\top \\
\equiv & \qquad \{ \text{ complements } \} \\
 & \neg.\sqcup.I\!A \;=\; \bot\!\!\bot \\
\equiv & \qquad \{ \text{ atomicity, (6.37) } \} \\
 & \sqcup.I\!A \sqcap \neg.\sqcup.I\!A \;=\; \bot\!\!\bot \\
\equiv & \qquad \{ \text{ complements } \} \\
 & \mathit{true} \quad .
\end{array}$$

By lemma 6.42(**b**) it follows that $\mathcal{A}$ is saturated.
□

Knowing that saturation implies complementation raises the question whether some additional property, not in terms of atoms, may be found to guarantee saturation for complemented (complete) lattices. Indeed, complete distributivity does the job.

**Theorem 6.44** A complete complemented distributive lattice is saturated iff it is completely distributive.

**Proof**
$\Rightarrow$: From theorem 6.17 the lattice is universally distributive. So by theorem 6.43 it is isomorphic to a powerset, which is completely distributive.
$\Leftarrow$: Define $f \in \mathcal{A} \longleftarrow \mathcal{A} \times \mathit{2}$ by

$$f.(x,0) \;=\; x \quad \text{and} \quad f.(x,1) \;=\; \neg x \quad ,$$

and for $\varepsilon \in \mathit{2} \longleftarrow \mathcal{A}$ define $a_\varepsilon$ by

$$a_\varepsilon \;=\; \sqcap.(x :\; x \in \mathcal{A} :\; f.(x,\varepsilon.x)) \quad .$$

Then $a_\varepsilon$ is an atom for every $\varepsilon$, since

$$\begin{array}{ll}
 & y \sqsubseteq a_\varepsilon \\
\Rightarrow & \qquad \{ \text{ definition } a_\varepsilon \ \} \\
 & y \sqsubseteq f.(y,\varepsilon.y) \sqcap f.(\neg y,\varepsilon.\neg y) \\
\Rightarrow & \qquad \{ \text{ calculus, definition of } f \ \} \\
 & y \sqsubseteq \neg y \;\vee\; (\varepsilon.y \;=\; 0 \;\wedge\; \varepsilon.\neg y \;=\; 1) \\
\Rightarrow & \qquad \{ \ y \sqsubseteq \neg y \Rightarrow y \;=\; \perp\!\!\perp, \text{ definition } a_\varepsilon \ \} \\
 & y \;=\; \perp\!\!\perp \;\vee\; a_\varepsilon \sqsubseteq y \quad .
\end{array}$$

Finally

$$\begin{array}{ll}
 & \sqcup.\mathbb{A} \\
\sqsupseteq & \qquad \{ \ a_\varepsilon \in \mathbb{A} \ \} \\
 & \sqcup.(\varepsilon :\; \varepsilon \in \mathit{2} \longleftarrow \mathcal{A} :\; \sqcap.(x :\; x \in \mathcal{A} :\; f.(x,\varepsilon.x))) \\
= & \qquad \{ \ (6.7) \ \} \\
 & \sqcap.(x :\; x \in \mathcal{A} :\; \sqcup.(k :\; k \in \mathit{2} :\; f.(x,k))) \\
= & \qquad \{ \ f.(x,0) \sqcup f.(x,1) \;=\; \top\!\!\top \ \} \\
 & \sqcap.(x :\; x \in \mathcal{A} :\; \top\!\!\top) \\
= & \qquad \{ \text{ calculus } \} \\
 & \top\!\!\top \quad .
\end{array}$$

$\square$

Complete distributivity is not of any use in the sequel, the notion and theorem 6.44 are only mentioned here for completeness.

**Exercise 6.45** Let $p \in \mathcal{A} \longleftarrow \mathcal{A}$ be such that $p.x \;=\; x \;\vee\; p.x \;=\; (x\sqcap)^\sharp.\bot\!\!\bot$ . Show that $\sqcap.(x:\; x \in \mathcal{A}:\; p.x)$ is an atom.
*Don't cheat by copying a part of the proof of theorem* 6.44!

$\square$

**Exercise 6.46**

**a** $p \in \mathbb{A}$ ,

**b** $p$ is irreducible,

**c** $p \;=\; x \sqcup y \Rightarrow p \;=\; x \;\vee\; p \;=\; y$ .

Show that **b** and **c** are equivalent and that the are implied by **a**. Furthermore, show that the three are equivalent if the lattice is complemented.

$\square$

**Exercise 6.47** Prove lemma 6.42.

$\square$

# Chapter 7

# Closure Operators and Fixed Points

So-called "closure operators" form an extremely important class of functions in mathematics and computing science since many problems can be expressed in terms of such operators. So-called "fixed points" of functions are just as important. Not surprisingly, since there is a close relationship between fixed points and closure operators allowing problems expressed in terms of the one always to be reformulated in terms of the other! In this chapter we lay bare the connection and explore its ramifications.

We do not know to whom the theory to be presented in this section should be credited. Probably to either Albert Tarski or to S.C. Kleene. The material presented differs from that typically to be found in computing science texts in that we do not assume a so-called "cpo" structure, nor that the functions we consider are continuous. Instead we assume monotonicity only of our functions and a complete lattice structure.

We begin with the definition and a short discussion of closure operators. Then we need to digress awhile to introduce so-called "prefix points", a key element in a famous fixed-point theorem due to Tarski. This digression then allows us to observe a rather special Galois connection defining a closure operator for each monotonic function as well as the function's least fixed points. Applications of these results are considered later in the monograph.

Dualisation of the theorems presented here to so-called "interior operators" and greatest fixed points is not explicitly discussed but will be made use of later. (We assume that by now the reader has become completely familiar with the process of dualisation.)

In the course of the previous chapters we have been preparing the reader for a switch to point-free proofs in preference to pointwise proofs (i.e. proofs at the level of function compositions rather than function applications). In this chapter we take the bull by the horns and conduct all proofs at the point-free level.

There are two advantages in doing so. One is that the proofs are often more compact. (This is not always the case, however.) The other is that in later chapters we will be able to abstract from the calculations in this chapter to "compositions" that are not necessarily function compositions. (In fact some of the models we consider in later chapters have a binary "composition" that has nothing whatsoever to do with function composition.)

In order that the switch should not come as a profound shock let us briefly summarise some calculation rules that will be most prominent in the following pages.

Suppose $(\mathcal{A}, \sqsubseteq)$ is a complete lattice. Our primary concern will be the complete lattice of monotonic endofunctions on $\mathcal{A}$ ordered by the relation $\dot{\sqsubseteq}$. Recal from definition 4.12, for $f, g \in \mathcal{A} \longleftarrow \mathcal{A}$,

$$f \ \dot{\sqsubseteq} \ g \quad \equiv \quad \forall(x :: \ f.x \ \sqsubseteq \ g.x) \ \ .$$

We call this lattice $MONO.\mathcal{A}$ . It forms a monoid $(MONO.\mathcal{A}, \bullet, I_{\mathcal{A}})$ where $\bullet$ denotes function composition and $I_{\mathcal{A}}$ is the identity function on $\mathcal{A}$. Moreover, for each function $f$, the function $(\bullet f)$ is universally $\sqcup$-junctive (see 4.18) and so has an upper adjoint $(\bullet f)^\sharp$. The existence of an upper adjoint will not be exploited in the text of this chapter, although you will find it vital to answering some of the exercises. A consequence of its existence will, however, be used extensively, namely that $(\bullet f)$ is monotonic with respect to the pointwise ordering $\dot{\sqsubseteq}$ (see (4.16)). Another monotonicity property that will be used extensively is that for all monotonic endofunctions $f, g$ and $h$

$$f \bullet g \ \dot{\sqsubseteq} \ f \bullet h \quad \Leftarrow \quad g \ \dot{\sqsubseteq} \ h \ \ .$$

(See (4.17).) Recall, however, that $(f\bullet)$ is *not* universally $\dot{\sqcup}$-junctive, in general.

## 7.1 Closure Operators

**Definition 7.1** For $f$ an endofunction on a poset $(\mathcal{A}, \sqsubseteq)$, we call $f$ a *closure* *operator* iff $g \mathrel{\dot{\sqsubseteq}} f \bullet h \;\equiv\; f \bullet g \mathrel{\dot{\sqsubseteq}} f \bullet h$ , for all functions $g$ and $h$ with range $\mathcal{A}$.

□

With this definition one can quickly establish some properties of closure operators.

**Corollary 7.2** For $f$ a closure operator we have

**a** $f$ is reflexive, i.e. $I_{\mathcal{A}} \mathrel{\dot{\sqsubseteq}} f$ ,

**b** $f$ is idempotent, i.e. $f = f \bullet f$ ,

**c** $f$ is monotonic, i.e. $g \mathrel{\dot{\sqsubseteq}} h \Rightarrow f \bullet g \mathrel{\dot{\sqsubseteq}} f \bullet h$ , for all functions $g$ and $h$ with range $\mathcal{A}$ .

**Proof** Part **a** is obtained by instantiating both $g$ and $h$ to $I_{\mathcal{A}}$ in definition 7.1.

Part **b** is proven by mutual containment. The containment $f \mathrel{\dot{\sqsubseteq}} f \bullet f$ follows directly from definition 7.1 with $g$ and $h$ both instantiated to $f$ . The other containment, $f \bullet f \mathrel{\dot{\sqsubseteq}} f$ , is then obtained by instantiating $g$ to $f$ and $h$ to $I_{\mathcal{A}}$ in the definition.

For part **c** we observe the following

$$
\begin{array}{ll}
 & f \bullet g \mathrel{\dot{\sqsubseteq}} f \bullet h \\
\equiv & \qquad \{ \text{ definition 7.1 } \} \\
 & g \mathrel{\dot{\sqsubseteq}} f \bullet h \\
\Leftarrow & \qquad \{ \text{ } \mathbf{a}\text{, monotonicity (4.16), transitivity } \} \\
 & g \mathrel{\dot{\sqsubseteq}} h \quad .
\end{array}
$$

□

As a dual to the closure operators, we have the following

**Definition 7.3** For $f$ an endofunction on a poset $(\mathcal{A}, \sqsubseteq)$, we call $f$ a *co-closure* or an *interior operator* iff $f \bullet g \mathrel{\dot{\sqsubseteq}} h \;\equiv\; f \bullet g \mathrel{\dot{\sqsubseteq}} f \bullet h$ , for all functions $g$ and $h$ with range $\mathcal{A}$.

□

In [42] an interior operator is called a *kernel*. For $f$ a closure or an interior operator, we call an element $z$ *closed* iff $f.z = z$ . It is common practice to refer to the closed elements of interior operators as *open* elements. We don't adhere to existing practice, because the closed and open elements are defined by the same equation.

The ceiling function, see section 5.2.1, can be seen as a closure operator. Consider the ceiling function as a function from reals to reals. We then have $x \leq \lceil y \rceil \equiv \lceil x \rceil \leq \lceil y \rceil$, which is the pointwise version of definition 7.1 with $f$ instantiated to the ceiling function. The closed elements in this case are precisely the integers. The floor function can in the same way be seen as a co-closure operator.

In fact, every Galois connection gives rise to a closure and an interior operator.

**Theorem 7.4** If $(F, G)$ is a Galois connection then

**a** $G \bullet F$ is a closure operator,

**b** $F \bullet G$ is an interior operator.

**Proof** Part **a** follows directly from the equivalence of corollary 5.15(**b**) and 5.15(**c**). Part **b** is the dual.

□

This gives a constructive way of defining a closure or co-closure operator which will be used shortly to construct a closure operator for every monotonic function and subsequently its least fixed point.

**Exercise 7.5** Prove the converse of corollary 7.2, namely that if $f$ satisfies 7.2(**a**), (**b**) and (**c**) then $f$ is a closure operator.

□

**Exercise 7.6** Let $f$ be an arbitrary closure operator over the poset $(\mathcal{A}, \sqsubseteq)$. Give a Galois connection such that $f = G \bullet F$ .

*Hint*: define a Galois connection between $\mathcal{A}$ and the closed elements of the closure operator.

□

## 7.2 Prefix Points

Tarski's fixed point theorem [89] —exercise 7.26 in this chapter— is a mainstay of programming language semantics. How the theorem was actually discovered is not for us to say, but we can speculate on a scenario that might have been the inspiration for the discovery of the theorem.

Apart from his work on fixed points, Tarski is also well known for his work on the calculus of relations [88] where closure operators (transitive closure, symmetric closure etc.) play a prominent rôle. The identification of an abstract notion of closure operator and of closed elements will therefore have been one of the earliest endeavours in developing the calculus. What is also likely to have been observed at an early stage is that the closed elements in the examples we have quoted form complete lattices in which the infima coincide with the infima in the parent lattice. So, for example, the closed elements of the transitive closure operator are the transitive relations (in the calculus of relations) and the infimum of a set of transitive relations is transitive. One is led to speculate that the closed elements of a given closure operator form a complete lattice for all closure operators. (This might even be regarded as a healthiness requirement on the combined notions of closure operator and closed element.) Indeed this is the case. If one proceeds to prove this theorem a surprise is in store! If the given closure operator is $f$ then the only fact that is needed in the proof about the closed elements is that closed element $x$ satisfies the inclusion $f.x \sqsubseteq x$ and $f$ is monotonic. Other properties of $f$ do not enter the picture. This suggests a further abstraction. Consider, for *arbitrary* monotonic endofunction $f$ (thus not necessarily a closure operator) the set of elements $x$ such that $f.x \sqsubseteq x$. These are known as the *prefix points* of $f$. Is this class worthy of study, and if so what are its characteristic properties? Indeed it is, as witnessed by this chapter.

The first fact about prefix points is the one that may have been stumbled on when trying to prove that closed elements form a complete lattice. We call it the prefix lemma.

**Lemma 7.7 (Prefix lemma)** Let $(\mathcal{A}, \sqsubseteq)$ be a complete lattice and let $f \in \mathcal{A} \longleftarrow \mathcal{A}$ be a monotonic endofunction on $\mathcal{A}$. Let $\mathcal{F}$ denote the set of prefix points of $f$, i.e. the subset of $\mathcal{A}$ consisting of all those elements $x$ such that $f.x \sqsubseteq x$. Then $(\mathcal{F}, \sqsubseteq)$ is a complete lattice such that $\sqcap_{\mathcal{F}} = \sqcap_{\mathcal{A}}$.

**Proof** It suffices to show that, for all $X \subseteq \mathcal{F}$, $\sqcap_{\mathcal{A}}.X \in \mathcal{F}$ (see 3.50(**b**)).

$$
\begin{array}{ll}
 & \sqcap_{\mathcal{A}}.X \;\in\; \mathcal{F} \\
\equiv & \qquad \{ \text{ definition of } \mathcal{F} \; \} \\
 & f.\sqcap_{\mathcal{A}}.X \;\sqsubseteq\; \sqcap_{\mathcal{A}}.X \\
\Leftarrow & \qquad \{ \text{ monotonicity of } f\text{: (4.7), transitivity } \} \\
 & \sqcap_{\mathcal{A}}.f.X \;\sqsubseteq\; \sqcap_{\mathcal{A}}.X \\
\Leftarrow & \qquad \{ \text{ monotonicity: (4.15) } \} \\
 & \forall(x :\; x \in X :\; f.x \sqsubseteq x) \\
\equiv & \qquad \{ \text{ definition of } \mathcal{F} \; \} \\
 & X \subseteq \mathcal{F} \quad .
\end{array}
$$

□

The reader may wish to pause and investigate how this lemma can be exploited to prove the theorem mentioned above (the set of closed elements of closure operator $f$ forms a complete lattice). We, ourselves, postpone that discussion to later.

## 7.3 Construction of Closure Operators

Simple as it may seem the prefix lemma, combined with what we already know about Galois connections, unleashes a flood of properties and constructions. For the remainder of this section take $(\mathcal{A}, \sqsubseteq)$ to be a complete lattice and $f$ a monotonic endofunction on $\mathcal{A}$. Let $\mathcal{F}$ denote the collection of prefix points of $f$, hence $\mathcal{F} \subseteq \mathcal{A}$. Let $\iota_{\mathcal{F}} \in \mathcal{A} \longleftarrow \mathcal{F}$ denote the embedding of $\mathcal{F}$ in $\mathcal{A}$. Note that $\iota_{\mathcal{F}}$ is injective. Furthermore we have

(7.8) $\quad f \bullet \iota_{\mathcal{F}} \;\dot{\sqsubseteq}\; \iota_{\mathcal{F}} \quad ,$

which expresses that all elements of $\mathcal{F}$ are prefix points of $f$, and

(7.9) $\quad f \bullet h \;\dot{\sqsubseteq}\; h \;\;\Rightarrow\;\; \iota_{\mathcal{F}} \bullet h \;=\; h \quad ,$

which says that, with the given antecedent, the range of $h$ is contained in $\mathcal{F}$.

From the prefix lemma we know that $\mathcal{F}$ is a complete lattice. But we can extract more from the prefix lemma.

**Theorem 7.10 (Closure Operators)** For $(\mathcal{A}, \sqsubseteq)$ a complete lattice and $f$ a monotonic endofunction on $\mathcal{A}$, there is a unique function $f^\star \in \mathcal{A} \longleftarrow \mathcal{A}$ such that

**a** $\quad f \bullet f^\star \;\dot{\sqsubseteq}\; f^\star \quad ,$

**b** $\quad (f^\star \bullet g \;\dot{\sqsubseteq}\; h \;\;\equiv\;\; g \;\dot{\sqsubseteq}\; h) \quad \Leftarrow \quad f \bullet h \;\dot{\sqsubseteq}\; h \quad .$

Moreover, $f^\star$ is a closure operator.

**Proof** Let $\mathcal{F}$ be the set of $f$-prefix points of $\mathcal{A}$. By the prefix lemma (lemma 7.7) the infima in $(\mathcal{A}, \sqsubseteq)$ and $(\mathcal{F}, \sqsubseteq)$ coincide. That is the same as saying that $\iota_\mathcal{F}$ (the function embedding elements of $\mathcal{F}$ in $\mathcal{A}$) is universally $\sqcap_{\mathcal{A}\leftarrow\mathcal{F}}$-junctive, or phrased differently, that $\iota_\mathcal{F}$ has a lower adjoint $\iota_\mathcal{F}^\flat$. Define $f^\star$ as $\iota_\mathcal{F} \bullet \iota_\mathcal{F}^\flat$, hence by theorem 7.4(**a**) $f^\star$ is a closure operator. So $f^\star$ satisfies:

(7.11) $\quad g \;\dot{\sqsubseteq}\; f^\star \bullet h \;\;\equiv\;\; f^\star \bullet g \;\dot{\sqsubseteq}\; f^\star \bullet h \quad ,$

for all functions $g$ and $h$ with range $\mathcal{A}$. It remains to prove that $f^\star$ thus defined has the properties stated in part **a** and **b**.

For part **a** we have

$$
\begin{array}{ll}
 & f \bullet f^\star \;\dot{\sqsubseteq}\; f^\star \\
\equiv & \qquad \{ \text{ definition } f^\star \ \} \\
 & f \bullet \iota_\mathcal{F} \bullet \iota_\mathcal{F}^\flat \;\dot{\sqsubseteq}\; \iota_\mathcal{F} \bullet \iota_\mathcal{F}^\flat \\
\Leftarrow & \qquad \{ \text{ (7.8); monotonicity (4.16), transitivity } \} \\
 & \mathit{true} \quad .
\end{array}
$$

For part **b** assume $f \bullet h \;\dot{\sqsubseteq}\; h$, hence by (7.9) we have $\iota_\mathcal{F} \bullet h \;=\; h$, then

$$
\begin{array}{ll}
 & f^\star \bullet h \;=\; h \\
\equiv & \qquad \{ \text{ assumption } \} \\
 & f^\star \bullet \iota_\mathcal{F} \bullet h \;=\; \iota_\mathcal{F} \bullet h \\
\equiv & \qquad \{ \text{ definition } f^\star \text{ and semi-inverse: } f^\star \bullet \iota_\mathcal{F} \;=\; \iota_\mathcal{F} \ \} \\
 & \mathit{true} \quad .
\end{array}
$$

Instantiating $f^\star \bullet h \;=\; h$ in (7.11) proves part **b**. That clauses **a** and **b** define a unique function, $f^\star$, is straightforward.

□

The form of 7.10(**b**) is delightfully attractive and amenable to straightforward calculation. Here is a first batch of properties that it predicts. Note that in part **g** we use $f^i$ to denote the $i$-fold composition of $f$ with itself. Thus $f^0$ is the identity function and $f^{i+1} \;=\; f \bullet f^i$ .

**Corollary 7.12**

**a** $g \mathrel{\dot{\sqsubseteq}} f^\star \bullet h \quad \equiv \quad f^\star \bullet g \mathrel{\dot{\sqsubseteq}} f^\star \bullet h$ ,

**b** $I_\mathcal{A} \mathrel{\dot{\sqsubseteq}} f^\star$ ,

**c** $f^\star \bullet f^\star = f^\star$ ,

**d** $f^\star$ is monotonic ,

**e** $f^\star \bullet g \mathrel{\dot{\sqsubseteq}} h \quad \Leftarrow \quad g \mathrel{\dot{\sqsubseteq}} h \wedge f \bullet h \mathrel{\dot{\sqsubseteq}} h$ ,

**f** $f^\star = I_\mathcal{A} \mathbin{\dot{\sqcup}} f \bullet f^\star$ ,

**g** $\forall(i:\ 0 \le i:\ f^i \mathrel{\dot{\sqsubseteq}} f^\star)$ ,

**h** $h = f^\star \bullet h \quad \equiv \quad f \bullet h \mathrel{\dot{\sqsubseteq}} h$ ,

**i** $f^\star = f \quad \equiv \quad f$ is a closure operator ,

**j** $(f^\star)^\star = f^\star$ ,

**k** $^\star$ is monotonic ,

**l** $^\star$ is a closure operator, i.e. $f \mathrel{\dot{\sqsubseteq}} g^\star \quad \equiv \quad f^\star \mathrel{\dot{\sqsubseteq}} g^\star$ .

**Proof**

Part **a** says that $f^\star$ is a closure operator and follows immediately from its definition, theorem 7.10. Parts **b**, **c** and **d** follow from **a** with the aid of corollary 7.2. Part **e** is just a weaker form of theorem 7.10(**b**) in which the equivalence has been weakened to a follows-from. We often use theorem 7.10(**b**) in this weaker form, and for this reason have stated it explicitly.

Part **f** has the following proof:

$$
\begin{array}{ll}
 & f^\star = I_\mathcal{A} \mathbin{\dot{\sqcup}} f \bullet f^\star \\
\equiv & \qquad \{\ \textbf{b} \text{ and theorem 7.10}(\textbf{a})\ \} \\
 & f^\star \mathrel{\dot{\sqsubseteq}} I_\mathcal{A} \mathbin{\dot{\sqcup}} f \bullet f^\star \\
\Leftarrow & \qquad \{\ \textbf{e} \text{ with } g,h := I_\mathcal{A}, I_\mathcal{A} \mathbin{\dot{\sqcup}} f \bullet f^\star\ \} \\
 & f \bullet (I_\mathcal{A} \mathbin{\dot{\sqcup}} f \bullet f^\star) \mathrel{\dot{\sqsubseteq}} I_\mathcal{A} \mathbin{\dot{\sqcup}} f \bullet f^\star \\
\Leftarrow & \qquad \{\ I_\mathcal{A} \mathbin{\dot{\sqcup}} f \bullet f^\star \mathrel{\dot{\sqsubseteq}} f^\star, \text{ see first step}\ \} \\
 & f \bullet f^\star \mathrel{\dot{\sqsubseteq}} I_\mathcal{A} \mathbin{\dot{\sqcup}} f \bullet f^\star \\
\equiv & \qquad \{\ \text{calculus}\ \} \\
 & \mathit{true} \quad .
\end{array}
$$

Part **g** is easily proven by induction on $i$. Part **b** is the case $i = 0$ and we have

$$\begin{array}{ll} & f^{i+1} \mathrel{\dot\sqsubseteq} f^\star \\ \Leftarrow & \quad \{ \text{ theorem 7.10}(\mathbf{a})\text{, transitivity } \} \\ & f^{i+1} \mathrel{\dot\sqsubseteq} f \bullet f^\star \\ \Leftarrow & \quad \{ \ f \text{ is monotonic } \} \\ & f^i \mathrel{\dot\sqsubseteq} f^\star \quad . \end{array}$$

Part **h** has already been proven in the course of proving theorem 7.10 (but the reader is invited to find an alternative proof).

The proof of part **i** is by mutual implication. If $f^\star = f$ then $f$ is a closure operator by definition, theorem 7.10. Now assume $f$ is a closure operator. Then $f^\star$ exists since $f$ is monotonic and we have to prove that $f^\star = f$.

$$\begin{array}{ll} & f^\star = f \\ \equiv & \quad \{ \ \mathbf{g}\text{: } f \mathrel{\dot\sqsubseteq} f^\star \ \} \\ & f^\star \mathrel{\dot\sqsubseteq} f \\ \Leftarrow & \quad \{ \ \mathbf{e} \ \} \\ & I_{\mathcal{A}} \mathrel{\dot\sqsubseteq} f \ \wedge \ f \bullet f \mathrel{\dot\sqsubseteq} f \\ \equiv & \quad \{ \ f \text{ is a closure operator } \} \\ & \mathit{true} \quad . \end{array}$$

Part **j** is an immediate consequence of **i** since $f^\star$ is a closure operator.

To prove monotonicity of $^\star$ we have to prove $f^\star \mathrel{\dot\sqsubseteq} g^\star \ \Leftarrow \ f \mathrel{\dot\sqsubseteq} g$ .

$$\begin{array}{ll} & f^\star \mathrel{\dot\sqsubseteq} g^\star \\ \Leftarrow & \quad \{ \ \mathbf{e} \text{ with } g,h := I_{\mathcal{A}}, g \ \} \\ & I_{\mathcal{A}} \mathrel{\dot\sqsubseteq} g^\star \ \wedge \ f \bullet g^\star \mathrel{\dot\sqsubseteq} g^\star \\ \Leftarrow & \quad \{ \ \mathbf{b}\text{; } g \bullet g^\star \mathrel{\dot\sqsubseteq} g^\star \ \} \\ & f \bullet g^\star \mathrel{\dot\sqsubseteq} g \bullet g^\star \\ \Leftarrow & \quad \{ \text{ monotonicity 4.16 } \} \\ & f \mathrel{\dot\sqsubseteq} g \quad , \end{array}$$

which proves part **k**.

Part **l** is proven by

$$\begin{array}{ll} & f^\star \mathrel{\dot\sqsubseteq} g^\star \\ \Rightarrow & \quad \{ \ \mathbf{g}\text{: } f \mathrel{\dot\sqsubseteq} f^\star\text{, transitivity } \} \end{array}$$

$$\begin{array}{ll} & f \mathrel{\dot{\sqsubseteq}} g^\star \\ \Rightarrow & \quad \{ \ \mathbf{k} \ \} \\ & f^\star \mathrel{\dot{\sqsubseteq}} (g^\star)^\star \\ \equiv & \quad \{ \ \mathbf{j} \ \} \\ & f^\star \mathrel{\dot{\sqsubseteq}} g^\star \quad . \end{array}$$

□

From corollary 7.12(**h**), with $h$ instantiated to the constant function $\widehat{x}$ for some $x$, and the prefix lemma we infer that the closed elements of a closure operator form a complete lattice — the property that we speculated may have led to the discovery of Tarski's fixpoint theorem.

We have deliberately chosen to use the symbol "$\star$" to suggest a connection with the so-called "Kleene star" introduced by S.C.Kleene in a famous paper on regular algebra [55]. The two operators are not the same since Kleene defines $f^*$ for *continuous* function $f$ to be $\sqcup.(i: \ 0 \le i : \ f^i)$ whereas the latter is only a lower bound on $f^\star$ (see 7.12(**g**)). (The assumption here is that $i$ ranges over the natural numbers. If $i$ is allowed to range over all ordinals then the two can indeed be proved to be equal.) We consider the Kleene star operator in detail in chapter 8.

Although Kleene's star and our closure operator do not necessarily coincide the most vital properties of the Kleene star are enjoyed by closure operators, most having already been listed in corollary 7.12. The following *decomposition rule* is particularly vital. Its proof also provides a good illustration of our constructive approach to calculation. The goal of the calculation is to find a decomposition of $(f \mathbin{\dot{\sqcup}} g)^\star$ into an expression involving $f^\star$. The right side of the theorem, thus, has to be discovered; we do this by beginning with a dummy right side and then working towards a substitution that fills in all its details.

**Theorem 7.13 (Closure Decomposition)** For monotonic endofunctions $f, g \in \mathcal{A} \longleftarrow \mathcal{A}$,

$$(f \mathbin{\dot{\sqcup}} g)^\star \quad = \quad f^\star \bullet (g \bullet f^\star)^\star \quad .$$

**Proof** In the first part of the proof we derive the right side:

By the construction of $h$:

$$(f \mathbin{\dot{\sqcup}} g)^\star \quad \dot{\sqsubseteq} \quad h$$

$$
\begin{array}{ll}
\Leftarrow & \quad \{\ \text{corollary 7.12}(\mathbf{e}),\ \text{suprema}\ \} \\
& I_{\mathcal{A}} \mathrel{\dot{\sqsubseteq}} h \ \wedge\ f \bullet h \mathrel{\dot{\sqsubseteq}} h \ \wedge\ g \bullet h \mathrel{\dot{\sqsubseteq}} h \\
\equiv & \quad \{\ \text{corollary 7.12}(\mathbf{h})\ \} \\
& I_{\mathcal{A}} \mathrel{\dot{\sqsubseteq}} h \ \wedge\ h = f^{\star} \bullet h \ \wedge\ g \bullet h \mathrel{\dot{\sqsubseteq}} h \\
\equiv & \quad \{\ \bullet \quad h = f^{\star} \bullet e\ ,\ \text{corollary 7.12}(\mathbf{c})\ \} \\
& I_{\mathcal{A}} \mathrel{\dot{\sqsubseteq}} f^{\star} \bullet e \ \wedge\ g \bullet f^{\star} \bullet e \mathrel{\dot{\sqsubseteq}} f^{\star} \bullet e \\
\Leftarrow & \quad \{\ \text{corollary 7.12}(\mathbf{b}),\ \text{transitivity}\ \} \\
& I_{\mathcal{A}} \mathrel{\dot{\sqsubseteq}} e \ \wedge\ g \bullet f^{\star} \bullet e \mathrel{\dot{\sqsubseteq}} e \\
\equiv & \quad \{\ \text{theorem 7.10}(\mathbf{b})\ \text{with}\ f := g \bullet f^{\star}\ \} \\
& (g \bullet f^{\star})^{\star} \mathrel{\dot{\sqsubseteq}} e \ \wedge\ g \bullet f^{\star} \bullet e \mathrel{\dot{\sqsubseteq}} e \\
\Leftarrow & \quad \{\ \text{theorem 7.10}(\mathbf{a})\ \} \\
& e = (g \bullet f^{\star})^{\star} \quad .
\end{array}
$$

From this calculation it follows that $(f \mathbin{\dot{\sqcup}} g)^{\star} \mathrel{\dot{\sqsubseteq}} f^{\star} \bullet (g \bullet f^{\star})^{\star}$ . The opposite inequality is easier to verify:

$$
\begin{array}{ll}
& f^{\star} \bullet (g \bullet f^{\star})^{\star} \\
\dot{\sqsubseteq} & \quad \{\ \text{monotonicity (4.16), (4.17) and of}\ {}^{\star}\ \} \\
& (f \mathbin{\dot{\sqcup}} g)^{\star} \bullet ((f \mathbin{\dot{\sqcup}} g) \bullet (f \mathbin{\dot{\sqcup}} g)^{\star})^{\star} \\
\dot{\sqsubseteq} & \quad \{\ \text{theorem 7.10}(\mathbf{a}),\ \text{monotonicity (4.17)}\ \} \\
& (f \mathbin{\dot{\sqcup}} g)^{\star} \bullet ((f \mathbin{\dot{\sqcup}} g)^{\star})^{\star} \\
= & \quad \{\ \text{corollary 7.12}(\mathbf{j}),\ \text{corollary 7.12}(\mathbf{c})\ \} \\
& (f \mathbin{\dot{\sqcup}} g)^{\star} \quad .
\end{array}
$$

Although the proof is short, the reader should be aware that each of the steps uses the stated properties (in particular monotonicity) several times over. □

The combination of corollary 7.12(**i**) and 7.12(**l**) is intriguing. On the one hand 7.12(**i**) says that every closure operator has the form $f^{\star}$ for some $f$; on the other hand 7.12(**l**) says that $^{\star}$ itself is a closure operator. So a solution to the equation $\theta \ :: \quad {}^{\star} = \theta^{\dot{\star}}$ exists. (Note that the dot above the star on the right of this equation is demanded by type considerations. If $^{\star}$ has type $T = (\mathcal{A} \longleftarrow \mathcal{A}) \longleftarrow (\mathcal{A} \longleftarrow \mathcal{A})$ then $\dot{\star}$ has type $T \longleftarrow T$ and we seek $\theta \in T$.) The question is, can we give an explicit formula? Indeed we can, the main clue being provided by corollary 7.2: $^{\star}$ is the reflexive, transitive closure operator. Specifically, let $sq$ denote the function $f \mapsto f \bullet f$ for arbitrary endofunction $f$. We then have

**Theorem 7.14** For $f$ a monotonic endofunction on $\mathcal{A}$

$$f^\star \;=\; (\widehat{I_\mathcal{A}} \ddot\sqcup sq)^{\dot\star}.f \quad .$$

Note that $\dot\star$ in the latter formula is the lifted version of $^\star$ .

**Proof** As a shorthand we define $f^* \;=\; (\widehat{I_\mathcal{A}} \ddot\sqcup sq)^{\dot\star}.f$ . Since $\dot\star$ is the lifted version of $^\star$, all the porperties derived for $^\star$ thus far will be used for $\dot\star$. The lifting of these properties will be implicit. We trust the reader can verify these properties, if he wishes to do so.

The proof will be by mutual containment. For one side we have

$$\begin{array}{ll}
 & f^* \;\dot\sqsubseteq\; f^\star \\
\Leftarrow & \qquad \{\ \text{corollary } 7.12(\mathbf{e})\ \} \\
 & f \;\dot\sqsubseteq\; f^\star \;\wedge\; (\widehat{I_\mathcal{A}} \ddot\sqcup sq).f^\star \;\dot\sqsubseteq\; f^\star \\
\equiv & \qquad \{\ \text{corollary } 7.12(\mathbf{g}),\ \text{application, definition of } sq\ \} \\
 & I_\mathcal{A} \dot\sqcup f^\star \bullet f^\star \;\dot\sqsubseteq\; f^\star \\
\equiv & \qquad \{\ \text{corollary } 7.12(\mathbf{b}) \text{ and } 7.12(\mathbf{c})\ \} \\
 & \mathit{true} \quad .
\end{array}$$

For the other containment, we first observe that theorem 7.10($\mathbf{a}$) gives (with $f := \widehat{I_\mathcal{A}} \ddot\sqcup sq$)

$$\begin{array}{ll}
 & \mathit{true} \\
\equiv & \qquad \{\ \text{theorem } 7.10(\mathbf{a})\ \} \\
 & (\widehat{I_\mathcal{A}} \ddot\sqcup sq).(\widehat{I_\mathcal{A}} \ddot\sqcup sq)^{\dot\star}.f \;\dot\sqsubseteq\; (\widehat{I_\mathcal{A}} \ddot\sqcup sq)^{\dot\star}.f \\
\equiv & \qquad \{\ \text{definition } f^*\ \} \\
 & (\widehat{I_\mathcal{A}} \ddot\sqcup sq).f^* \;\dot\sqsubseteq\; f^* \\
\equiv & \qquad \{\ \text{application}\ \} \\
 & I_\mathcal{A} \dot\sqcup sq.(f^*) \;\dot\sqsubseteq\; f^* \\
\equiv & \qquad \{\ \text{definition } \dot\sqcup \text{ and } sq\ \} \\
 & I_\mathcal{A} \;\dot\sqsubseteq\; f^* \;\wedge\; f^* \bullet f^* \;\dot\sqsubseteq\; f^* \quad .
\end{array}$$

We can now prove the other containment as follows

$$\begin{array}{ll}
 & f^\star \;\dot\sqsubseteq\; f^* \\
\Leftarrow & \qquad \{\ \text{corollary } 7.12(\mathbf{e})\ \} \\
 & I_\mathcal{A} \;\dot\sqsubseteq\; f^* \;\wedge\; f \bullet f^* \;\dot\sqsubseteq\; f^* \\
\Leftarrow & \qquad \{\ \text{see above}\ \}
\end{array}$$

$$
\begin{array}{ll}
 & f \;\dot{\sqsubseteq}\; f^* \\
\equiv & \quad \{ \text{ corollary 7.12}(\mathbf{b}) \ \} \\
 & \mathit{true} \quad .
\end{array}
$$

Which completes the (non-trivial) proof.
□

Note that theorem 7.14 reinforces the similarity between our closure operator $^\star$ and the Kleene star.

Before moving on to discuss fixed points let us document the relationship between the suprema and infima in the lattice of prefix points vis-à-vis the suprema and infima in the parent lattice.

**Theorem 7.15**

**a** $\quad f^\star \bullet \sqcup_{\mathcal{A}} \;=\; f^\star \bullet \sqcup_{\mathcal{A}} \bullet f^\star$ ,

**b** $\quad \sqcap_{\mathcal{A}} \bullet f^\star \;=\; f^\star \bullet \sqcap_{\mathcal{A}} \bullet f^\star$ ,

**c** $\quad \iota_{\mathcal{F}} \bullet \sqcup_{\mathcal{F}} \;=\; f^\star \bullet \sqcup_{\mathcal{A}} \bullet \iota_{\mathcal{F}}$ .

**Proof** Part **a** is proven by

$$
\begin{array}{ll}
 & f^\star \bullet \sqcup_{\mathcal{A}} \\
= & \quad \{ \text{ definition of } f^\star \ \} \\
 & \iota_{\mathcal{F}} \bullet \iota_{\mathcal{F}}^\flat \bullet \sqcup_{\mathcal{A}} \\
= & \quad \{ \text{ lower adjoints are universally } \sqcup\text{-junctive, semi-inverse } \} \\
 & \iota_{\mathcal{F}} \bullet \sqcup_{\mathcal{F}} \bullet \iota_{\mathcal{F}}^\flat \bullet \iota_{\mathcal{F}} \bullet \iota_{\mathcal{F}}^\flat \\
= & \quad \{ \text{ lower adjoints are universally } \sqcup\text{-junctive, definition of } f^\star \ \} \\
 & f^\star \bullet \sqcup_{\mathcal{A}} \bullet f^\star \quad .
\end{array}
$$

Part **b** is proven in the same vein:

$$
\begin{array}{ll}
 & \sqcap_{\mathcal{A}} \bullet f^\star \\
= & \quad \{ \text{ definition } f^\star \ \} \\
 & \sqcap_{\mathcal{A}} \bullet \iota_{\mathcal{F}} \bullet \iota_{\mathcal{F}}^\flat \\
= & \quad \{ \ \iota_{\mathcal{F}} \text{ is universally } \sqcap\text{-junctive, semi-inverse } \} \\
 & \iota_{\mathcal{F}} \bullet \iota_{\mathcal{F}}^\flat \bullet \iota_{\mathcal{F}} \bullet \sqcap_{\mathcal{F}} \bullet \iota_{\mathcal{F}}^\flat \\
= & \quad \{ \ \iota_{\mathcal{F}} \text{ is universally } \sqcap\text{-junctive, definition } f^\star \ \} \\
 & f^\star \bullet \sqcap_{\mathcal{A}} \bullet f^\star \quad .
\end{array}
$$

For part **c**, observe that $\sqcup_{\mathcal{F}} = \iota_{\mathcal{F}}^{\flat} \bullet \sqcup_{\mathcal{A}} \bullet \iota_{\mathcal{F}}$ (follows from theorem 5.46(**a**) with the obvious instantiations). Using Leibniz with $\iota_{\mathcal{F}}\bullet$ and the definition of $f^{\star}$ gives the desired result.
□

**Exercise 7.16** Let $f$ be a monotonic endofunction. We call $\varphi$ an $f$-*closure* iff $\varphi$ is a closure operator and $f \bullet \varphi \ \dot{\sqsubseteq}\ \varphi$ . Prove that $f^{\star}$ is the least $f$-closure.
□

**Exercise 7.17** In theorem 7.14 we observed a lifting of various properties derived earlier. This exercise goes one step further. Prove that for a monotonic endofunction $f$ on $\mathcal{A}$ we have $(f\bullet)^{\dot\star} = (f^{\star})\bullet$ .
*Hint*: You might want to recall that $(\bullet g)$ has an upper adjoint $(\bullet g)^{\sharp}$ for arbitrary $g$.
□

## 7.4 Fixed Points

The prefix points of monotonic endofunction $f$ form a complete lattice, but how do we know that the lattice is non-trivial (containing, say, only the top element of the parent lattice)? The following remarkable theorem, commonly known as the Knaster-Tarski theorem, says that not only is the lattice of prefix points non-empty but the least element $x$ in the lattice is also one in which the inclusion $f.x \sqsubseteq x$ can be strengthened to an equality. A point $x$ such that $f.x = x$ is called a *fixed point* of $f$. The theorem thus states that, for all monotonic endofunctions $f$, there is a least fixed point of $f$ which coincides with the least prefix point of $f$.

**Theorem 7.18 (Knaster-Tarski)** If $f$ is a monotonic endofunction on complete lattice $\mathcal{A}$ then the equation

(7.19) $x :: \quad x = f.x$

has a unique least solution, denoted $\mu f$, with the characteristic properties

(7.20) $\mu f = f.\mu f$ ,

(7.21) $\mu f \sqsubseteq y \;\Leftarrow\; f.y \sqsubseteq y$ .

**Proof** Applying corollary 7.12(**f**) to $\perp\!\!\!\perp$ gives

$$f^\star.\perp\!\!\!\perp \quad = \quad f.f^\star.\perp\!\!\!\perp \quad .$$

I.e. equation (7.19) does indeed have a solution, namely $x = f^\star.\perp\!\!\!\perp$ . The pointwise interpretation of corollary 7.12(**e**) gives

$$f^\star.x \sqsubseteq y \quad \Leftarrow \quad x \sqsubseteq y \quad \wedge \quad f.y \sqsubseteq y \quad ,$$

which gives, taking $x := \perp\!\!\!\perp$,

$$f^\star.\perp\!\!\!\perp \sqsubseteq y \quad \Leftarrow \quad f.y \sqsubseteq y \quad .$$

Thus we may define $\mu f$ to be $f^\star.\perp\!\!\!\perp$ and, by so doing, we satisfy (7.20) and (7.21). That (7.20) and (7.21) uniquely define $\mu f$ is straightforward, as is the fact the $\mu f$ is the least solution.
□

In this chapter we have taken the unconventional approach of deriving the Knaster-Tarski theorem as a corollary to a property of Galois connections. Thus, for monotonic function $f$, the least fixed point of $f$, $\mu f$, equals $f^\star.\perp\!\!\!\perp$ where $f^\star$ is the canonical closure operator induced by $f$. It would have been entirely possible to have taken the reverse approach, namely to have established the Knaster-Tarski theorem and then used it to characterise $f^\star$. That is part **c** of exercise 7.24.

**Exercise 7.22** Suppose $\oplus \in \mathcal{A} \longleftarrow \mathcal{A} \times \mathcal{A}$ is a binary function that is monotonic in both its arguments. Show that the function $(y \mapsto x \oplus y)^\star$ is a monotonic function. Define the functions $f$, $g$ and $h$ by

**a** $\quad f = (x \mapsto (y \mapsto x \oplus y)^\star.x) \quad ,$

**b** $\quad g = (x \mapsto x \oplus x)$ and

**c** $\quad h = (x \mapsto \mu(y \mapsto x \oplus y)) \quad .$

Prove the following:

**d** $\quad f^\star \; = \; g^\star \quad ,$

**e** $\quad \mu f \; = \; \mu h \quad ,$

**f** $\mu g \;=\; \mu h$ .

□

**Exercise 7.23** For $f$ and $g$ monotonic endofunctions on complete lattice $(\mathcal{A},\sqsubseteq)$ prove the following so-called *fixed point fusion law*:

$$f.\mu(g\bullet f) \;=\; \mu(f\bullet g) \quad .$$

□

**Exercise 7.24** The fixpoint operator (the $\mu$) maps a monotonic endofunction on $\mathcal{A}$ to an element of $\mathcal{A}$. This operator can be lifted to an operator $\dot{\mu}$ that maps a monotonic endofunction over the lattice of monotonic endofunctions on $\mathcal{A}$ (i.e. a function of type $(\mathcal{A}\longleftarrow\mathcal{A})\longleftarrow(\mathcal{A}\longleftarrow\mathcal{A})$ ) to a monotonic endofunction on $\mathcal{A}$ (hence of type $\mathcal{A}\longleftarrow\mathcal{A}$). If $f$ is a monotonic endofunction on $\mathcal{A}$, then $(f\bullet)$ is of type $(\mathcal{A}\longleftarrow\mathcal{A})\longleftarrow(\mathcal{A}\longleftarrow\mathcal{A})$.

For $f$ and $g$ monotonic endofunctions on $\mathcal{A}$, prove the following properties:

**a** $\dot{\mu}(\widehat{f}\,) \;=\; f$ ,

**b** $\dot{\mu}(f\bullet) \;=\; \widehat{\mu f}$ ,

**c** $f^\star\bullet g \;=\; \dot{\mu}(\widehat{g}\,\dot{\sqcup}\,(f\bullet))$ .

□

**Exercise 7.25** An alternative proof of the closure decomposition theorem can be constructed using exercises 7.22(**f**), (7.23) and 7.24(**c**). Unlike the proof given in section 7.3 it is not necessary to prove mutual inclusion; equality can be proved directly. Construct such a proof. (Hint: use exercise 7.24(**c**) in the form

$$f^\star\bullet g \;=\; \dot{\mu}(h\mapsto g\,\dot{\sqcup}\,f\bullet h)$$

so that you are in a position to apply 7.22(**f**).)

□

**Exercise 7.26** [Tarski's theorem [89]] For $(\mathcal{A},\sqsubseteq)$ a complete lattice and $f$ a monotonic endofunction on $\mathcal{A}$, prove that the collection of fixed points of $f$ forms a complete lattice. Give an expression for the suprema or infima in that lattice.

□

## 7.5 Two Example Closure Operators

Providing non-trivial examples of closures is impossible without assuming extra structure in the underlying lattice. Two trivial examples can be given nonetheless and are well worth documenting. The first of these is the identity function on the given lattice.

(7.27) $(I_{\mathcal{A}})^\star \;=\; I_{\mathcal{A}}$ .

**Proof**

$$\begin{array}{ll}
 & (I_{\mathcal{A}})^\star \;=\; I_{\mathcal{A}} \\
\equiv & \qquad \{ \text{ corollary 7.12}(\mathbf{b}) \text{ } \} \\
 & (I_{\mathcal{A}})^\star \;\dot{\sqsubseteq}\; I_{\mathcal{A}} \\
\Leftarrow & \qquad \{ \text{ corollary 7.12}(\mathbf{e}) \text{ with } f,g,h := I_{\mathcal{A}}, I_{\mathcal{A}}, I_{\mathcal{A}} \text{ } \} \\
 & I_{\mathcal{A}} \;\dot{\sqsubseteq}\; I_{\mathcal{A}} \quad .
\end{array}$$

□

Moreover,

(7.28) $f^\star \quad = \quad (I_{\mathcal{A}} \,\dot{\sqcup}\, f)^\star$ .

The proof is a straightforward use of closure-decomposition with $f,g \;:=\; I_{\mathcal{A}}, f$ combined with (7.27).

The second example is the class of constant functions. Suppose $a \in \mathcal{A}$. Let $\widehat{a}$ denote the constant function $(x \mapsto a)$ always returning $a$. Then,

(7.29) $\widehat{a}^\star \quad = \quad (a\sqcup)$ .

**Proof**

$$\begin{array}{ll}
 & \widehat{a}^\star \\
= & \qquad \{ \text{ corollary 7.12}(\mathbf{f}) \text{ } \} \\
 & I_{\mathcal{A}} \,\dot{\sqcup}\, \widehat{a} \bullet \widehat{a}^\star \\
= & \qquad \{ \text{ } \widehat{a} \bullet f \;=\; \widehat{a} \text{ for all } f \text{ } \} \\
 & I_{\mathcal{A}} \,\dot{\sqcup}\, \widehat{a} \\
= & \qquad \{ \text{ calculus } \} \\
 & (a\sqcup) \quad .
\end{array}$$

□

Just as for the identity function, the use of the closure-decomposition rule enables one to simplify a closure term when one of its components is a constant function (which is quite a commonplace occurrence).

(7.30) $(\widehat{a} \mathbin{\dot\sqcup} f)^\star \quad = \quad f^\star \bullet (a\sqcup) \quad = \quad (\widehat{a} \mathbin{\dot\sqcup} f)^\star \bullet (a\sqcup)$ .

**Proof** We note that

$$\begin{array}{ll} & (\widehat{a} \bullet f^\star)^\star \\ = & \qquad \{ \ \widehat{a} \bullet g \quad = \quad \widehat{a} \text{ , for all } g \ \} \\ & (\widehat{a})^\star \\ = & \qquad \{ \ (7.29) \ \} \\ & (a\sqcup) \quad . \end{array}$$

So, by the closure-decomposition rule, $(\widehat{a} \mathbin{\dot\sqcup} f)^\star \quad = \quad f^\star \bullet (a\sqcup)$ . The second equality follows trivially from the observation that $(a\sqcup) \ = \ (a\sqcup) \bullet (a\sqcup)$ .

□

# Chapter 8

# Regular Algebra

In chapter 7 we considered closure operators in general. In this chapter we are going to study one in particular, the *reflexive*, *transitive* closure operator.

We already have one very good reason for wanting to study it: as we saw in theorem 7.14, every closure operator is the reflexive, transitive closure of some monotonic function. That, however, is a very abstract reason; there are several concrete reasons why such a study is justified. One such is that, in many areas of mathematics, ordering relations are commonly introduced by considering the reflexive, transitive closure of a primitive relation. For example, the at-most relation on natural numbers is the reflexive, transitive closure of the successor function. This, indeed, is the area from which the terminology ("reflexive", "transitive") is borrowed. It may help if you keep the relational model in mind to interpret the theorems we present (but, of course, not individual calculation steps!). We, ourselves, continue to pitch the discussion at an abstract, axiomatic level thus admitting more models (some of which will be introduced later).

In order to *define* a reflexive, transitive closure operator a modicum of algebraic structure in the underlying lattice is essential. In order that such an operator be mathematically interesting, more algebraic structure is desirable. Just how much will become apparent in the course of this chapter. From the point of view of the models that are prevalent in many application areas yet more structure is usually assumed — in particular the structure of what we will call a "regular algebra". A regular algebra is quite a rich structure so, in order to make clear the rôle of each of its elements, we break it down into a hierarchy of structures. In this hierarchy the algebras that we dub *semi-regular* play the leading rôle.

Particularly interesting about reflexive, transitive closures is that there are several ways of defining them. The most direct way is as the least reflexive, transitive lattice element that includes the given primitive element. Other definitions are less direct but more suited to some sorts of calculation. We consider three different definitions — the direct one and two indirect definitions — and we establish their equivalence. We also look at the relationship between transitive closures and reflexive, transitive closures.

## 8.1 Factors

So that the reader may have a clear idea of where we are heading we introduce the definition of a regular algebra in this section even though we do not make use of all elements of the definition until section 8.4. The most unusual elements of the definition are the dual notions of "left" and "right factor".

**Definition 8.1 (Regular Algebra)** A *regular algebra* consists of a set $\mathcal{A}$ with the following algebraic structure.

- An ordering relation $\sqsubseteq$ is defined on $\mathcal{A}$ such that $(\mathcal{A}, \sqsubseteq)$ is a complete lattice. As usual we will denote the supremum operator in $\mathcal{A}$ by $\sqcup$.

- The set $\mathcal{A}$ contains a distinguished element $I$ and is closed under a binary operator $\circ$ such that $(\mathcal{A}, \circ, I)$ is a monoid. (I.e. $I$ is a unit of $\circ$, and $\circ$ is associative.) The operator $\circ$ will be referred to as the *composition* operator.

- For all $x$ in $\mathcal{A}$, the functions $(x\circ)$ and $(\circ x)$ are universally $\sqcup$-junctive.

$\Box$

On account of the last requirement, the composition operator in a regular algebra is monotonic in both its arguments. Moreover, exploiting the monoid structure to extend composition to an arbitrary number of arguments (the composition of zero lattice elements being $I$) it is also the case that an $n$-fold composition is monotonic in all $n$ arguments.

The last requirement is equivalent to the functions $(x\circ)$ and $(\circ x)$ having upper adjoints (for each lattice element $x$). We denote these adjoint functions by $(x\backslash)$ and $(/x)$ respectively. Thus, in a regular algebra two binary *factoring* operators are automatically defined by the Galois connections:

$$(8.2) \quad x \circ y \sqsubseteq z \quad \equiv \quad y \sqsubseteq x\backslash z$$
$$(8.3) \quad x \circ y \sqsubseteq z \quad \equiv \quad x \sqsubseteq z/y \quad .$$

A term of the form $x\backslash z$ is called a *right factor* of $z$, and a term of the form $z/y$ is called a *left factor* of $z$.

We encountered factors earlier in this monograph. In section 5.5.4 factors were examined in two specific settings, the relations over a universe and in the realm of regular languages. These are not the only fields in which factors are used. The interested reader can find some bibliographic references at the beginning of section 5.5.4.

The reason that we call these two operators "factoring" operators is that if one draws an analogy between composition and multiplication then the operators $\backslash$ and $/$ behave somewhat like division in that we have the following *cancellation* properties.

$$(8.4) \quad x \circ x\backslash y \quad \sqsubseteq \quad y$$
$$(8.5) \quad x/y \circ y \quad \sqsubseteq \quad x \quad .$$

(The analogy should not be stretched any further. Composition is not assumed to be commutative and arguments of composition and factoring may only be cancelled when they are adjacent as in the equations above.)

A model of a regular algebra, of particular interest to computing scientists, occurs in language theory. Suppose $T$ is some finite, non-empty set of "symbols". Define $T^*$ to be the set of all strings of symbols (including the empty string), and let $(\mathcal{A}, \sqsubseteq)$ be the set of all subsets of $T^*$ ordered by set inclusion. Define the binary operator $\circ$ by

$$L \circ M \quad = \quad \cup.(x, y : \; x \in L \;\wedge\; y \in M : xy)$$

for all $L, M \subseteq T^*$. Also define $I$ to be the set containing just the empty string. Then, with these definitions, we have constructed a regular algebra. (This, indeed, is the model in which the name "regular algebra" first appeared.)

We often summarise the existence of left and right factors — perhaps somewhat sloppily — by saying "composition is universally $\sqcup$-junctive". But, it is important to note that the property entails two separate axioms. One is that, for all $x$, the function $(\circ x)$ is universally $\sqcup$-junctive and so has an upper adjoint.

The other is that, for all $x$, the function $(x\circ)$ is universally $\sqcup$-junctive and so has an upper adjoint. A model that satisifies the first axiom but not the second is the monoid *MONO*.$\mathcal{A}$ formed by the monotonic endofunctions, ordered pointwise, on a complete lattice $\mathcal{A}$. (See the discussion at the beginning of chapter 7 for a definition of *MONO*.$\mathcal{A}$.) In this chapter we derive several properties by combining properties obtained by assuming universal $\sqcup$-junctivity of $(\circ x)$, for some $x$, with the dual properties obtained by assuming universal $\sqcup$-junctivity of $(x\circ)$. The first set of properties remains valid in *MONO*.$\mathcal{A}$ (and thus this chapter extends the study of $f^\star$ begun in chapter 7) but their duals are typically *not* valid in *MONO*.$\mathcal{A}$. (See for example exercise 8.45).

In order to distinguish more clearly between those properties that rely on both $\sqcup$-junctivity properties from those that rely on just one we propose the introduction of the term "semi-regular" algebra. By a *semi-regular algebra* we mean a complete lattice $(\mathcal{A},\sqsubseteq)$ exhibiting a monoid structure $(\mathcal{A},\circ,I)$ such that the operator $\circ$ is monotonic in its second argument and, for all $x$, the function $(\circ x)$ is universally $\sqcup$-junctive. The composition operator in a semi-regular algebra is thus monotonic in both its arguments but, unlike a regular algebra, $\sqcup$-junctive in only one.

Our axiomatisation of a regular algebra is not the only one that has been proposed. Conway [30] discusses several alternatives, although none of his axiom systems concides with ours. Ironically, although Conway introduced and exploited factors he never saw fit to base an axiomatisation of regular algebra on their existence!

Other axiomatisations of regular algebra typically postulate the existence of a unary operator — the so-called "Kleene star" — with certain properties including a decomposition rule similar to the closure decomposition rule in chapter 7. It will be our goal in this chapter to show how the Kleene star can be defined in a regular algebra in such a way that it automatically satisfies such a decomposition rule. Our discussion is organised roughly according to the four requirements on a regular algebra. To begin with we assume only that $(\mathcal{A},\sqsubseteq)$ is a complete lattice containing a distingushed element $I$, and on which is defined a binary operator $\circ$ that is monotonic in both its arguments. No further assumptions will be made until section 8.3. Then we consider the consequences of admitting a semi-regular algebra, and finally we consider regular algebras.

## 8.2 The Kleene Star

### 8.2.1 Direct Definition

Throughout the remainder of this section we assume that $(\mathcal{A}, \sqsubseteq)$ is a complete lattice containing a distingushed element $I$, and on which is defined a binary operator $\circ$ that is monotonic in both its arguments.

Let $x$ be an element of $\mathcal{A}$. We call $x$ *reflexive* if $I \sqsubseteq x$. We call $x$ *transitive* if $x \circ x \sqsubseteq x$.

The definitions of reflexive and transitive correspond in the relational model to what we normally understand by reflexivity and transitivity of a relation. For

$$
\begin{array}{ll}
 & [\![R \text{ is reflexive}]\!] \\
\equiv & \qquad \{ \text{ definition } \} \\
 & [\![I \sqsubseteq R]\!] \\
\equiv & \qquad \{ \text{ interpretations of } I \text{ and } \sqsubseteq \text{ in the relational model } \} \\
 & \forall(x,y :: \ x = y \Rightarrow x[\![R]\!]y) \\
\equiv & \qquad \{ \text{ one-point rule } \} \\
 & \forall(x :: \ x[\![R]\!]x) \quad ,
\end{array}
$$

and

$$
\begin{array}{ll}
 & [\![R \text{ is transitive}]\!] \\
\equiv & \qquad \{ \text{ definition } \} \\
 & [\![R \circ R \sqsubseteq R]\!] \\
\equiv & \qquad \{ \text{ interpretation of } \sqsubseteq \text{ in the relational model } \} \\
 & \forall(x,z :: \ x[\![R \circ R]\!]z \ \Rightarrow \ x[\![R]\!]z) \\
\equiv & \qquad \{ \text{ interpretation of } \circ \text{, range disjunction } \} \\
 & \forall(x,y,z :: \ x[\![R]\!]y \wedge y[\![R]\!]z \ \Rightarrow \ x[\![R]\!]z) \quad .
\end{array}
$$

The transitive closure of $x$ is conventionally denoted $x^+$, and the reflexive, transitive closure $x^*$. The operator $^*$ is often referred to as the *Kleene star* [55]. We stick to these conventions in order to give the formulae we derive a familiar appearance.

The transitive closure of $x$ is the least transitive lattice element that includes $x$. Letting *sq* denote the function $(x \mapsto x \circ x)$ we therefore define

(8.6) $\quad x^+ \ = \ sq^\star.x \quad .$

The reflexive, transitive closure of $x$ is the least reflexive, transitive lattice element that includes $x$. Letting $\hat{I}$ denote the constant function $(x \mapsto I)$ we therefore define

$$(8.7) \quad x^* \;=\; (\hat{I} \,\dot{\sqcup}\, sq)^\star.x \quad .$$

Note that these equations are literal translations of the English descriptions. From the definition of closure operators (theorem 7.10) and corollary 7.12(**b**) we have:

$$(8.8) \quad x^+ \circ x^+ \;=\; sq.sq^\star.x \;\sqsubseteq\; sq^\star.x \;=\; x^+ \quad ,$$

$$(8.9) \quad x \;\sqsubseteq\; x^+ \quad ,$$

and,

$$(8.10) \quad (x \sqsubseteq y \;\;\equiv\;\; x^+ \sqsubseteq y) \;\;\Leftarrow\;\; y \circ y \sqsubseteq y \quad .$$

Thus $x^+$ is transitive — (8.8) —, includes $x$ — (8.9) — and is least among such values — (8.10) . Similarly,

$$(8.11) \quad I \;\sqsubseteq\; x^* \;\;\wedge\;\; x^* \circ x^* \;\sqsubseteq\; x^* \quad ,$$

$$(8.12) \quad x \;\sqsubseteq\; x^* \quad ,$$

and

$$(8.13) \quad (x \sqsubseteq y \;\;\equiv\;\; x^* \sqsubseteq y) \;\;\Leftarrow\;\; (I \sqsubseteq y \;\;\wedge\;\; y \circ y \sqsubseteq y) \quad .$$

By instantiating the properties of closure operators discussed in the last section we obtain several properties of the transitive closure and the reflexive and transitive closure. Some of these are: from 7.12(**a**)

$$(8.14) \quad x \sqsubseteq y^+ \;\;\equiv\;\; x^+ \sqsubseteq y^+ \quad \text{and} \quad x \sqsubseteq y^* \;\;\equiv\;\; x^* \sqsubseteq y^* \quad ,$$

from 7.12(**h**)

$$(8.15) \quad x = x^+ \;\;\equiv\;\; x \circ x \sqsubseteq x \quad \text{and} \quad x = x^* \;\;\equiv\;\; I \sqsubseteq x \wedge x \circ x \sqsubseteq x \quad ,$$

from 7.12(**c**)

$$(8.16) \quad x^+ = (x^+)^+ \quad \text{and} \quad x^* = (x^*)^* \quad ,$$

from 7.12(**k**) (since $sq \mathrel{\dot{\sqsubseteq}} \hat{I} \mathbin{\dot{\sqcup}} sq$)

(8.17) $x^+ \sqsubseteq x^*$ .

From 7.15(**b**)

(8.18) $x^+ \sqcap y^+ = (x^+ \sqcap y^+)^+$ and $x^* \sqcap y^* = (x^* \sqcap y^*)^*$ .

Finally, from (7.30) we obtain

(8.19) $x^* \;=\; (I \sqcup x)^+ \;=\; (I \sqcup x)^*$ .

**Exercise 8.20** Note that none of these properties depends in any way on the fact that composition is universally $\sqcup$-junctive, or on the fact that $(\mathcal{A}, \circ, I)$ is a monoid. Only the existence of $I$, and the existence and monotonicity of a binary composition operator are needed. The challenge is thus to discover what consequences this extra structure has on the properties of the two operators individually, and on their relationship to each other. In this exercise we consider some of these consequences. Specifically:

(8.21) $\perp\!\!\perp^* \;=\; I^+ \;=\; I^* \;=\; I$ ,

(8.22) $x^* \;=\; I \sqcup x^+$ ,

and

(8.23) $x^* \;=\; x^* \circ x^*$ .

Prove these properties.

□

### 8.2.2 Indirect Definition

Yet more properties of the star operator can be derived by establishing its equality to two other closure operators. Specifically, let $a$ be an element of a regular algebra and consider the two closure operators $(a\circ)^\star$ and $(\circ a)^\star$. Then, we claim,

(8.24) $a^* \;=\; (a\circ)^\star.I \;=\; (\circ a)^\star.I$ .

Note that the assumption we have just made is that $a$ is an element of a *regular* algebra. Both $(a\circ)^\star$ and $(\circ a)^\star$ are well defined at the level of the primitive algebraic structure we are now considering but there is very little to say about their properties at this level (and absolutely nothing about their relationship to the Kleene star)! Shortly, therefore, we move on to consider semi-regular algebras. In a semi-regular algebra we will be able to establish the equality of the first two terms. Then, in a regular algebra, we can dualise all equations in a semi-regular algebra by just turning all the compositions around. In this way we establish equality between the first and third terms.

For future reference we state the characteristic properties of $(a\circ)^\star$.

(8.25) $a \circ (a\circ)^\star.x \sqsubseteq (a\circ)^\star.x$ ,

and

(8.26) $((a\circ)^\star.x \sqsubseteq y \equiv x \sqsubseteq y) \Leftarrow a \circ y \sqsubseteq y$ .

## 8.3 Semi-regular Algebras

The time has come to assume the structure offered by a semi-regular algebra. Recall that a semi-regular algebra is a complete lattice $(\mathcal{A}, \sqsubseteq)$ exhibiting a monoid structure $(\mathcal{A}, \circ, I)$ such that the operator $\circ$ is monotonic in its second argument and, for all $x$, the function $(\circ x)$ is universally $\sqcup$-junctive. The composition operator in a semi-regular algebra is thus monotonic in both its arguments but, unlike a regular algebra, $\sqcup$-junctive in only one. In particular the factoring operator / defined by equation 8.3 exists but its counterpart \ need not.

The next two subsections consider $a^*$ and $(a\circ)^\star$ separately. In the third subsection we put the results we have obtained together to obtain the relationship between the two claimed in (8.24).

### 8.3.1 A Leapfrog Rule

The first property we prove we call a *leapfrog rule*. We give it this name because it gives a condition under which an element $x$ may "leapfrog" from one side to the other of a star term.

(8.27) $a^* \circ x \sqsubseteq x \circ b^* \Leftarrow a \circ x \sqsubseteq x \circ b$ .

**Proof**

$$
\begin{array}{ll}
 & a^* \circ x \;\sqsubseteq\; x \circ b^* \\
\equiv & \qquad \{ \text{ factors: (8.3) } \} \\
 & a^* \;\sqsubseteq\; (x \circ b^*) \,/\, x \\
\equiv & \qquad \{ \text{ We apply (8.13) postponing temporarily the proof that} \\
 & \qquad\qquad (x \circ b^*)/x \text{ is reflexive and transitive } \} \\
 & a \;\sqsubseteq\; (x \circ b^*)/x \\
\equiv & \qquad \{ \text{ factors: (8.3) } \} \\
 & a \circ x \;\sqsubseteq\; x \circ b^* \\
\Leftarrow & \qquad \{ \text{ by (8.12), } b \sqsubseteq b^* \text{ ;} \\
 & \qquad\; \text{composition is monotonic in its 2nd argument } \} \\
 & a \circ x \;\sqsubseteq\; x \circ b \quad .
\end{array}
$$

The postponed second step is derived as follows:

$$
\begin{array}{ll}
 & I \;\sqsubseteq\; (x \circ b^*)/x \\
\equiv & \qquad \{ \text{ factors: (8.3), } I \text{ is a unit } \} \\
 & x \sqsubseteq x \circ b^* \\
\equiv & \qquad \{ \text{ by (8.11), } I \sqsubseteq b^* \; \} \\
 & \mathit{true} \quad ,
\end{array}
$$

and

$$
\begin{array}{ll}
 & (x \circ b^*)/x \;\circ\; (x \circ b^*)/x \;\sqsubseteq\; (x \circ b^*)/x \\
\equiv & \qquad \{ \text{ factors: (8.3) } \} \\
 & (x \circ b^*)/x \;\circ\; (x \circ b^*)/x \;\circ\; x \;\sqsubseteq\; x \circ b^* \\
\Leftarrow & \qquad \{ \text{ cancellation: (8.5), and monotonicity } \} \\
 & (x \circ b^*)/x \circ x \circ b^* \;\sqsubseteq\; x \circ b^* \\
\Leftarrow & \qquad \{ \text{ cancellation: (8.5), and monotonicity } \} \\
 & x \circ b^* \circ b^* \;\sqsubseteq\; x \circ b^* \\
\equiv & \qquad \{ \text{ by (8.11), } b^* \text{ is transitive } \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

### 8.3.2 Closure Fusion

Now we turn to $(a\circ)^\star$. Here the first property we observe can be summarised in words by: the function $(a\circ)^\star$ is completely determined by its value at $I$. In a formula, and in more detail:

$$(8.28)\quad (a\circ)^\star.b \;=\; (a\circ)^\star.I \circ b \;\;,$$

our second example of a *fusion property*. (Recall the fixed point fusion property in exercise 7.23.) Note that an immediate consequence of the fusion property is that for all $b$ and $c$ we have

$$(8.29)\quad (a\circ)^\star.b \circ c \;=\; (a\circ)^\star.(b \circ c) \;\;.$$

Thus a composition of a term of the form $(a\circ)^\star.b$ and some other term can always be "fused together" to form a term of the first form.

The proof of (8.28) is delightfully straightforward. For a change we shall appeal to the unicity of adjoints in a Galois connection rather than the rule of indirect equality. (A proof using the latter rule is almost identical to the one we give but requires a couple of extra steps.) The function $(a\circ)^\star$ is by definition the lower adjoint of the function embedding lattice elements $y$ satisfying $a\circ y \sqsubseteq y$ in the lattice $\mathcal{A}$. It suffices therefore to show that the function $(b \mapsto (a\circ)^\star.I \circ b)$ , firstly, maps elements of $\mathcal{A}$ into such lattice elements and, secondly, has the same embedding function as its upper adjoint.

The first proof obligation is soon dismissed. By (8.25) with $x$ instantiated to $I$,

$$(8.30)\quad a\circ(a\circ)^\star.I \sqsubseteq (a\circ)^\star.I \;\;.$$

So, by the monotonicity of composition,

$$(8.31)\quad a\circ(a\circ)^\star.I \circ b \;\sqsubseteq\; (a\circ)^\star.I \circ b \;\;.$$

The second proof obligation amounts to proving the equivalence

$$(8.32)\quad (a\circ)^\star.I\circ b \sqsubseteq y \quad\equiv\quad b \sqsubseteq y$$

for all $y$ such that $a\circ y \sqsubseteq y$ . Let us make that assumption of $y$. Then,

$$
\begin{array}{ll}
 & (a\circ)^\star.I \;\circ\; b \;\;\sqsubseteq\;\; y \\
\equiv & \qquad \{ \;\text{factors: (8.3)}\; \} \\
 & (a\circ)^\star.I \;\;\sqsubseteq\;\; y/b \\
\equiv & \qquad \{ \;\text{We aim to use (8.26). Now,} \\
 & \qquad\qquad \begin{array}{ll}
 & a \circ y/b \;\;\sqsubseteq\;\; y/b \\
 \equiv & \qquad \{ \;\text{factors: (8.3)}\; \} \\
 & a \circ y/b \circ b \;\;\sqsubseteq\;\; y \\
 \Leftarrow & \qquad \{ \;\text{cancellation: (8.5)}\; \} \\
 & a \circ y \sqsubseteq y \\
 \equiv & \qquad \{ \;\bullet \quad a \circ y \sqsubseteq y \; \} \\
 & \mathit{true} \quad .
 \end{array} \\
 & \qquad\;\; \text{Thus (8.26) can indeed be applied.} \; \} \\
 & I \sqsubseteq y/b \\
\equiv & \qquad \{ \; I \circ b \;=\; b\text{, factors: (8.3)} \; \} \\
 & b \sqsubseteq y \quad .
\end{array}
$$

### 8.3.3 Coincidence of the Direct and Indirect Definitions

In this section we prove the identity:

(8.33) $a^* \quad = \quad (a\circ)^\star.I \quad .$

The key to the proof is the closure fusion property (8.28). For, instantiating $b$ to $(a\circ)^\star.I$, we readily see that $(a\circ)^\star.I$ is transitive:

$$
\begin{array}{ll}
 & (a\circ)^\star.I \\
= & \qquad \{ \;\text{corollary 7.12}(\mathbf{c})\; \} \\
 & (a\circ)^\star \,.\, (a\circ)^\star \,.\, I \\
= & \qquad \{ \;(8.28)\; \} \\
 & (a\circ)^\star.I \;\circ\; (a\circ)^\star.I \quad .
\end{array}
$$

The proof of (8.33) is now a piece of cake:

$$
\begin{array}{ll}
 & a^* \sqsubseteq (a\circ)^\star.I \\
\Leftarrow & \qquad \{ \;(8.13)\; \} \\
 & a \sqsubseteq (a\circ)^\star.I \quad\wedge\quad (a\circ)^\star.I \;\circ\; (a\circ)^\star.I \sqsubseteq (a\circ)^\star.I \quad\wedge\quad I \sqsubseteq (a\circ)^\star.I \\
\equiv & \qquad \{ \;\text{(8.25) and 7.12}(\mathbf{b})\text{; above; 7.12}(\mathbf{b})\; \} \\
 & \mathit{true} \quad ,
\end{array}
$$

and

$$
\begin{array}{ll}
 & (a\circ)^\star.I \ \sqsubseteq\ a^* \\
\Leftarrow & \qquad \{\ (8.26)\ \} \\
 & I \sqsubseteq a^* \quad \wedge \quad a \circ a^* \sqsubseteq a^* \\
\Leftarrow & \qquad \{\ (8.11) \text{ and monotonicity of } \circ\ \} \\
 & a \sqsubseteq a^* \\
\equiv & \qquad \{\ (8.12)\ \} \\
 & \mathit{true} \quad .
\end{array}
$$

Useful though it may be, the form of (8.33) is, to our taste, somewhat unpleasant making it difficult to memorize. A slight modification lifting the property from element level to the level of functions makes a world of difference. Specifically,

$$(8.34) \quad (a^*)\circ \quad = \quad (a\circ)^\star \quad ,$$

since for all elements $b$

$$
\begin{array}{ll}
 & a^* \ \circ\ b \\
= & \qquad \{\ (8.33)\ \} \\
 & (a\circ)^\star.I \ \circ\ b \\
= & \qquad \{\ (8.28)\ \} \\
 & (a\circ)^\star.b \quad .
\end{array}
$$

**Exercise 8.35** Prove that in a semi-regular algebra $x^+ = (x\circ)^\star.x$ .

□

**Exercise 8.36** By combining (8.27) and (8.33) it is straightforward to prove the identity:

$$(a\circ)^\star.x \quad \sqsubseteq \quad x \ \circ\ (b\circ)^\star.I \qquad \Leftarrow \qquad a\circ x \sqsubseteq x \circ b \quad .$$

Give a direct proof of this identity using, instead of (8.27), the Galois connection defining $(a\circ)^\star$, namely:

$$((a\circ)^\star.x \sqsubseteq y \quad \equiv \quad x \sqsubseteq y) \qquad \Leftarrow \qquad a\circ y \sqsubseteq y \quad .$$

□

### 8.3.4 Star Decomposition

In chapter 7 we claimed that the closure-decomposition rule is vitally important although seldom recognised as such. Some examples of its use have already been given but the next calculation stands out.

What we show is that in a semi-regular algebra the Kleene star operator obeys exactly the same decomposition rule as the closure operator. Specifically, for all $a$ and $b$,

$$(8.37)\quad (a \sqcup b)^* \;=\; a^* \circ (b \circ a^*)^* \quad .$$

The indirect definition of reflexive transitive closure motivates a search for the existence of such a property. The question that naturally arises is how the *family* of functions $(x\circ)^\star$, where $x$ is taken to range over all lattice elements, behaves with respect to the underlying lattice structure. In other words, is it possible to express the function $((a \sqcup b)\circ)^\star$ in terms of the functions $(a\circ)^\star$ and $(b\circ)^\star$ ? Given this question and equation (8.34) one is thus led to seek a decomposition of the element $(a \sqcup b)^*$ in terms of the elements $a^*$ and $b^*$.

The closure decomposition rule is the obvious tool to use to pursue such an investigation, and indeed its use is very straightforward — so long as one is comfortable with working at the level of functions rather than at the level of functions applied to arguments! Since typically that is not the case some preliminary remarks are required before we embark on the calculation.

We note first that the associativity of composition can be expressed as an equation between functions of the form $(x\circ)$. By omitting the (implicitly universally quantified) argument $z$ in the equation

$$(8.38)\quad (x \circ y) \circ z \;=\; x \circ (y \circ z) \quad ,$$

we get the equation

$$(8.39)\quad (x \circ y)\circ \;=\; (x\circ) \bullet (y\circ) \quad .$$

(The principal operator on the right of this equation is indeed function composition — there is no typographical error!) Second we note that the fact that for all $z$ the function $(\circ z)$ distributes through $\sqcup$ can also be expressed without the need for the argument $z$. Specifically,

$$(8.40)\quad (x \sqcup y) \circ z \;=\; (x \circ z) \sqcup (y \circ z)$$

converts to

(8.41) $(x \sqcup y)\circ \quad = \quad (x\circ) \mathbin{\dot\sqcup} (y\circ) \quad .$

With these preliminaries we now proceed to investigate the function $((a \sqcup b)\circ)^{\star}$. We have

$$
\begin{array}{ll}
 & ((a \sqcup b)\circ)^{\star} \\
= & \qquad \{ \;\; (8.41) \;\; \} \\
 & ((a\circ) \mathbin{\dot\sqcup} (b\circ))^{\star} \\
= & \qquad \{ \;\; \text{closure decomposition, theorem 7.13} \;\; \} \\
 & (a\circ)^{\star} \;\bullet\; ((b\circ) \bullet (a\circ)^{\star})^{\star} \\
= & \qquad \{ \;\; \text{(8.34) applied twice} \;\; \} \\
 & (a^*)\circ \quad \bullet \quad ((b\circ) \bullet (a^*)\circ)^{\star} \\
= & \qquad \{ \;\; (8.39)) \;\; \} \\
 & (a^*)\circ \quad \bullet \quad ((b \;\circ\; a^*)\circ)^{\star} \\
= & \qquad \{ \;\; (8.34), (8.39) \;\; \} \\
 & (a^* \;\circ\; (b \;\circ\; a^*)^*)\circ \quad .
\end{array}
$$

Hence,

$$
\begin{array}{ll}
 & (a \sqcup b)^* \\
= & \qquad \{ \;\; I \text{ is unit of composition} \;\; \} \\
 & (a \sqcup b)^* \;\circ\; I \\
= & \qquad \{ \;\; (8.34) \;\; \} \\
 & ((a \sqcup b)\circ)^{\star}.I \\
= & \qquad \{ \;\; \text{above} \;\; \} \\
 & a^* \;\circ\; (b \;\circ\; a^*)^* \;\circ\; I \\
= & \qquad \{ \;\; I \text{ is unit of composition} \;\; \} \\
 & a^* \;\circ\; (b \;\circ\; a^*)^* \quad .
\end{array}
$$

## 8.4 Regular Algebras

In a semi-regular algebra there is an asymmetry between composition on the left and composition on the right. In a regular algebra this asymmetry completely disappears. Any property proved in a semi-regular algebra can thus be dualised in a regular algebra by turning all compositions around. In particular, in a regular algebra the dual of (8.27) is also true.

(8.42) $x \circ a^* \;\sqsubseteq\; b^* \circ x \quad \Leftarrow \quad x \circ a \sqsubseteq b \circ x \quad .$

Interchanging $a$ and $b$ in (8.27) and combining it with (8.42) we thus obtain:

(8.43) $x \circ a^* = b^* \circ x \quad \Leftarrow \quad x \circ a = b \circ x$ .

The antecedent in this equation can be made true if $a$ is substituted for $x$, $b \circ a$ is substituted for $a$, and $b$ is instantiated to $a \circ b$. We thus obtain the *leapfrog rule*:

(8.44) $a \circ (b \circ a)^* \quad = \quad (a \circ b)^* \circ a$ .

The star-decomposition rule has already been proved in section 8.3.4. We think it is sufficiently important however to prove it yet again directly from the direct definition of reflexive and transitive closure.

To prove the rule we exploit the rule of indirect equality with the domain predicate $p$ defined by $p.y$ equivales $y$ is both reflexive and transitive. That is, we show that for all reflexive and transitive $y$,

$$(a \sqcup b)^* \sqsubseteq y \quad \equiv \quad a^* \circ (b \circ a^*)^* \sqsubseteq y$$

following which we verify that $a^* \circ (b \circ a^*)^*$ is reflexive and transitive. (By definition $(a \sqcup b)^*$ is reflexive and transitive.)

**Proof**

$$
\begin{array}{ll}
 & (a \sqcup b)^* \sqsubseteq y \\
\equiv & \qquad \{ \ \bullet \quad y \text{ is reflexive and transitive, (8.13)} \ \} \\
 & a \sqcup b \sqsubseteq y \\
\equiv & \qquad \{ \ \text{suprema} \ \} \\
 & a \sqsubseteq y \wedge b \sqsubseteq y \\
\equiv & \qquad \{ \ \bullet \quad y \text{ is reflexive and transitive, (8.13)} \ \} \\
 & a^* \sqsubseteq y \ \wedge \ b \sqsubseteq y \\
\equiv & \qquad \{ \ (\Rightarrow) \quad \bullet \quad y \circ y \sqsubseteq y \\
 & \qquad\quad (\Leftarrow) \quad \bullet \quad I \sqsubseteq a^* \ \} \\
 & a^* \sqsubseteq y \ \wedge \ b \circ a^* \sqsubseteq y \\
\equiv & \qquad \{ \ \bullet \quad y \text{ is reflexive and transitive, (8.13)} \ \} \\
 & a^* \sqsubseteq y \ \wedge \ (b \circ a^*)^* \sqsubseteq y \\
\equiv & \qquad \{ \ (\Rightarrow) \quad \bullet \quad y \circ y \sqsubseteq y \\
 & \qquad\quad (\Leftarrow) \quad \bullet \quad I \sqsubseteq a^* \ \} \\
 & a^* \circ (b \circ a^*)^* \sqsubseteq y \quad .
\end{array}
$$

It remains to prove that $a^* \circ (b \circ a^*)^*$ is reflexive and transitive. At a glance it is reflexive. For transitivity we have:

$$
\begin{array}{ll}
 & a^* \circ (b \circ a^*)^* \circ a^* \circ (b \circ a^*)^* \;\; = \;\; a^* \circ (b \circ a^*)^* \\
\Leftarrow & \qquad \{ \;\; (8.23) \text{ with } x \;:=\; b \circ a^*, \text{ Leibniz } \;\; \} \\
 & a^* \circ (b \circ a^*)^* \circ a^* \;\; = \;\; a^* \circ (b \circ a^*)^* \\
\equiv & \qquad \{ \;\; \text{leapfrog rule: (8.44), with } a \;:=\; a^* \;\; \} \\
 & (a^* \circ b)^* \circ a^* \circ a^* \;\; = \;\; a^* \circ (b \circ a^*)^* \\
\equiv & \qquad \{ \;\; (8.23) \text{ with } x \;:=\; a \;\; \} \\
 & (a^* \circ b)^* \circ a^* \;\; = \;\; a^* \circ (b \circ a^*)^* \\
\equiv & \qquad \{ \;\; \text{leapfrog rule: (8.44), with } a \;:=\; a^* \;\; \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

Note that all elements of the structure of a regular algebra are used in this proof of star-decomposition.

One reason that the leapfrog rule (8.44) and the star decomposition rule (8.37) are so important — particularly in computing science — is that a number of common programming problems fit into the abstract framework of a regular algebra, and their solution is readily formulated using the two rules. Examples of such problems are several path-finding problems [7, 27, 10], but their discussion is beyond the scope of this text.

**Exercise 8.45** One might speculate whether the leapfrog rule holds with the closure star replacing the Kleene star. That is, is it the case that, for all monotonic functions $f$ and $g$,

(8.46) $\quad (f \bullet g)^\star \bullet f \quad = \quad f \bullet (g \bullet f)^\star \;\;$ ?

Also, if one combines the star-decomposition rule with the leapfrog rule one obtains an alternative form of the star-decomposition rule, namely:

(8.47) $\quad (a \sqcup b)^* \quad = \quad (a^* \circ b)^* \circ a^* \quad ,$

and one might speculate whether this rule remains valid when the Kleene star is replaced by the closure star, i.e. whether or not, for all monotonic functions $f$ and $g$,

(8.48) $\quad (f \,\dot\sqcup\, g)^\star \quad = \quad (f^\star \bullet g)^\star \bullet f^\star \quad .$

Investigate whether (8.46) and/or (8.48) is valid. (Note that if a counterexample to (8.48) can be constructed then both are false, but the falsehood of (8.46) does not necessarily imply that (8.48) is false.)

□

**Exercise 8.49 (Conway's Factor Matrix)** J.H. Conway [30] was the first to introduce factors in the context of regular languages. The chapter on factors in his book introduces the notion of the "factor matrix" of a language and shows how it can be used to approximate a language by other languages. In the following chapter he also uses factors in combination with so-called "biregulators" to prove that various operations on languages preserve the property of being regular. Conway's style of proof in the chapter on factors is very wordy and certainly not calculational. This is rectified somewhat in the chapter on biregulators (an algebra of transducers) where his techniques for proving regularity-preserving properties are particularly effective.

This exercise enables you to reconstruct, in a calculational style, the main properties of Conway's factor matrix. Some preparatory definitions are necessary.

Conventionally a matrix has a finite number of elements. This is also the case for Conway's factor matrix since he restricted his attention to regular languages. We do not want to make that restriction and since, in the current circumstances, there is no reason to restrict matrices to a finite number of elements we shall not do that either.

The definition we use here of a *square matrix over* $\mathcal{A}$ *indexed by a* (non-empty) *index set* $\mathcal{I}$ is simply a function $\mathbf{M} \in \mathcal{A} \longleftarrow \mathcal{I} \times \mathcal{I}$. If $i$ and $j$ are elements of $\mathcal{I}$ then the application of $\mathbf{M}$ to $(i,j)$ is denoted $i\,\mathbf{M}\,j$ and is called *the* $(i,j)$th *element of* $\mathbf{M}$.

Now suppose $(\mathcal{A}, \sqsubseteq, \circ, I)$ is a regular algebra. (So $(\mathcal{A}, \sqsubseteq)$ is a complete lattice and $(\mathcal{A}, \circ, I)$ is a monoid.)

Note that associativity of composition is equivalent to

(8.50) $X\backslash(Y/Z) = (X\backslash Y)/Z$ ,

for all $X$, $Y$ and $Z \in \mathcal{A}$. Moreover, that $I$ is a left unit of composition is equivalent to

(8.51) $I\backslash X = X$

for all $X \in \mathcal{A}$, and that it is a right unit to

(8.52) $X/I = X$

for all $X \in \mathcal{A}$. Property (8.50) permits one to drop the parentheses and write $X\backslash Y/Z$, which we do from now on. (Note however that $(X/Y)\backslash Z \neq X/(Y\backslash Z)$ in general!)

Let $\mathcal{I}$ be a non-empty set and consider the set of all square matrices over $\mathcal{A}$ with index set $\mathcal{I}$. It is easy to show that this set forms a regular algebra with the following definitions (whereby $i$, $j$ and $k$ range over $\mathcal{I}$):

$$\begin{array}{rcl}
\mathbf{M} \sqsubseteq \mathbf{N} & \equiv & \forall(i,j :: \ i\,\mathbf{M}\,j \ \sqsubseteq \ i\,\mathbf{N}\,j) \\
i(\mathbf{M} \circ \mathbf{N})k & = & \sqcup.(j : \ j \in \mathcal{I} : \ i\,\mathbf{M}\,j \circ j\,\mathbf{N}\,k) \quad , \qquad \text{and} \\
i\,\mathbf{I}\,j & = & I \qquad \text{if} \ \ i = j \\
 & & \perp\!\!\perp \qquad \text{otherwise} \ \ .
\end{array}$$

(The verification of this assertion you may regard as part 0 of this exercise.) In this algebra, we say that matrix $\mathbf{M}$ is *reflexive* if $\mathbf{I} \sqsubseteq \mathbf{M}$ and *transitive* if $\mathbf{M} \circ \mathbf{M} \sqsubseteq \mathbf{M}$ . This is, of course, the standard definition of reflexivity and transitivity.

Now let $E$ denote a fixed element of $\mathcal{A}$. (We use "$E$" as did Conway to help the reader to relate the properties stated here to those in Conway's book.) Conway defines a *factor* of $E$ to be any element of $\mathcal{A}$ that can be expressed in the form $X\backslash E/Y$ for some $X$ and $Y$. He calls an element of $\mathcal{A}$ a *left factor* of $E$ if it can be expressed in the form $E/Y$ for some $Y$ and a *right factor of* $E$ if it can be expressed in the form $X\backslash E$ for some $X$. The function $(X,Y \longmapsto X\backslash E/Y)$, where $X$ and $Y$ range over all elements of $\mathcal{A}$, thus forms a matrix of factors with index set $\mathcal{A}$ but this matrix is *not* Conway's factor matrix. Conway's factor matrix is a matrix indexed by *the left factors* (or equally the right factors) of $E$, this index set being finite in the case that $E$ is a regular language (as opposed to $\mathcal{A}$ which is infinite). In more detail, his theorem states that the factors of $E$ organise themselves into a reflexive, transitive matrix indexed by the left (or right) factors of $E$. Moreover, $E$ itself and all left and right factors of $E$ are elements of the matrix.

In this exercise we lead you step-by-step to a proof of this theorem.

**Step 1.** According to our definition of a matrix, the binary operators $\backslash$ and $/$, with domains restricted to $\mathcal{I} \times \mathcal{I}$ for arbitrary set $\mathcal{I}$, $\mathcal{I} \subseteq \mathcal{A}$ , are both square matrices over $\mathcal{A}$.

Show that, for arbitrary index set $\mathcal{I}$, both $\backslash$ and $/$ are reflexive and transitive matrices.

The clue we obtain from this step to the construction of the factor matrix is that it suffices to construct a suitable index set for the matrix $\backslash$ (or for the matrix $/$ ).

**Step 2.** Define the functions $\triangleleft$ and $\triangleright$ by

$$(8.53)\quad X\triangleleft \;=\; E/X \;\;,$$

$$(8.54)\quad X\triangleright \;=\; X\backslash E \;\;.$$

By definition, the range of $\triangleleft$ is the set of *left* factors of $E$ and the range of $\triangleright$ is the set of *right* factors of $E$.

Observe a Galois connection between $\triangleleft$ and $\triangleright$ and hence prove the following:

$$(8.55)\quad X\triangleleft\triangleright\triangleleft \;=\; X\triangleleft \;\;,$$

$$(8.56)\quad X\triangleright\triangleleft\triangleright \;=\; X\triangleright \;\;,$$

$$(8.57)\quad E\triangleleft\triangleright \;=\; E \;=\; E\triangleright\triangleleft \;\;.$$

This step records a (1-1) correspondence between left and right factors of $E$. Thus any matrix indexed by left factors can be mapped directly into a matrix indexed by right factors, and vice-versa. Moreover — property (8.57) — $E$ is both a left and right factor of itself.

Let $\mathcal{L}$ denote the set of left factors of $E$. The conclusion from steps 1 and 2 is that there are only two reasonable candidates for Conway's factor matrix, the matrix $\backslash$ indexed by $\mathcal{L}$ and the matrix $/$ also indexed by $\mathcal{L}$. After a moment's thought it is obvious that the latter matrix is uninteresting, so we consider the former.

**Step 3.** Define the *factor matrix* of $E$ to be the binary operator $\backslash$ restricted to $\mathcal{L}\times\mathcal{L}$. Thus entries in the matrix take the form $L_0\backslash L_1$ where $L_0$ and $L_1$ are left factors of $E$. By step 1 this is a reflexive, transitive matrix. Also, by definition of a left factor and a factor, all entries in the matrix are factors of $E$.

Suppose that $F$ is a factor, $L$ is a left factor, and $R$ is a right factor of $E$. Construct left factors $L_0$, $L_1$, $L_2$, $L_3$, $L_4$ and $L_5$ such that

$$(8.58)\quad F \;=\; L_0\backslash L_1 \;\;,$$

$$(8.59)\quad L \;=\; L_2\backslash L_3 \;\;, \qquad \text{and}$$

$$(8.60)\quad R \;=\; L_4\backslash L_5 \;\;.$$

(Hint: observe that (8.50) with $Y$ instantiated to $E$ gives the identity

$$(8.61)\quad X\backslash(Z\triangleleft) \;=\; (X\triangleright)/Z \;\; .$$

Combine this with (8.55), (8.56) and (8.57).)

From your constructions of $L_2$–$L_5$ satisfying (8.59) and (8.60) you should observe that $L_2$ is independent of $L$ and $L_5$ is independent of $R$. Prove that

$$(8.62)\quad E \;=\; L_2\backslash L_5 \;\; ,$$

and show that, for all $X$,

$$\begin{array}{lll}(8.63) & & X \text{ is a left factor of } E \\ & \equiv & \exists(L:\; L \text{ is a left factor of } E:\; X \;=\; L_2\backslash L) \;\; ,\end{array}$$

and

$$\begin{array}{lll}(8.64) & & X \text{ is a right factor of } E \\ & \equiv & \exists(L:\; L \text{ is a left factor of } E:\; X \;=\; L\backslash L_5) \;\; .\end{array}$$

This completes the proof of Conway's theorem. A matrix has been exhibited containing all factors and only the factors of $E$, indexed by left factors of $E$, that is reflexive and transitive. The import of (8.63) and (8.64) is that a "row" of the matrix (a set of entries all having the same first index) contains all (and only) the left factors of $E$, and a "column" of the matrix (a set of entries all having the same second index) all (and only) the right factors of $E$. In addition, from (8.62) we see that $E$ is the matrix entry at the intersection of this row and column. (Note, however, that factors and left and right factors of $E$, including $E$ itself, may appear repeatedly in the matrix. Conway's wordy theorems and proofs are confusing on this point and there is one unfortunate misprint that claims exactly the opposite!)

The factor matrix crops up surprisingly often. In [2] it was shown that, in the context of language theory (specifically, where elements of $\mathcal{A}$ are sets of words on which is defined a length function), there exists a unique least matrix, dubbed the *factor graph*, whose reflective, transitive closure is the factor matrix of a given language $E$. Further, in [9] it was shown that several pattern-matching algorithms, including the well-known Knuth-Morris-Pratt algorithm [56], boil down to constructing the factor graph of a language defined by the given patterns.

□

**Exercise 8.65** The route taken in exercise 8.49 to the construction of the factor matrix is a direct one but there are several detours that one take on the way. This exercise records a couple.

From the Galois connection between $\lhd$ and $\rhd$ constructed in exercise 8.49 you will have observed that both $\lhd\!\rhd$ and $\rhd\!\lhd$ are closure operators. This will help you to solve the following question.
Show that, for all $X$ and $Y$,

(8.66) $X \circ Y \sqsubseteq E \quad \equiv \quad X{\rhd\!\lhd} \circ Y{\lhd\!\rhd} \sqsubseteq E$ .

Using this (or otherwise) prove that:

(8.67) $(X \circ Y){\rhd} \quad = \quad (X{\rhd\!\lhd} \circ Y){\rhd}$ .

The dual of (8.67) is

(8.68) $(X \circ Y){\lhd} \quad = \quad (X \circ Y{\lhd\!\rhd}){\lhd}$ .

These two formulae are slightly more general than (8.55) and (8.56): the former is obtained by instantating $Y$ to $I$ in (8.67) and the latter by instantiating $X$ to $I$ in (8.68).
□

## 8.5 Concluding Remarks

The principal results in this section — that $(a\circ)^\star.I$ is the reflexive, transitive closure of $a$, and the star-decomposition and leapfrog rules — are standard, but our approach to them is not. (The standard way of defining $(a\circ)^\star.b$ is as the least fixed point of the function $(x \mapsto b \sqcup a \circ x)$. That is,

(8.69) $(a\circ)^\star.b \quad = \quad b \sqcup a \circ (a\circ)^\star.b$

and

(8.70) $(a\circ)^\star.b \sqsubseteq y \quad \Leftarrow \quad b \sqcup a \circ y \sqsubseteq y$ .)

Why, one is entitled to ask, should we pay so much attention to well-known facts, and what is the justification for such idiosyncracy?

The answer to these questions cannot be given completely at this stage and the reader must exercise some patience. We can however hint at some reasons.

The Kleene star is often used to denote the list constructor. Thus if $a$ denotes a type then $a^*$ denotes all lists with elements drawn from $a$. The choice of the same symbol to denote the reflexive, transitive closure operator on relations and the list type constructor is not accidental but motivated by agreements between their properties. In particular, just as there are three distinct ways to define the reflexive, transitive closure operator, there are three distinct ways to define lists. One is to define the so-called "cons" lists, another is to define "snoc" lists and a third is to define "join" lists. Cons lists are constructed from the empty list by appending ("consing") elements to the head of a list, whilst snoc lists are constructed from the empty list by appending ("snocing") elements to the tail of a list. Join lists consist of the empty list, singleton lists (i.e. lists of length one), or are formed by appending ("joining") two lists to each other, the join operation being by definition associative and having the empty list as unit.

These three list constructors are isomorphic in the sense that there are bijections mapping join lists to snoc lists and cons lists. The informal descriptions we have just given of the three type constructors also bear a strong resemblance to the three different ways we have presented of defining the reflexive transitive closure $a^*$ — if one reads "$\circ$" as "append". The description of join lists has the appearance of the direct definition, whilst the description of cons and snoc lists is suggestive of the two indirect definitions, cons lists corresponding to $(a\circ)^\star.I$ and snoc lists to $(\circ a)^\star.I$ .

The question arises whether the proofs of the equality of $a^*$, $(a\circ)^\star.I$ and $(\circ a)^\star.I$ can be somehow adapted to constructions of the bijections between join, snoc and cons lists. Other questions also suggest themselves: can we give a constructive interpretation to the star-decomposition rule and to the leapfrog rule (for example), and if so can the proofs we have given be adapted to the construction of interesting and useful programs. More generally, can we give the theory of closure operators a constructive interpretation that enables us to derive useful and interesting programs on type structures other than lists.

As we shall see, the answer is yes. For example, the constructive interpretation of the star-decomposition rule is a problem known as the "lines-unlines problem" [22] (the problem of splitting a paragraph of text into several lines and separator symbols). The two different proofs discussed here lead to two different solutions to the lines-unlines problem. The step from these proofs to constructive proofs coincides with the step from Galois connections to the categorical notion of an "adjunction". But, at this stage in the presentation, insufficient theory has been developed to permit us to make that step.

**Note** At the time of writing (August 1992) the above remarks should be regarded as an objective rather than an accomplished fact. We have done sufficient work to convince us that the objective is attainable but that work is as yet not properly documented and incomplete.

# Part II

# Theory of Datatypes

# Chapter 9

# The Algebraic Framework

A major component of our endeavour is the development of a calculus of programming that permits and, indeed, encourages clear and economical calculation. For this we need an elegant algebraic setting. Although from the mathematical point of view, there is nothing wrong with a standard set-theoretic approach nor with the algebraically more attractive predicate calculus, we are dissatisfied with the persistent appearance of arguments and dummies in those systems. This invites us to look for a setting one abstraction level higher that fits our manipulative needs.

In order to choose such an abstract setting ("syntax" for short) several design criteria should be established. Here some of ours are mentioned, not as dictates but just for the sake of clarifying our point of view.

- The syntax should reflect the structure of the everyday mathematical view of relations as tightly as possible (excluding historical oddities, inelegancies and prejudice).

- The syntax should be built up in layers. If possible, those layers should be well-known syntactical unities with proven "elegance".

- The meta-language used for juggling with the syntax is the predicate calculus.

- There should be a clear distinction between terms in the meta-language and terms in the syntax.

Fortunately we don't have to start from scratch. The road towards an "axiomatic theory of relations" is already paved with the pioneering work of Tarski [88]. Besides, the above point of view is apparent in most of the curricula nowadays, be it not always explicit. Without further ado we present the most basic part of the syntax. In section 12 this syntax is supplemented by axioms for elementary data types.

## 9.1 The Setting

### 9.1.1 Plat Calculus and the Knaster-Tarski Theorem

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 [23]. By "complete lattice" we mean that the extremums

$$\begin{array}{ll} & \sqcup(R:\ R\in\mathcal{V}:\ R) \\ \text{and} & \sqcap(R:\ R\in\mathcal{V}:\ R) \end{array}$$

exist for all bags of specs $\mathcal{V}$. "Completely distributive lattice" means that

$$\begin{array}{lllll} & R\ \sqcap & \sqcup(S:\ S\in\mathcal{V}:\ S) & = & \sqcup(S:\ S\in\mathcal{V}:\ R\sqcap S) \\ \text{and} & R\ \sqcup & \sqcap(S:\ S\in\mathcal{V}:\ S) & = & \sqcap(S:\ S\in\mathcal{V}:\ R\sqcup S) \end{array}$$

for all specs $R$ and all bags of specs $\mathcal{V}$. 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.

Among the more significant properties of such a structure is the (well-known) "Knaster-Tarski fixpoint theorem". Since we shall use the theorem frequently we summarise it here (to the extent and in the form appropriate to our own needs). Specifically, it says that, for arbitrary monotonic function $\theta$, the equation

$$X :: \quad X = \theta.X$$

has a smallest solution, which henceforth we denote by $\mu\theta$, characterised by the two properties:

$$\mu\theta \;=\; \theta.\mu\theta$$

and, for all $X$,

$$X \sqsupseteq \mu\theta \;\Leftarrow\; X \sqsupseteq \theta.X$$

Moreover, such an equation also has a largest solution, which henceforth we denote by $\nu\theta$, characterised by the properties:

$$\nu\theta \;=\; \theta.\nu\theta$$

and, for all $X$,

$$X \sqsubseteq \nu\theta \;\Leftarrow\; X \sqsubseteq \theta.X$$

For an excellent account of plat calculus (although that name is not used!), including a modern proof of the Knaster-Tarski theorem and a clear and careful exposition of its implications, we would recommend the reader to refer to [36].

### 9.1.2 Composition and Factors

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: composition is universally cup-distributive. I.e. for bags of specs $\mathcal{V}, \mathcal{W} \subseteq \mathcal{A}$,

$$(\sqcup\mathcal{V}) \circ (\sqcup\mathcal{W}) \quad = \quad \sqcup(P,Q: \ \ P \in \mathcal{V} \wedge Q \in \mathcal{W}: \ \ P \circ Q)$$

In particular,

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

Another, less immediate and somewhat unfamiliar consequence of this interface, is the existence of so-called "left" and "right factors" defined as follows.

**Definition 9.1** For specs $R$ and $S$ we define the *right factor* $R\backslash S$ by

(a) $\quad R\backslash S \sqsupseteq X \quad \equiv \quad S \sqsupseteq R \circ X$

and the *left factor* $S/R$ by

(b) $\quad S/R \sqsupseteq X \quad \equiv \quad S \sqsupseteq X \circ R$

□

Left and right factors are thus defined to be the largest solutions to inequations in a variable $X$ (the inequation to the right of the equivalence in their respective definitions). Although we shall have no use for it here we mention that the operators "\" and "/" associate with each other (i.e. $P\backslash(Q/R) \ = \ (P\backslash Q)/R$), thus justifying writing $P\backslash Q/R$ and that such is a *factor* of $Q$.

Equations (9.1a) and (9.1b) are instances of what are known as "Galois connections". (See the appendix for further discussion.) Our use of the word "factor" is intended to suggest an analogy between composition and multiplication, and between factoring and division. This analogy is further reinforced by the following easily derived *cancellation properties* of factors.

**Lemma 9.2 (Factor Cancellation)**

(a) $S \sqsupseteq R \circ (R\backslash S)$
(b) $R \sqsupseteq (R/S) \circ S$
(c) $R\backslash(R \circ S) \sqsupseteq S$
(d) $(R \circ S)/S \sqsupseteq R$
(e) $R \circ R\backslash(R \circ S) = R \circ S$
(f) $(R \circ S)/S \circ S = R \circ S$
(g) $R\backslash(R \circ R\backslash S) = R\backslash S$
(h) $(R/S \circ S)/S = R/S$
$\square$

Evidence for the claim that definitions (9.1a) and (9.1b) and, in particular, the calculational possibilities they admit are important but not well known is the fact that they have surfaced in various guises and under various names over the last fifty years beginning, to our knowledge, with [37] (under the names left and right "residuals") and involving diverse application areas such as the structure of natural language [57], regularity properties of generalised-sequential machines [30] (under the name used here of left and right "factors"), the well-known Knuth-Morris-Pratt string searching algorithm [9], and program specification [52] (under the names "weakest pre- and post-specification"). We prefer Conway's [30] more anonymous terminology to that used by Hoare and He [52]. The term "residual", which is also used by Birkhoff [23], would have been equally acceptable. Note, however, that of the above-referenced works, Hoare and He's calculational formulation of the properties of "factors" is the single most significant contribution to the present work.

*Remark* In addition to the use of different terminology our choice of notation is exactly opposite to Hoare and He's: they would write $S/R$ where we write $R\backslash S$, and vice-versa $R\backslash S$ where we write $S/R$. Our own choice of notation is justified by the — for us very important — property that in the use of lemma 9.2 the "cancelled" expressions are adjacent. We reject outright the notation adopted by Birkhoff [23] as unsystematic and inappropriate to compact calculation. *End of Remark*

### 9.1.3 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.

(We are not alone in purposefully adopting a syntactically ambiguous notation, although the practice is sometimes frowned on. DeMorgan [32] is an outstanding precedent. He writes "not-$L$-verse" where we write $\neg L\cup$. See [61] for detailed references and citations from DeMorgan's work.)
*End of Remark*

The interface with the second layer is given by the two equations:

$$(R \circ S)\cup \quad = \quad S\cup \circ R\cup$$

and $$I\cup \quad = \quad I$$

### 9.1.4 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 the two factoring operators ("$/$" and "$\backslash$") 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.

### 9.1.5 The Exchange and Rotation Rules

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

**The Middle Exchange Rule**

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

The rule is so named because the middle term on the right side is exchanged with the left side of the inequality.

There are several variations on the rule. The "left" and "right" exchange rules are obtained by instantiating, respectively, $P$ and $Q$ to $I$ and simplifying.

**The Left Exchange Rule**

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

**The Right Exchange Rule**

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

The "rotation rule" is obtained by making the substitutions $Y := R^{\cup}$, $P := S$, $X := \neg T$ and $Q := I$ and again simplifying.

**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.

It is our experience that the middle exchange 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.

$$\begin{array}{lll} T & \sqsupseteq & R \circ S \\ \neg S & \sqsupseteq & R^{\cup} \circ \neg T \\ \neg R & \sqsupseteq & \neg T \circ S^{\cup} \end{array}$$

To counter these arguments we would point out that the middle exchange 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 middle exchange 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 rule relating factors, reverse and the complement operator. This alternative is discussed further in the appendix.

(In point of fact, Maddux [61] observes that the so-called "Schröder" rule was stated much earlier by De Morgan. Schröder subsequenty elaborated on the rule, listing all possible variations on the rule with three variables and extending it to "relative addition", $R \dagger S$, defined by $R \dagger S \;=\; \neg(\neg R \circ \neg S)$.)

## 9.2 Models

Various models of the above axioms are discussed in the appendix with regard to the following questions:

(a) Are the layers and axioms independent?

(b) Are the successive extensions conservative?

(c) Does the axiomatisation characterise the set-theoretic relations completely?

Here we shall content ourselves with a summary of the conclusions, namely: the set-theoretic relations do indeed form a model of the axiom system but the axiom system is not complete for this model; the middle exchange rule and the cone rule (discussed in section 12.1.1) are independent of the other axioms but the reverse structure is not.

A final comment with regard to the idiosyncracies of our naming conventions. The following sections must serve a dual purpose. The technical aim is to build up a theory of types based upon the above syntax. To do this in a way that is evidently free from logical inconsistencies necessitates making a clear distinction between the theory itself and the metalanguage. For this reason we have chosen to call elements of $\mathcal{A}$ "specs" rather than "relations" and to use the symbols "$\sqcap$" and "$\sqcup$" etc. rather than "$\cap$" and "$\cup$" etc. To serve the second purpose we intersperse the development with references to the relational model. The reader may prefer to construct their own proofs of the various lemmas, theorems etc. in this one interpretation, but they do so at their own peril.

# Chapter 10

# Foundations

The purpose of this section is to build up a vocabulary for our later discussion of polynomial relators and relational catamorphisms. 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.*

## 10.1 Monotypes

The notion of a guard as a primitive entity in a programming language was first introduced in Dijkstra's guarded command language [34]. It is a useful notion since it is more flexible than the older, more conventional notion of a conditional statement. Its particular merit is that it introduces partiality into programs and at the same time facilitates the introduction of indeterminacy thereby streamlining the derivation of programs.

A guard acts as a filter on the domain of execution of a statement. Operationally it can be viewed as a partial skip. Mathematically, a guard is just a device that enables sets — subsets of the set of all states — to be incorporated into program statements.

In the spec calculus there are two mechanisms for viewing sets as specs, and thus modelling guards, each of which has its own merits. The first is via so-called "monotypes", the second via "conditions". Axiomatically, these have the following definitions. First: we say that spec $A$ is a *monotype* iff $I \sqsupseteq A$. Second: we say that spec $p$ is a *right condition* iff $p = \top\!\top \circ p$. The dual notion of *left* condition is obtained by reversing the positions of $\top\!\top$ and $p$ in the left side of the defining equation.

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

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

The *right condition* boolean is the relation

$$\{(x,\, \mathbf{true}),\, (x,\, \mathbf{false}) \mid x \in \mathbb{U}\}$$

It is clear that for any given universe $\mathbb{U}$ there is a one-to-one correspondence between the subsets of $\mathbb{U}$ and the monotypes. Specifically, the set $A$ is represented by the monotype $\underline{A}$ where $x\underline{A}y \;\equiv\; x = y \in A$. Equally clear is the existence of a one-to-one correspondence between the subsets of $\mathbb{U}$ and the right conditions on $\mathbb{U}$ . That is, if $A$ is some set then the right condition defined by $A$ is that relation $A_r$ such that for all $x$ and $y$, $xA_ry \;\equiv\; y \in A$. Similarly, the left condition corresponding to $A$ is that relation $A_l$ such that for all $x$ and $y$, $xA_ly \;\equiv\; x \in A$.

Using monotypes to represent subsets of $\mathbb{U}$ as specs a guard on a spec is modelled by composition of the spec, either on the left or on the right, with such a monotype. Thus, if $R$ and $S$ are specs and $A$ is a monotype then $A \circ R$ and $S \circ A$ are both specs, the first being spec $R$ after restricting elements in its left domain to those in $A$, and the second being the spec $S$ after restricting elements in its right domain to those in $A$. Using conditions a guard on the left domain of spec $R$ is modelled by the intersection of $R$ with a left condition, and a guard on the right domain of $R$ by its intersection with a right condition. In principle, this poses a dilemma in the choice of representation of guards in the spec calculus. Should one choose monotypes or conditions?

We choose monotypes — there being several reasons for doing so. One is the simple fact that guarding both on the left and on the right of a spec is accomplished in one go with monotypes whereas demands two sorts of conditions (left and right conditions). Moreover, they have very simple and convenient properties. Specifically, for all monotypes $A$ and $B$

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

(10.2) $A \circ B \;=\; B \circ A \;=\; A \sqcap B$

and for all bags of monotypes $\mathcal{B}$,

$$(10.3)\quad \top\!\!\top \circ \sqcap\mathcal{B} \;=\; \sqcap(\top\!\!\top \circ \mathcal{B})$$

The most compelling reason, however, for choosing to represent sets by monotypes is the dominant position occupied by composition among programming primitives. Introducing a guard in the middle of a sequential composition of specs is a frequent activity that is easy to express in terms of monotypes but difficult to express with conditions.

Nevertheless conditions do have their place from time to time. They too have attractive calculational properties. From the above it is clear that there is a one-to-one correspondence between monotypes and both types of condition (documented formally below). Exploitation of this correspondence is central to many calculations in the spec calculus.

Monotypes are obviously closed under $\sqcup$. They are not, however, closed under $\sqcap$ or $\neg$. (A *non-empty* intersection of monotypes is a monotype but the empty intersection is, by definition, $\top\!\!\top$ which is not a monotype. The complement of a monotype is never a monotype.) Nevertheless, with suitably adapted $\sqcap$ and $\neg$ operators the monotypes do form a complete, completely distributive lattice, albeit not a sub-lattice of the spec lattice. The clue to its construction lies in the fact that the conditions *do* form a sub-lattice of the spec and are in one-to-one correspondence with the monotypes.

Right conditions are closed under negation: for all specs $p$,

$$\begin{array}{cl}
 & \neg p \;=\; \top\!\!\top \circ \neg p \\
\equiv & \qquad \{\ \neg p \sqsubseteq \top\!\!\top \circ \neg p\ \} \\
 & \neg p \;\sqsupseteq\; \top\!\!\top \circ \neg p \\
\equiv & \qquad \{\ \text{right-exchange rule}\ \} \\
 & p \;\sqsupseteq\; \top\!\!\top^{\cup} \circ p \\
\equiv & \qquad \{\ \top\!\!\top^{\cup} = \top\!\!\top,\quad p \sqsubseteq \top\!\!\top \circ p\ \} \\
 & p \;=\; \top\!\!\top \circ p
\end{array}$$

They are closed under cup: for all sets of right conditions $\mathcal{P}$

$$\begin{array}{cl}
 & \sqcup\mathcal{P} \\
= & \qquad \{\ \bullet\ \ \mathcal{P} \text{ is a set of conditions: definition of condition}\ \} \\
 & \sqcup(\top\!\!\top \circ \mathcal{P}) \\
= & \qquad \{\ \text{universal } \sqcup\text{-junctivity of composition}\ \} \\
 & \top\!\!\top \circ \sqcup\mathcal{P}
\end{array}$$

and under cap: for all sets of right conditions $\mathcal{P}$,

$$
\begin{array}{ll}
 & \sqcap\mathcal{P} \\
\sqsubseteq & \quad \{\ I \ \sqsubseteq\ \top\!\!\top\ \} \\
 & \top\!\!\top \circ \sqcap\mathcal{P} \\
\sqsubseteq & \quad \{\ \text{monotonicity}\ \} \\
 & \sqcap(\top\!\!\top \circ \mathcal{P}) \\
= & \quad \{\ \bullet\ \ \mathcal{P} \text{ is a set of conditions: definition of condition}\} \\
 & \sqcap\mathcal{P}
\end{array}
$$

In summary, the right conditions form a power set lattice with top and bottom $\top\!\!\top$ and $\perp\!\!\perp$, respectively, and meet, join and complement operators the standard spec operators $\sqcap$, $\sqcup$ and $\neg$ .

Henceforth we shall always denote monotypes by the capital letters $A$, $B$ or $C$. Conditions will be denoted by the lower case letters $p$, $q$ or $r$.

## 10.2 Left and Right Domains

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 first, domains of specs are monotypes: for all specs $R$,

$$(10.4)\quad monotype.R{<} \qquad \text{and} \qquad monotype.R{>}$$

(Note that the infix dot denotes function application and that unary operators always take precedence in our formulae over binary operators. Thus you should parse "$monotype.R{<}$" as "$monotype.(R{<})$".) Second, the domain operators are defined by a Galois connection between the lattice of all specs and the sublattice of the monotypes: For all specs $R$ and monotypes $A$,

$$(10.5)\quad A \sqsupseteq R{>} \ \equiv\ \top\!\!\top \circ A \sqsupseteq R$$

and

$$(10.6)\quad A \sqsupseteq R{<} \ \equiv\ A \circ \top\!\!\top \sqsupseteq R$$

According to a general theorem on Galois connections it follows that the domain operators are universally $\sqcup$-junctive. In particular, for all specs $R$ and $S$,

$$(10.7)\quad (R \sqcup S)< \;=\; R< \;\sqcup\; S<$$

$$(10.8)\quad (R \sqcup S)> \;=\; R> \;\sqcup\; S>$$

An additional consequence is that "<" and ">" are monotonic.

Consequences of the specific form of (10.5) and (10.6) are the one-to-one corespondences between monotypes and left and right conditions mentioned several times earlier: for all specs $R$,

$$(10.9)\quad \top\!\!\top \circ R> \;=\; \top\!\!\top \circ R \quad\text{and}\quad (\top\!\!\top \circ R)> \;=\; R>$$

$$(10.10)\quad R< \circ \top\!\!\top \;=\; R \circ \top\!\!\top \quad\text{and}\quad (R \circ \top\!\!\top)< \;=\; R<$$

In particular, for all right conditions $p$ and monotypes $A$,

$$(10.11)\quad \top\!\!\top \circ p> \;=\; p \quad\text{and}\quad (\top\!\!\top \circ A)> \;=\; A$$

Relational calculus 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. For all monotypes $A$ and all specs $R$,

$$(10.12)\quad A \circ R = R \;\equiv\; A \sqsupseteq R<$$

$$(10.13)\quad R \circ A = R \;\equiv\; A \sqsupseteq R>$$

The following properties of "<" also prove to be very useful. For all specs $R$ and $S$,

$$(10.14)\quad R< \;=\; (R^{\cup})>$$

$$(10.15)\quad R< \circ S \;=\; R \circ \top\!\!\top \;\sqcap\; S$$

$$(10.16)\quad (R \circ S)< \;=\; (R \circ S<)<$$

$$(10.17)\quad R< \;\sqsupseteq\; (R \circ S)<$$

$$(10.18)\quad (R \sqcap S \circ T)< \;=\; (R \circ T^{\cup} \;\sqcap\; S)<$$

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

$$(10.19)\quad R> \;=\; (R^{\cup})<$$

$$(10.20)\quad S \circ R> \;=\; \top\!\!\top \circ R \;\sqcap\; S$$

$$(10.21)\quad (R \circ S)> \;=\; (R> \circ S)>$$

$$(10.22)\quad S> \;\sqsupseteq\; (R \circ S)>$$

$$(10.23)\quad (R \sqcap S \circ T)> \;=\; (S^{\cup} \circ R \;\sqcap\; T)>$$

Of these five pairs of properties, four are evident when specs are interpreted as relations. One pair, properties (10.15) and (10.20), 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.
We sometimes write

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

as a synonym for

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

It is immediate from (10.12) and (10.13) that

(10.25) $R< \circ R \quad = \quad R \quad = \quad R \circ R>$

Indeed this law is used so frequently that, after a while, we hardly bother to mention it. Using the notation we have just introduced (10.25) can be rephrased in the form

$$R \in R< \sim R>$$

Note that (10.24) 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 (10.2) with $B$ instantiated to $A$ that, for all monotypes $A$,

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

and, more specifically,

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

Properties (10.14), (10.19) and (10.27) 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.

Finally, 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.

## 10.3 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 10.28**

(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 *to* $A$ *from* $B$ 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 to $A$ from $B$.

**Theorem 10.29** 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 10.30** If $f$ is an imp then, for all non-empty sets of specs $\mathcal{V}$,

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

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 10.31** If $f$ is a co-imp then, for all non-empty sets of specs $\mathcal{V}$,

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

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 10.32** 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 (10.32a) 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 (10.32b) 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 later in this section. 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,

$$A \sim B \quad \supseteq \quad A \longleftarrow B$$

and, for imp $f$,

$$f \in f< \longleftarrow f>$$

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

Note also that, for monotypes $A$, $B$ and $C$,

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

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 [63].)

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.

The operators $\longleftarrow$ and $\sim$ are the first examples of several typing operators introduced throughout the paper. All such operators are indicated by some sort of arrow and/or wavy line. (Other examples are $\dot{\leftsquigarrow}$ and $\leftrightsquigarrow$.) These are always used independently of the inclusion operators in the plat calculus and have the same precedence.

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 [69].)

## 10.4 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, [65]. In this subsection we propose a modest extension to the notion of functor to which we give the name "relator".

By rights, now is the time at which we should attempt to motivate this extension. This we shall not do, however, since the whole paper itself is the motivation for the proposed extension! Suffice it to say at this point that our definition arose by distilling the minimum additional requirements needed to guarantee that a functor be "naturally polymorphic" according to the definition given in section 11.2 of this paper. This was followed by a (successful) painstaking investigation — reported here — of whether those requirements were sufficient to enable us to verify a substantial number of other properties that we deemed desirable.

**Definition 10.33** A *relator* is a function, $F$, from specs to specs such that

(a) $\quad I \;\sqsupseteq\; F.I$

(b) $\quad F.R \sqsupseteq F.S \;\;\Leftarrow\;\; R \sqsupseteq S$

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

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

□

In view of (10.33d) we take the liberty of writing simply "$F.R{\cup}$" without parentheses, thus avoiding explicit use of the property.

The above ostensibly defines an *endo*relator, i.e. a unary relator from a given spec algebra $\mathcal{A}$ to itself. But we also wish to allow it to serve as the definition of a relator mapping specs of one spec algebra, $\mathcal{A}$ say, into another, $\mathcal{B}$ say. In particular we wish to use exactly the same definition for relators that map 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 (10.33c) 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 $j$, $1 \leq j \leq n$, 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.

One case that we make particular use of is when $F$ maps a pair of specs into a spec. (Both argument specs and the result spec are assumed, for the time being, to be in the same spec algebra.) 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) $\quad I \;\sqsupseteq\; I \otimes I$

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

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

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

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.

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

(To our knowledge there is no universally accepted name for what we have called an "abide" law even though examples are not difficult to find. A very familiar example is provided by multiplication and division in real arithmetic. Using a dot to denote multiplication and a horizontal bar to denote division we have:

$$\frac{u \cdot v}{x \cdot y} \quad = \quad \frac{u}{x} \cdot \frac{v}{y}$$

Another elementary example is furnished by addition and subtraction. We have $(u + v) - (x + y) = (u - x) + (v - y)$. (Somewhat tongue in cheek, this leads us to wonder whether this is the reason that both subtraction and division are denoted by a horizontal bar!) Hoare [51] depicts several abide laws in the predicate calculus in the same way, and we shall encounter others later in the text. In the category theory literature the term "interchange" rule (or law) is used.)

As already announced relators commute with the domain operators.

**Theorem 10.34** If $F$ is a relator then

(a) $\quad F.(R{>}) \quad = \quad (F.R){>}$

(b) $\quad F.(R{<}) \quad = \quad (F.R){<}$

$\Box$

For the proof of this theorem see the appendix.

In view of theorem 10.34 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 10.35** If $F$ is a relator then

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

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

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

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

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

**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 10.34.
$\square$

## 10.5 $\sqcap$-and $\sqcup$-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 10.36** 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 $\mathcal{I}$-$\sqcup$-*junctive* if for all $\mathcal{I}$-bags, $\mathcal{R}$,

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

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

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

The function $G$ is said to be $\mathcal{I}$-$\sqcup$-*continuous* if (10.37) holds for all non-empty, linear bags $\mathcal{R}$. The notion of $\mathcal{I}$-$\sqcap$-*continuity* is similarly defined.
□

Using the word "junctive" to stand for both "$\sqcap$-junctive" and "$\sqcup$-junctive", and "continuous" to stand for "$\sqcap$-continuous" and "$\sqcup$-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, $\mathcal{I}$.

Finally, the omission of any qualification on the word junctive means finitely junctive. (We reserve the shortest term for this case because it is the most commonly occurring and most important case.)

The relationship between these various types of junctivity properties is discussed in some depth by Dijkstra and Scholten [36] (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 $\sqcup$-junctive for all specs $R$. Moreover (see theorems 10.30 and 10.31), $(f\circ)$ is positively $\sqcap$-junctive for all co-imps $f$, and $(\circ f)$ is positively $\sqcap$-junctive for all imps $f$.

Just as we did for relators we shall apply definition 10.36 to functions $G$ that are not necessarily unary. When applied to non-unary functions there is a subtle nuance in the definition that may not be immediately evident. To clarify the matter let us spell out the definition in the case of a binary operator: $\mathcal{I}$-$\sqcap$-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 $\mathcal{I}$-$\sqcup$-junctivity.

Note that finite, positive $\sqcup$-junctivity of $\otimes$ is the same as saying that $\otimes$ abides with cup.

The nuance that we alluded to resides in the difference between "junctivity" and "distributivity" properties. In the case that a function is unary the two classes are indistinguishable. The examples just quoted are a case in point: we could equally well say, for example, that $(\circ f)$ is positively $\sqcap$-*distributive* for all imps $f$. The meaning is just the same. For functions of higher arity, in particular binary operators, there is a difference. To illustrate this consider the addition, division and multiplication operators in real arithmetic. We say that multiplication "distributes over" addition to express in words the law

$$(u+v)\cdot(x+y) \quad = \quad (u\cdot x)+(v\cdot x)+(u\cdot y)+(v\cdot y)$$

On the other hand, multiplication abides with division. I.e.

$$(u\cdot v)/(x\cdot y) \quad = \quad (u/x)\cdot(v/x)$$

In the terminology we have just introduced we would say that the binary division operator is multiplication-junctive, but is not multiplication-distributive. In contrast, the binary multiplication operator is addition-distributive but not addition-junctive. The notion of junctivity is more primitive because we may always define distributivity as coordinatewise junctivity. (A possible cause of confusion is that it is common to talk about a binary operator distributing over another when what is meant is that a unary operator formed by fixing one argument of a binary operator distributes over another binary operator. For example, it is common to summarise the law

$$(x+y)/z \quad = \quad (x/z)+(y/z)$$

for non-zero $z$ by saying that division distributes over addition. What is actually meant is that for each non-zero $z$ the unary operator $(/z)$ distributes over addition. The *binary* division operator is neither $+$-junctive nor $+$-distributive.)

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

# Chapter 11

# 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 the catamorphisms to be introduced in section 13. 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 [26] which work was anticipated by Reynolds [79]. Identifying definitions of "relator" and "catamorphism" that would guarantee their naturality was a major design goal of our work.

## 11.1 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 $(x,y)\in R$,

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

for all $R,S\in\overline{\mathcal{A}}$ and all $x,y$ and $z\in\mathcal{A}$. As discussed in the appendix, $\overline{\text{SPEC}}$ is, with these definitions, 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 identified 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

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

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

$$(11.1)\quad \forall(x,y::\ x\,\langle\overline{A}\rangle\,y \quad\equiv\quad x=y\ \wedge\ x\in A)$$

The operators "$\sim$" and "$\longleftarrow$" were defined in section 10.3 as set-forming operators. Using (11.1) 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 catamorphism 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 11.2** Let $F$ be a relator. Then, for all monotypes $A$ and $B$,

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

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

□

To understand these statements one must understand at what level each of the operators is being used. Theorem 11.2(a) is exemplary. Reintroducing the bar notation it states that

$$F \mathbin{\bar{\circ}} (A \sim B) \quad \overline{\in} \quad (F.A \sim F.B) \mathrel{\overline{\Longleftarrow}} (A \sim B)$$

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.

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

where

$$\begin{array}{lcl} P1 & \equiv & (F.A \sim F.B) \;\sqsupseteq\; F \circ (A \sim B) \circ (A \sim B)^{\cup} \circ F^{\cup} \\ P2 & \equiv & (F \circ (A \sim B))_{>} \;=\; A \sim B \end{array}$$

Property P1 is verified as follows:

$$\begin{array}{ll}
 & F.A \sim F.B \sqsupseteq F \circ (A \sim B) \circ (A \sim B)^{\cup} \circ F^{\cup} \\
\equiv & \qquad \{ \text{ definition of } \sqsupseteq \text{ at higher order } \} \\
 & \forall(R,S :\ R\langle F \circ (A \sim B) \circ (A \sim B)^{\cup} \circ F^{\cup}\rangle S \\
 & \qquad :\ R\langle F.A \sim F.B\rangle S) \\
\equiv & \qquad \{ \text{ definition of higher order operators,} \\
 & \qquad\quad \text{monotype.}(A \sim B) \ \} \\
 & \forall(R,S :\ \exists(T :\ T \in A \sim B :\ R = F.T \wedge S = F.T) \\
 & \qquad :\ R\langle F.A \sim F.B\rangle S) \\
\equiv & \qquad \{ \text{ calculus, definition of } A \sim B \ \} \\
 & \forall(T :\ A \circ T = T = T \circ B \\
 & \qquad :\ \forall(R,S :\ R = F.T \wedge S = F.T : \\
 & \qquad\qquad R = S \wedge F.A \circ R = R = R \circ B)) \\
\equiv & \qquad \{ \text{ predicate calculus } \} \\
 & \forall(T :\ A \circ T = T = T \circ B \\
 & \qquad :\ F.A \circ F.T = F.T = F.T \circ B) \\
\equiv & \qquad \{ \ F \text{ is a relator and so distributes over composition } \} \\
 & \textbf{true}
\end{array}$$

Secondly, property P2 is verified as follows:

$$\begin{array}{ll}
 & (F \circ (A \sim B))_{>} \\
= & \qquad \{ \text{ domains: (10.21) } \} \\
 & (F_{>} \circ (A \sim B))_{>} \\
\equiv & \qquad \{ \ F \text{ is total. I.e. } F_{>} = I \ \} \\
 & (A \sim B)_{>} \\
\equiv & \qquad \{ \ A \sim B \text{ is a monotype by definition, (10.27) } \} \\
 & A \sim B
\end{array}$$

## 11.2 The Naturality Operators

Saving one bound variable is hardly justification for such a spate of definitions. The motivation for presenting theorem 11.2 was to be able to compare it to theorem 11.5 below. First, yet three more definitions.

**Definition 11.3 (The Naturality Operators)** Let $R$ and $S$ be specs. Then we define the relations $R \overset{\cdot}{\leftsquigarrow} S$, $R \overset{\cdot}{\rightsquigarrow} S$ and $R \overset{\cdot}{\leftrightsquigarrow} S$ by

(a) $\quad U\,\langle R \overset{\cdot}{\leftsquigarrow} S\rangle\, V \;\equiv\; R \circ V \sqsupseteq U \circ S$

(b) $\quad U\,\langle R \overset{\cdot}{\rightsquigarrow} S\rangle\, V \;\equiv\; R \circ V \sqsubseteq U \circ S$

(c) $\quad U\,\langle R \overset{\cdot}{\leftrightsquigarrow} S\rangle\, V \;\equiv\; R \circ V = U \circ S$

□

The above definition of the $\overset{\cdot}{\leftsquigarrow}$ operator was introduced in [6]; it is related by part (a) of the following theorem to definitions introduced variously by deBruin [26], Reynolds [79] and Wadler [91].

**Theorem 11.4**

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

**Proof** We prove part (a) only. We begin by transforming the right side of the claimed equivalence to a dummy-free form.

$$
\begin{array}{ll}
 & \forall(u,v ::\; f.u\langle R\rangle g.v \Leftarrow u\langle S\rangle v) \\
\equiv & \qquad \{\ \text{one-point rule}\ \} \\
 & \forall(u,v ::\; \exists(s,t :\; s = f.u \,\wedge\, t = g.v :\; s\langle R\rangle t) \Leftarrow u\langle S\rangle v) \\
\equiv & \qquad \{\ \text{definition of higher order specs}\ \} \\
 & \forall(u,v ::\; \exists(s,t :\; u\langle f^{\cup}\rangle s \wedge t\langle g\rangle v :\; s\langle R\rangle t) \Leftarrow u\langle S\rangle v) \\
\equiv & \qquad \{\ \text{definition of composition at higher order}\ \} \\
 & \forall(u,v ::\; u\langle f^{\cup} \circ R \circ g\rangle v \Leftarrow u\langle S\rangle v) \\
\equiv & \qquad \{\ \text{definition of } \sqsupseteq\ \} \\
 & f^{\cup} \circ R \circ g \;\sqsupseteq\; S
\end{array}
$$

Now we show that this is equivalent to $f\,\langle R \overset{.}{\leftsquigarrow} S\rangle\, g$.

$$
\begin{array}{ll}
 & f^{\cup} \circ R \circ g \;\sqsupseteq\; S \\
\Rightarrow & \qquad \{\ \text{monotonicity}\ \} \\
 & f \circ f^{\cup} \circ R \circ g \;\sqsupseteq\; f \circ S \\
\Rightarrow & \qquad \{\ f \text{ is a function, Thus } f \circ f^{\cup} \;\sqsubseteq\; I\ \} \\
 & R \circ g \sqsupseteq f \circ S \\
\Rightarrow & \qquad \{\ \text{monotonicity}\ \} \\
 & f^{\cup} \circ R \circ g \;\sqsupseteq\; f^{\cup} \circ f \circ S \\
\Rightarrow & \qquad \{\ f \text{ is total, thus } f^{\cup} \circ f \;\sqsupseteq\; I\ \} \\
 & f^{\cup} \circ R \circ g \;\sqsupseteq\; S
\end{array}
$$

The theorem now follows from the definition of $R \overset{.}{\leftsquigarrow} S$.

□

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

## 11.3 Naturality of Relators, Reverse and Composition

The reader is invited to compare the following theorem with theorem 11.2.

**Theorem 11.5 (Naturality of Relators)** If $F$ is a relator then for all specs $R$ and $S$

(a) $\quad F \in (F.R \overset{.}{\leftsquigarrow} F.S) \overset{.}{\leftsquigarrow} (R \overset{.}{\leftsquigarrow} S)$

(b) $\quad F \in (F.R \overset{.}{\rightsquigarrow} F.S) \overset{.}{\leftsquigarrow} (R \overset{.}{\rightsquigarrow} S)$

(c) $\quad F \in (F.R \overset{.}{\leftrightsquigarrow} F.S) \overset{.}{\leftsquigarrow} (R \overset{.}{\leftrightsquigarrow} S)$

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

$$
\begin{array}{ll}
 & F \;\in\; (F.R \;\overset{.}{\leftsquigarrow}\; F.S) \;\overset{.}{\leftsquigarrow}\; (R \;\overset{.}{\leftsquigarrow}\; S) \\
\equiv & \qquad \{\ \text{definition of } \in\ \} \\
 & F\,\langle (F.R \;\overset{.}{\leftsquigarrow}\; F.S) \;\overset{.}{\leftsquigarrow}\; (R \;\overset{.}{\leftsquigarrow}\; S)\rangle\, F \\
\equiv & \qquad \{\ \text{theorem 11.4, relators are total functions}\ \} \\
 & \forall (U, V :: \;\; F.U\,\langle F.R \;\overset{.}{\leftsquigarrow}\; F.S\rangle\, F.V \;\Leftarrow\; U\,\langle R \;\overset{.}{\leftsquigarrow}\; S\rangle\, V) \\
\equiv & \qquad \{\ \text{definition 11.3(a)}\ \}
\end{array}
$$

$$
\begin{array}{ll}
 & \forall(U,V ::\ F.R \circ F.V \sqsupseteq F.U \circ F.S \quad \Leftarrow \quad R \circ V \sqsupseteq U \circ S) \\
\equiv & \qquad \{ \text{ relators distribute through composition and are monotonic } \} \\
 & \textbf{true}
\end{array}
$$

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 11.5 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, and about catamorphisms. 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 11.5 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 11.6 (Naturality of Composition)**
For all $\dot{\sim}\ \in \{\dot{\leftsquigarrow},\ \dot{\rightsquigarrow},\ \dot{\leftrightsquigarrow}\}$ and all specs $P_1$, $P_2$, $Q_1$, $Q_2$, $R$, $S$, $T$,

$$P_1 \circ Q_1 \langle R \dot{\sim} T\rangle\, P_2 \circ Q_2 \quad \Leftarrow \quad P_1 \langle R \dot{\sim} S\rangle\, P_2 \quad \wedge \quad Q_1 \langle S \dot{\sim} T\rangle\, Q_2$$

In particular,

$$P \circ Q \in R \dot{\sim} T \quad \Leftarrow \quad P \in R \dot{\sim} S \quad \wedge \quad Q \in S \dot{\sim} T$$

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

$$
\begin{array}{ll}
 & P_1 \langle R \dot{\sim} S\rangle\, P_2 \quad \wedge \quad Q_1 \langle S \dot{\sim} T\rangle\, Q_2 \\
\equiv & \qquad \{ \text{ definition of } \dot{\sim} \ \} \\
 & R \circ P_2 \trianglelefteq P_1 \circ S \quad \wedge \quad S \circ Q_2 \trianglelefteq Q_1 \circ T
\end{array}
$$

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

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 $\{\dot{\leftsquigarrow},\ \dot{\rightsquigarrow},\ \dot{\leftrightsquigarrow}\}$ and $\unlhd$ 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 $\dot{\leftsquigarrow}$ then $\unlhd$ is $\sqsupseteq$, if $\dot{\sim}$ is $\dot{\leftrightsquigarrow}$ then $\unlhd$ is $=$, and if $\dot{\sim}$ is $\dot{\rightsquigarrow}$ then $\unlhd$ is $\sqsubseteq$.) At other times they are used singly. In addition we sometimes use $\unrhd$ in the rôle of a variable (always in combination with $\unlhd$) in which case it designates the reverse of the relation designated by $\unlhd$. *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 11.7 (Naturality of Reverse)** For all specs $R$ and $S$,

(a) $\quad \cup \ \in \ (R^{\cup} \,\dot{\leftsquigarrow}\, S^{\cup}) \,\dot{\leftrightsquigarrow}\, (S \,\dot{\rightsquigarrow}\, R)^{\cup}$

(b) $\quad \cup \ \in \ (R^{\cup} \,\dot{\rightsquigarrow}\, S^{\cup}) \,\dot{\leftrightsquigarrow}\, (S \,\dot{\leftsquigarrow}\, R)^{\cup}$

(c) $\quad \cup \ \in \ (R^{\cup} \,\dot{\leftrightsquigarrow}\, S^{\cup}) \,\dot{\leftrightsquigarrow}\, (S \,\dot{\leftrightsquigarrow}\, R)^{\cup}$

**Proof** We prove (a) as an example.

$$
\begin{array}{ll}
& \cup \ \in \ (R^{\cup} \,\dot{\leftsquigarrow}\, S^{\cup}) \,\dot{\leftrightsquigarrow}\, (S \,\dot{\rightsquigarrow}\, R)^{\cup} \\
\equiv & \quad \{ \text{ theorem 11.4(c), reverse is an isomorphism } \} \\
& \forall(U,V :: \ \ U^{\cup} \, \langle R^{\cup} \,\dot{\leftsquigarrow}\, S^{\cup} \rangle \, V^{\cup} \ \ \equiv \ \ U \, \langle (S \,\dot{\rightsquigarrow}\, R)^{\cup} \rangle \, V) \\
\equiv & \quad \{ \text{ definition 11.3(a), reverse and definition 11.3(b) } \} \\
& \forall(U,V :: \ \ R^{\cup} \circ V^{\cup} \ \ \sqsupseteq \ \ U^{\cup} \circ S^{\cup} \ \ \equiv \ \ S \circ U \ \sqsubseteq \ V \circ R) \\
\equiv & \quad \{ \text{ reverse } \} \\
& \mathbf{true}
\end{array}
$$

□

## 11.4 Natural Simulations and Natural Isomorphisms

In this section we briefly introduce two concepts vital to the development of a theory of data refinement (or reification, as it is sometimes called), namely, natural simulations and natural isomorphisms. Elsewhere in the report we return to the topic, giving examples and establishing properties of simulations and isomorphisms.

**Definition 11.8 (Natural Formations)** We say that spec $\gamma$ is a *natural transformation* to relator $F$ from relator $G$ and write $\gamma \in F \leftrightsquigarrow G$ if and only if, for all specs $R$, $\gamma \in F.R \mathrel{\dot{\leftrightsquigarrow}} G.R$.

We say that spec $\gamma$ is a natural *down-formation* to relator $F$ from relator $G$ and write $\gamma \in F \leftsquigarrow G$ if and only if, for all specs $R$, $\gamma \in F.R \mathrel{\dot{\leftsquigarrow}} G.R$.

Finally, we say that spec $\gamma$ is a natural *up-formation* to relator $F$ from relator $G$ and write $\gamma \in F \rightsquigarrow G$ if and only if, for all specs $R$, $\gamma \in F.R \mathrel{\dot{\leftsquigarrow}} G.R$.

□

Expanding the definition of $\leftrightsquigarrow$ one obtains the identity:

$$(11.9) \quad \gamma \in F \leftrightsquigarrow G \quad \equiv \quad \forall(R :: \ F.R \circ \gamma \ = \ \gamma \circ G.R)$$

It is this form of the definition that we use most frequently. The corresponding identities for down- and up-formations are also frequently used.

*Warning* In the following discussion we will be particulary concerned with the composition of relators. To facilitate the calculations we assume that function application associates to the right. For example, $F.G.X$ should be read as $F.(G.X)$. We denote the composition of relators $F$ and $G$ by $F \bullet G$. This choice of notation is akin to that in category theory (where both application of functors to their arguments and composition of functors are denoted by juxtaposition) but completely opposite to the convention in the lambda calculus. Suffice it to say that we have no use of curried functions as they are used in the lambda calculus, and, hence, there is no advantage to us of letting function application associate to the left. *End of Warning*

**Definition 11.10 (Natural Simulation)** Relator $F$ is said to (naturally) *simulate* relator $G$ if and only if there exists a spec $\gamma$ such that

(a) $\gamma \in F {<\!\!\sim\!\!>} G$
(b) $F.I \;\sqsupseteq\; \gamma \circ \gamma^{\cup}$
(c) $\gamma^{\cup} \circ \gamma \;=\; G.I$

The spec $\gamma$ itself is called the *witness* to the simulation.
□

We denote the fact that $\gamma$ witnesses a simulation of relator $G$ by relator $F$ by $\gamma \in F \gtrsim G$. When the existence of a witness is known but not directly relevant we write $F \gtrsim G$.

More insight is gained into the definition of a simulation by considering the following equivalent definition.

**Theorem 11.11** For relators $F$ and $G$ and spec $\gamma$, $\gamma \in F \gtrsim G$ equivales the conjunction of the three conditions:

(a) $\gamma \in F {<\!\!\sim\!\!>} G$
(b) $\gamma$ is a bijection
(c) $F.I \;\sqsupseteq\; \gamma{<}$ and $G.I \;=\; \gamma{>}$
□

The proof is very routine and so has been omitted. (Two implications have to be proven. In one direction the main observation is that both $F.I$ and $G.I$ are monotypes. In the other direction (10.18) must be applied.)

**Definition 11.12 (Natural Isomorphism)** Relators $F$ and $G$ are said to be (naturally) *isomorphic* if and only if there exists a spec $\gamma$ such that

(a) $\gamma \in F {<\!\!\sim\!\!>} G$
(b) $\gamma \circ \gamma^{\cup} \;=\; F.I$
(c) $\gamma^{\cup} \circ \gamma \;=\; G.I$

The spec $\gamma$ itself is called the *witness* to the isomorphism.
□

Again there is an equivalent definition that adds extra insight.

**Theorem 11.13** For relators $F$ and $G$ and spec $\gamma$, $\gamma \in F \cong G$ equivales the conjunction of the three conditions:

(a) $\gamma \in F \leftrightsquigarrow G$
(b) $\gamma$ is a bijection
(c) $F.I \;=\; \gamma{<}$ and $G.I \;=\; \gamma{>}$
□

Note that the conjunction of (b) and (c) is just the statement that $\gamma$ is a bijection to $F.I$ from $G.I$.

We denote the fact that $\gamma$ witnesses an isomorphism between relators $F$ and $G$ by $\gamma \in F \cong G$. (Note that the order of $F$ and $G$ is relevant.) When the existence of a witness is known but not directly relevant we write $F \cong G$. (In this case the order is not irrelevant.)
*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 $\leftrightsquigarrow$ by $\mathrel{<\!\sim}$. 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*

The first examples of natural simulations and natural isomorphisms appear in section 12.5. In this section we limit ourselves to a few abstract properties of natural isomorphisms.

Suppose $F$ is a relator and $\gamma$ is a bijection with $\gamma{<} \;=\; F.I$. Define $F^{\gamma}$ by

(11.14) $F^{\gamma}.R \;=\; \gamma^{\cup} \circ F.R \circ \gamma$

Then we have:

**Theorem 11.15**

(a) $\gamma{>} \;=\; F^{\gamma}.I$
(b) $F^{\gamma}$ is a relator.
(c) $F^{F.I} \;=\; F$
(d) For all relators $G$,
$$\gamma \in F \cong G \;\equiv\; G \;=\; F^{\gamma}$$
and $$G \;=\; F^{\gamma} \;\equiv\; G^{\gamma^{\cup}} \;=\; F$$
(e) If $\delta$ is a bijection such that $\delta{<} \;=\; \gamma{>}$ then
$$(F^{\gamma})^{\delta} \;=\; F^{\gamma \circ \delta}$$
(f) For all relators $G$,

$$(G \bullet F)^{G.\gamma} \;=\; G \bullet F^{\gamma}$$
and
$$(F \bullet G)^{(F.G.I \;\circ\; \gamma)} \;=\; F^{\gamma} \bullet G$$

□

**Proof** Properties (a) and (b) are trivial. (They have been included because they are a necessary preliminary to the remaining parts of the theorem.) So is (c). The first part of (d) consists, in fact, of two implications that have to be proved independently. The first is $\gamma \in F \cong F^{\gamma}$ which is quite simple to prove:

$$
\begin{array}{ll}
 & \gamma \in F \cong F^{\gamma} \\
\equiv & \quad \{ \text{ definition } \} \\
 & \mathit{bijection}.\gamma \;\wedge\; \gamma< = F.I \;\wedge\; \gamma> = F^{\gamma}.I \\
\equiv & \quad \{ \text{ assumption } \} \\
 & \gamma> = F^{\gamma}.I \\
\equiv & \quad \{ \text{ (a) } \} \\
 & \mathbf{true}
\end{array}
$$

The second is $\gamma \in F \cong G \;\Rightarrow\; G = F^{\gamma}$. This is also quite simple to prove. We have, for all specs $X$,

$$
\begin{array}{ll}
 & G.X = F^{\gamma}.X \\
\equiv & \quad \{ \text{ definition } \} \\
 & G.X = \gamma\cup \circ F.X \circ \gamma \\
\Leftarrow & \quad \{ \text{ assumption: } \gamma\cup \circ \gamma = G.I \;\} \\
 & \gamma \circ G.X = F.X \circ \gamma \\
\equiv & \quad \{ \text{ assumption: } \gamma \in F \leftrightsquigarrow G \;\} \\
 & \mathbf{true}
\end{array}
$$

A similar proof is needed for the second part of (d).

For (e) one must first check that $G.\gamma< = G.F.I$. This is immediate from theorem 10.34 and the assumption that $\gamma< = F.I$. One must also check that $G.\gamma$ is a bijection, which it is by theorem 10.35(b) and (c). The remainder of the proof — checking that, for all specs $X$,

$$(G \bullet F)^{G.\gamma}.X \;=\; G.F^{\gamma}.X$$

— is a straightforward application of the fact that relators distribute through composition and commute with reverse.

Property (f) involves a similar set of proof obligations, and is just as straightforward.

□

**Corollary 11.16** "Naturally isomorphic" is an equivalence relation on relators. Moreover, for all relators $F$, $G$ and $H$,

$$F \cong G \quad \Rightarrow \quad F \bullet H \cong G \bullet H \ \wedge \ H \bullet F \cong H \bullet G$$

□

We leave this proof as an instructive exercise to the reader. As a hint we would remark that the properties of reflexivity, symmetry and transitivity correspond to theorem 11.15(c), (d) and (e), respectively. Preservation under composition is captured by (f).

To conclude this section we summarise several properties of the naturality operators in four inference rules. Note that (b) and (c) are instances of (d).

**Theorem 11.17 (Poly Rules)** For $\dot{\sim} \ \in \ \{{<\!\!\sim}, {<\!\!\sim\!\!>}, \overset{>}{\sim}, \cong\}$

(a) $\gamma \in F \mathrel{\dot{\sim}} G \ \wedge \ \delta \in G \mathrel{\dot{\sim}} H \quad \Rightarrow \quad \gamma \circ \delta \in F \mathrel{\dot{\sim}} H$
(b) $\gamma \in F \mathrel{\dot{\sim}} G \quad \Rightarrow \quad H.\gamma \in H \bullet F \mathrel{\dot{\sim}} H \bullet G$
(c) $\gamma \in F \mathrel{\dot{\sim}} G \quad \Rightarrow \quad \gamma \circ G.H.I \in F \bullet H \mathrel{\dot{\sim}} G \bullet H$
(d) $\gamma \in F \mathrel{\dot{\sim}} G \ \wedge \ \delta \in H \mathrel{\dot{\sim}} K \ \Rightarrow \ \gamma \circ G.\delta \in F \bullet H \mathrel{\dot{\sim}} G \bullet K$
□

# Chapter 12

# Polynomial Data Types and Relators

In order that our theory be of any use we need to ensure that we can indeed define some non-trivial monotypes and relators. There are just three components needed to build a significant theory of datatypes, namely a unit type, a disjoint sum operator and a cartesian product operator. In this section we present an axiomatisation of these three components.

## 12.1 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.

### 12.1.1 The Cone Rule

We begin by postulating an axiom dubbed "the cone rule". This axiom could equally well have been included in section 9.1. 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. [83]) 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 middle exchange rule.

**Lemma 12.1** For all specs $R$, the following statements are all equivalent:

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

**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). That (d) implies (a) follows by the following simple argument:

$$\begin{array}{ll}
& \bot\!\!\bot \;=\; \top\!\!\top \circ R \circ \top\!\!\top \\
\equiv & \quad \{ \text{ calculus } \} \\
& \bot\!\!\bot \;\sqsupseteq\; \top\!\!\top \circ R \circ \top\!\!\top \\
\Rightarrow & \quad \{ \text{ plat calculus } \} \\
& \bot\!\!\bot \;\sqsupseteq\; I \circ R \circ I \\
\equiv & \quad \{ \text{ calculus } \} \\
& \bot\!\!\bot \;=\; R
\end{array}$$

Finally, (e) is equivalent to (b), and (f) to (c), on account of (10.15) and (10.20).
□

One consequence of the cone rule is that $\top\!\!\top$ and $\bot\!\!\bot$ are different. More significantly, by combining the cone rule and lemma 12.1, 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 12.2** $\perp\!\!\perp \;=\; R \circ \top\!\!\top \circ S \quad\equiv\quad \perp\!\!\perp \;=\; R \;\vee\; \perp\!\!\perp \;=\; S$

**Proof** Follows from is clearly trivial. Implication admits a simple and elegant proof.

$$
\begin{array}{ll}
 & \perp\!\!\perp = R \circ \top\!\!\top \circ S \\
\equiv & \qquad \{ \text{ cone rule } \} \\
 & \top\!\!\top \neq \top\!\!\top \circ R \circ \top\!\!\top \circ S \circ \top\!\!\top \\
\equiv & \qquad \{ \ \top\!\!\top \;=\; \top\!\!\top \circ \top\!\!\top \ \} \\
 & \top\!\!\top \neq \top\!\!\top \circ R \circ \top\!\!\top \circ \top\!\!\top \circ S \circ \top\!\!\top \\
\Rightarrow & \qquad \{ \ \top\!\!\top \;=\; \top\!\!\top \circ \top\!\!\top \ \} \\
 & \top\!\!\top \neq \top\!\!\top \circ R \circ \top\!\!\top \;\vee\; \top\!\!\top \neq \top\!\!\top \circ S \circ \top\!\!\top \\
\equiv & \qquad \{ \text{ cone rule } \} \\
 & \perp\!\!\perp = R \vee \perp\!\!\perp = S
\end{array}
$$

□

### 12.1.2 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 $\mathbb{1}$, such that

$$(12.3)\quad \perp\!\!\perp \;\neq\; \mathbb{1}$$

and

$$(12.4)\quad I \;\sqsupseteq\; \mathbb{1} \circ \top\!\!\top \circ \mathbb{1}$$

There are several ways to convince oneself that axioms (12.3) and (12.4) 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.

### 12.1.3 An Atomic Monotype

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

**Theorem 12.5** $\mathbb{1}$ is a monotype.

**Proof**

$$
\begin{array}{ll}
 & I \\
\sqsupseteq & \quad \{\ (12.4)\ \} \\
 & \mathbb{1} \circ \top\!\!\top \circ \mathbb{1} \\
\sqsupseteq & \quad \{\ \text{domains: (10.18), monotonicity}\ \} \\
 & \mathbb{1}< \circ \mathbb{1} \\
= & \quad \{\ \text{domains: (10.25)}\ \} \\
 & \mathbb{1}
\end{array}
$$

□

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

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

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 12.6** $\mathbb{1}$ is an atom.

**Proof** Let $X$ be a spec such that $\mathbb{1} \sqsupseteq X$. Then, by the definition of an atom, we must prove that $\perp\!\!\perp = X \ \vee \ \mathbb{1} = X$. Assume $X \neq \perp\!\!\perp$. Then, by the cone rule, $\top\!\!\top = \top\!\!\top \circ X \circ \top\!\!\top$ . Aiming at the use of this property we calculate as follows:

$$
\begin{array}{ll}
 & \mathbb{1} \sqsubseteq X \\
\equiv & \quad \{\ (\circ\top\!\!\top) \text{ is an order isomorphism on monotypes }\ \} \\
 & \mathbb{1} \circ \top\!\!\top \sqsubseteq X \circ \top\!\!\top \\
\Leftarrow & \quad \{\ (12.4)\ \} \\
 & \mathbb{1} \circ \top\!\!\top \sqsubseteq \mathbb{1} \circ \top\!\!\top \circ \mathbb{1} \circ X \circ \top\!\!\top \\
\Leftarrow & \quad \{\ \text{monotonicity}\ \} \\
 & \top\!\!\top \sqsubseteq \top\!\!\top \circ \mathbb{1} \circ X \circ \top\!\!\top \\
\Leftarrow & \quad \{\ \top\!\!\top = \top\!\!\top \circ X \circ \top\!\!\top \text{ , monotonicity }\ \} \\
 & \mathbb{1} \circ X = X \\
\Leftarrow & \quad \{\ \text{monotypes: 10.2}\ \} \\
 & X \sqsubseteq \mathbb{1}
\end{array}
$$

□

Properties (12.3), (12.5) and (12.6) express, respectively, that $\mathbb{1}$ is non-empty, and is a monotype corresponding to a set containing at most one element.

### 12.1.4 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 $\mathbb{1} \longleftarrow A$. Introducing the definition

(12.7) $\; ! \;\; = \;\; \mathbb{1} \circ \top\!\!\top$

this characterisation of the unit type is mimicked in our theory by the following two consequences of axioms (12.3) and (12.4). For all monotypes $A$,

(12.8) $\; ! \circ A \;\in\; \mathbb{1} \longleftarrow A$

(12.9) $\; R \in \mathbb{1} \sim A \quad \wedge \quad R{>} \;=\; A \quad \equiv \quad R \;=\; ! \;\circ\; A$

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

Equivalent, more succinct, and more fundamental, renderings of (12.8) and (12.9) are

(12.10) $\; ! \quad$ is an imp, $\quad$ and

(12.11) $\; \mathbb{1} \quad = \quad \mathbb{1} \;\circ\; \top\!\!\top \;\circ\; \mathbb{1}$

from which follows

(12.12) $\; \mathbb{1} \;\sqsupseteq\; R{<} \quad \equiv \quad R \;=\; ! \;\circ\; R{>}$

(Note: the equivalence of (12.8) and (12.9) to (12.10) and (12.12) involves a non-trivial proof but is nonetheless left to the reader.) Here are their proofs.

**Proof of (12.10)**

$$
\begin{array}{ll}
 & ! \;\;\text{ is an imp} \\
\equiv & \quad \{ \text{ definition, reverse } \} \\
 & I \;\sqsupseteq\; \mathbb{1} \;\circ\; \top\!\!\top \;\circ\; \top\!\!\top{\cup} \;\circ\; \mathbb{1}{\cup} \\
\equiv & \quad \{ \; \mathbb{1} \text{ is a monotype, (10.1), properties of } \top\!\!\top \; \}
\end{array}
$$

$$
\begin{array}{ll}
 & I \;\sqsupseteq\; \mathbb{1} \circ \top\!\!\top \circ \mathbb{1} \\
\equiv & \quad \{\ 12.4\ \} \\
 & \textbf{true}
\end{array}
$$

$\square$

**Proof of (12.11)**

$$
\begin{array}{ll}
 & \mathbb{1} \circ \top\!\!\top \circ \mathbb{1} \\
\sqsupseteq & \quad \{\ \top\!\!\top \sqsupseteq I\ \} \\
 & \mathbb{1} \circ \mathbb{1} \\
= & \quad \{\ \mathbb{1} \text{ is a monotype, (10.1)}\ \} \\
 & \mathbb{1} \\
\sqsupseteq & \quad \{\ \text{(12.4), monotonicity of composition}\ \} \\
 & \mathbb{1} \circ \mathbb{1} \circ \top\!\!\top \circ \mathbb{1} \\
= & \quad \{\ \mathbb{1} \text{ is a monotype, (10.1)}\ \} \\
 & \mathbb{1} \circ \top\!\!\top \circ \mathbb{1}
\end{array}
$$

$\square$

**Proof of (12.12)**

$$
\begin{array}{ll}
 & \mathbb{1} \;\sqsupseteq\; R{<} \\
\equiv & \quad \{\ (10.12)\ \} \\
 & R \;=\; \mathbb{1} \circ R \\
\equiv & \quad \{\ (12.11)\ \} \\
 & R = \mathbb{1} \circ \top\!\!\top \circ \mathbb{1} \circ R \;\;\wedge\;\; R = \mathbb{1} \circ R \\
\equiv & \quad \{\ \text{(10.1), } \mathbb{1} \text{ is a monotype}\ \} \\
 & R \;=\; \mathbb{1} \circ \top\!\!\top \circ R \\
\equiv & \quad \{\ \text{(10.20) with } R,S \;:=\; S,\top\!\!\top\ \} \\
 & R \;=\; \mathbb{1} \circ \top\!\!\top \circ R{>}
\end{array}
$$

$\square$

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

### 12.1.5 A Summary of Basic Properties

The "foundations" that were laid in sections 10 and 11 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 $\mathbb{1}$ is a relator (because $\mathbb{1}$ 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 12.13**

(a) $\quad !{<} \;=\; \mathbb{1}$

(b) $\quad !{>} \;=\; I$

□

Verification of both of these is straightforward and is left to the reader. (For part (a) make use of (12.11). For part (b) make use of the cone rule.)
The final question in this list is answered by the following theorem.

**Theorem 12.14**

$$! \in \mathbb{1} \mathrel{\dot{\leftrightsquigarrow}} \top\!\!\top$$

In particular, for all specs $R$,

$$! \in \mathbb{1} \mathrel{\dot{\leftsquigarrow}} R$$

**Proof**

$$
\begin{array}{ll}
 & ! \in \mathbb{1} \mathrel{\dot{\leftrightsquigarrow}} \top\!\!\top \\
\equiv & \qquad \{ \text{ definition !, definition } \in \ \} \\
 & \mathbb{1} \circ \top\!\!\top \;\langle \mathbb{1} \mathrel{\dot{\leftrightsquigarrow}} \top\!\!\top \rangle\; \mathbb{1} \circ \top\!\!\top \\
\equiv & \qquad \{ \text{ definition } \dot{\leftrightsquigarrow} \ \} \\
 & \mathbb{1} \circ \mathbb{1} \circ \top\!\!\top \;=\; \mathbb{1} \circ \top\!\!\top \circ \top\!\!\top \\
\equiv & \qquad \{ \ \mathbb{1} \text{ is a monotype, } \top\!\!\top = \top\!\!\top \circ \top\!\!\top \ \} \\
 & \textbf{true}
\end{array}
$$

The corollary follows because equality is a special case of inclusion and $\dot{\leftsquigarrow}$ 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

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

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 "$\dot{\leftsquigarrow}$". 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.

## 12.2 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:

$$(12.15)\ P \vartriangle Q \;=\; (\ll^{\cup} \circ P) \sqcap (\gg^{\cup} \circ Q)$$
$$(12.16)\ P \triangledown Q \;=\; (P \circ \hookrightarrow^{\cup}) \sqcup (Q \circ \hookleftarrow^{\cup})$$
$$(12.17)\ P \times Q \;=\; (P \circ \ll) \vartriangle (Q \circ \gg)$$
$$(12.18)\ P{+}Q \;=\; (\hookrightarrow \circ P) \triangledown (\hookleftarrow \circ Q)$$

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,

$$x \langle\ll\rangle (x, y)$$
$$y \langle\gg\rangle (x, y)$$

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

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

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

Our first axiom is that the injections are both imps.

$$(12.19)\ I \;\sqsupseteq\; (\hookrightarrow \circ \hookrightarrow^{\cup}) \sqcup (\hookleftarrow \circ \hookleftarrow^{\cup})$$

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

$$(12.20)\ I \;\sqsupseteq\; (\ll^{\cup} \circ \ll) \sqcap (\gg^{\cup} \circ \gg)$$

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

(Berghammer and Zierer [14] and de Roever [81] introduce an almost identical axiom to (12.20) 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 (12.19) and (12.20) take the following form when rephrased in terms of the product and sum operations.

$$(12.21)\quad I \quad \sqsupseteq \quad I{+}I$$
$$(12.22)\quad I \quad \sqsupseteq \quad I \times I$$

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 $\times$)

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

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

$$(12.23)\quad (P \vartriangle Q)^{\cup} \circ (R \vartriangle S) \quad = \quad (P^{\cup} \circ R) \sqcap (Q^{\cup} \circ S)$$

The dual axiom for disjoint sum is:

$$(12.24)\quad (P \triangledown Q) \circ (R \triangledown S)^{\cup} \quad = \quad (P \circ R^{\cup}) \sqcup (Q \circ S^{\cup})$$

(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:

$$(12.25)\quad \ll> \quad = \quad \gg>$$

Property (12.25) is equivalent to

$$(12.26)\quad \top\!\top \circ \ll \quad = \quad \top\!\top \circ \gg$$

Its dual is therefore the trivially true

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

There are thus no further axioms for disjoint sum.

The five properties (12.19), (12.20), (12.23), (12.24) and (12.25) 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 $\sqcap$-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.

## 12.3 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. the 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 manifold. 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

12.3.1, the computation rules in subsection 12.3.2 and the terminality property in subsection 12.3.6 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.

### 12.3.1 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 (12.22)). 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.

(12.27) $(P \times Q)\cup \;=\; P\cup \times Q\cup$

It remains to show that composition distributes over cartesian product:

**Theorem 12.28 (Product-Split and Product-Product Fusion)**

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

**Proof** We only prove the (a)-part, the other part follows immediately from (a) and (12.17).

$$
\begin{array}{ll}
 & (P \times Q) \circ (R \vartriangle S) \\
= & \quad \{ \text{ (12.27), reverse } \} \\
 & (P\cup \times Q\cup)\cup \circ (R \vartriangle S) \\
= & \quad \{ \text{ (12.17) } \} \\
 & ((P\cup \circ \ll) \vartriangle (Q\cup \circ \gg))\cup \circ (R \vartriangle S) \\
= & \quad \{ \text{ (12.23) } \} \\
 & (P\cup \circ \ll)\cup \circ R \;\sqcap\; (Q\cup \circ \gg)\cup \circ S \\
= & \quad \{ \text{ reverse } \} \\
 & \ll\cup \circ P \circ R \;\sqcap\; \gg\cup \circ Q \circ S \\
= & \quad \{ \text{ (12.15) } \} \\
 & (P \circ R) \vartriangle (Q \circ S)
\end{array}
$$

□

Properties (12.28a) and (12.28b) 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 “catamorphism” 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 catamorphism. (Typically, as in (12.28a) and (12.28b) one of the specs to be fused is itself a catamorphism.) Later on, when we discuss relators defined via fixed-points we shall observe a connection between catamorphism fusion and loop fusion, and such properties will prove their worth in enabling us to derive efficient programs. 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 12.28(b), for instance, is both a fusion property — because a product is a particular form of catamorphism — and an abide law — composition and product abide with each other. *End of Remark*

**Corollary 12.29** $\times$ 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 12.30 (Split-Spec and Split-Imp Fusion)**

(a) $(R \vartriangle S) \circ T \ \sqsubseteq\ (R \circ T) \ \vartriangle\ (S \circ T)$
(b) $(R \vartriangle S) \circ T \ =\ (R \circ T) \ \vartriangle\ (S \circ T)$
$\quad \Leftarrow \quad R \sqsupseteq R \circ T \circ T\cup \ \vee\ S \sqsupseteq S \circ T \circ T\cup$

In particular, for all imps $f$,

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

Also

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

Finally,

$$\begin{array}{lll}
\text{(e)} & (R \vartriangle S) \circ T \;=\; (R \circ T) \vartriangle (S \circ T) & \\
& \quad \Leftarrow \quad (R \vartriangle S)_< \sqsupseteq ((R \circ T) \vartriangle (S \circ T))_< & \\
& \qquad \wedge \;\; T \sqsupseteq R^\cup \circ R \circ T \;\wedge\; T \sqsupseteq S^\cup \circ S \circ T &
\end{array}$$

**Proof** As indicated (c) is an easy consequence of (b). (Just check that the premisses in (b) are fulfilled by imps $f$. Straightforward unfolding of the definition of split augmented by, in the case of part (a), monotonicity, in the case of part (b), lemma D14, in the case of part (d), theorem D15 and in the case of part (e), theorem D20 suffices to establish the remaining parts.
□

### 12.3.2 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 12.31**

$$\begin{array}{lll}
\text{(a)} & \top\!\!\top \circ \gg \;\sqsupseteq\; \ll \\
\text{(b)} & \top\!\!\top \circ \ll \;\sqsupseteq\; \gg
\end{array}$$

**Proof** Immediate from (12.26), $\top\!\!\top \circ \ll \,\sqsupseteq\, \ll$ and $\top\!\!\top \circ \gg \,\sqsupseteq\, \gg$.
□

We shall have further use of lemma 12.31 later, but an immediate corollary is that one can express various combinations of one projection in terms of the split and product operators:

**Lemma 12.32**

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

In particular,

$$\begin{array}{lll}
\text{(e)} & \ll^\cup \;=\; I \vartriangle \top\!\!\top \\
\text{(f)} & \gg^\cup \;=\; \top\!\!\top \vartriangle I \\
\text{(g)} & \ll^\cup \circ \ll \;=\; I \times \top\!\!\top \\
\text{(h)} & \gg^\cup \circ \gg \;=\; \top\!\!\top \times I
\end{array}$$

**Proof** We only prove (a).

$$
\begin{array}{ll}
 & R \vartriangle \top\!\!\top \\
= & \quad \{\ (12.15)\ \} \\
 & \ll\cup \circ R \ \sqcap\ \gg\cup \circ \top\!\!\top \\
= & \quad \{\ \text{lemma 12.31(a), reverse, } \top\!\!\top \circ R \sqsubseteq \top\!\!\top\ \} \\
 & \ll\cup \circ R
\end{array}
$$

To prove (c) use exactly the same strategy.
□

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 12.33 (Computation Rules for Split)**

(a) $\ll \circ (P \vartriangle Q) \;=\; P \circ Q>$
(b) $\gg \circ (P \vartriangle Q) \;=\; Q \circ P>$

**Proof**

$$
\begin{array}{ll}
 & \ll \circ (P \vartriangle Q) \\
= & \quad \{\ (12.32)(e),\ \text{reverse}\ \} \\
 & (I \vartriangle \top\!\!\top)\cup \circ (P \vartriangle Q) \\
= & \quad \{\ \text{axiom: } (12.23)\ \} \\
 & P \ \sqcap\ \top\!\!\top \circ Q \\
= & \quad \{\ \text{domains: } (10.20)\ \} \\
 & P \circ Q>
\end{array}
$$

□

We mention one, easily derived, corollary of lemma 12.32 and theorem 12.33.

**Theorem 12.34**

$$\ll \circ \gg\cup \;=\; \top\!\!\top \;=\; \gg \circ \ll\cup$$

□

Theorem 12.34 is important if only because it is an important stepping stone to proving that product is "strict". (See section 12.3.5)

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 12.35**

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

**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.

$$\begin{array}{ll} & (P\circ\ll)> \\ = & \quad\{\ (10.21)\ \} \\ & (P>\ \circ\ \ll)> \\ = & \quad\{\ \text{definition of right domain, } P> \text{ is a monotype}\ \} \\ & I\ \ \sqcap\ \ \ll\cup\circ P>\circ\ll \\ = & \quad\{\ P> \text{ is a monotype, monotonicity}\ \} \\ & I\ \ \sqcap\ \ \ll\cup\circ P>\circ\ll\ \ \sqcap\ \ \ll\cup\circ\ll \\ = & \quad\{\ I\sqcap\ll\cup\circ\ll\ \ =\ \ \{\ (12.25)\ \}\ \ I\sqcap\gg\cup\circ\gg\ \ \} \\ & I\ \ \sqcap\ \ \ll\cup\circ P>\circ\ll\ \ \sqcap\ \ \gg\cup\circ\gg \\ = & \quad\{\ (12.20)\text{, monotonicity}\ \} \\ & \ll\cup\circ P>\circ\ll\ \ \sqcap\ \ \gg\cup\circ\gg \end{array}$$

□

**Theorem 12.36 (Computation Rules for Product)**

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

**Proof**

$$\begin{array}{ll} & \ll\ \circ\ (P\times Q) \\ = & \quad\{\ \text{definition of } \times\text{, computation rule 12.33(a)}\ \} \\ & P\ \circ\ \ll\ \circ\ (Q\circ\gg)> \end{array}$$

$$
\begin{array}{ll}
= & \quad \{ \ (12.35) \ \} \\
 & P \circ \ll \circ (I \times Q{>})
\end{array}
$$

□

The occurrence of the right domains in the right sides of theorems 12.33 and 12.36 alerts one to an important observation about product: the operands interact with each other in a curious and sometimes troublesome way. Rules that permit one to cancel one of the operands of a product or split — inevitably with provisos — are therefore useful. One such is the following.

**Theorem 12.37**

$$
\begin{array}{llllll}
\text{(a)} & R & = & \ll \circ R \times S \circ \ll^{\cup} & \Leftarrow & S \neq \bot\!\!\bot \\
\text{(b)} & S & = & \gg \circ R \times S \circ \gg^{\cup} & \Leftarrow & R \neq \bot\!\!\bot
\end{array}
$$

**Proof** We conduct the proof in two stages. First we establish

$$\ll \circ R \times S \circ \ll^{\cup} \quad = \quad R \sqcap \top\!\!\top \circ S \circ \top\!\!\top$$

$$
\begin{array}{ll}
 & \ll \circ R \times S \circ \ll^{\cup} \\
= & \quad \{ \ 12.32(e) \ \} \\
 & \ll \circ R \times S \circ I \vartriangle \top\!\!\top \\
= & \quad \{ \ \text{product-split fusion: } 12.28(a) \ \} \\
 & \ll \circ R \vartriangle (S \circ \top\!\!\top) \\
= & \quad \{ \ \text{computation rule } 12.33(a) \ \} \\
 & R \circ (S \circ \top\!\!\top){>} \\
= & \quad \{ \ \text{domains: } (10.20) \ \} \\
 & R \sqcap \top\!\!\top \circ S \circ \top\!\!\top
\end{array}
$$

Now, it is easy to apply the cone rule and obtain the required result:

$$
\begin{array}{ll}
 & R \\
= & \quad \{ \ \text{calculus} \ \} \\
 & R \sqcap \top\!\!\top \\
= & \quad \{ \ S \neq \bot\!\!\bot, \text{ cone rule} \ \} \\
 & R \sqcap \top\!\!\top \circ S \circ \top\!\!\top \\
= & \quad \{ \ \text{above} \ \} \\
 & \ll \circ R \times S \circ \ll^{\cup}
\end{array}
$$

□

### 12.3.3 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 12.38**

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

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

$$\begin{array}{ll} & \ll \circ \ll^{\cup} \\ = & \qquad \{\ (12.32)\ \} \\ & (I \vartriangle \top\!\!\top)^{\cup} \circ (I \vartriangle \top\!\!\top) \\ = & \qquad \{\ \text{axiom } (12.23\ \} \\ & I \sqcap (\top\!\!\top^{\cup} \circ \top\!\!\top) \\ = & \qquad \{\ \text{calculus}\ \} \\ & I \end{array}$$

□

**Corollary 12.39** $\ll$ and $\gg$ are both imps.

**Proof** Immediate from the definition of an imp.
□

Now we turn to product and split. Since product is a binary relator we have:

**Theorem 12.40** $\times$ preserves both imps and co-imps.
□

For split the situation is a little more interesting.

**Theorem 12.41**

(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 (12.23). Part (b) is a little less straightforward:

$$
\begin{array}{ll}
 & R \vartriangle S \ \text{ is an imp} \\
\equiv & \qquad \{ \ \text{definition} \ \} \\
 & (R \vartriangle S) \circ (R \vartriangle S)^{\cup} \ \sqsubseteq \ I \\
\equiv & \qquad \{ \ (12.15), \text{reverse} \ \} \\
 & (\ll^{\cup} \circ R \sqcap \gg^{\cup} \circ S) \circ (R^{\cup} \circ \ll \sqcap S^{\cup} \circ \gg) \ \sqsubseteq \ I \\
\Leftarrow & \qquad \{ \ \text{monotonicity} \ \} \\
 & \ll^{\cup} \circ R \circ R^{\cup} \circ \ll \ \sqcap \ \gg^{\cup} \circ S \circ S^{\cup} \circ \gg \ \sqsubseteq \ I \\
\Leftarrow & \qquad \{ \ (12.20) \ \} \\
 & R \circ R^{\cup} \ \sqsubseteq \ I \ \wedge \ S \circ S^{\cup} \ \sqsubseteq \ I \\
\equiv & \qquad \{ \ \text{definition} \ \} \\
 & R \text{ is an imp } \wedge \ S \text{ is an imp}
\end{array}
$$

□

**Corollary 12.42**

(a) $I \vartriangle I$ is a bijection
(b) $I \vartriangle R$ is a coimp for all specs $R$.

□

Specs of the form $I \vartriangle R$ form primitive instances of what Meertens [70] calls "paramorphisms". In particular, the doubling function $I \vartriangle I$ is important for various reasons. (One reason not elaborated further here is that in category theory it is one of the units in the defining adjunction of cartesian product. The other unit is the pair $(\ll, \gg)$. In the current relational setting product is not categorical but does fulfill a weaker notion of adjunction in which the two units are $I \vartriangle I$ and the pair $(\ll, \gg)$. Since we have observed that the two projections are imps it would be remiss of us not to at least mention that $I \vartriangle I$ is a co-imp.)

(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.)

### 12.3.4 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 12.38, for instance, tells us immediately that the projections are surjective. I.e.

**Theorem 12.43** $\ll\!< \;=\; I$ and $\gg\!< \;=\; I$
□

Moreover, lemma 12.35, with $P$ and $Q$ both instantiated to $I$, predicts their right domains:

**Theorem 12.44** $\ll\!> \;=\; I\times I$ and $\gg\!> \;=\; I\times I$
□

Since product is a (binary) relator we can immediately instantiate theorem 10.34 obtaining:

**Theorem 12.45**

$$\begin{array}{lrcl} (a) & (P\times Q)< & = & P<\times Q< \\ (b) & (P\times Q)> & = & P>\times Q> \end{array}$$
□

As discussed earlier it is important to establish rules that permit one to ignore one of the operands of a split or product. For the calculation of left domains we have the following such rule:

**Theorem 12.46**

$$\begin{array}{ll} (a) & (\ll\circ R\times S)< \,=\, R< \quad\Leftarrow\quad S\neq\perp\!\!\!\perp \\ (b) & (\gg\circ R\times S)< \,=\, S< \quad\Leftarrow\quad R\neq\perp\!\!\!\perp \end{array}$$

There is clearly a similar rule for right domains obtained by applying reverse to the arguments of the left-domain operators.

**Proof** Consider (a). Beginning with $(\ll\circ R\times S)<$ the goal in the calculation is to work towards an application of theorem 12.37. Since the term "$\ll\cup$" appears in the latter a way must be found to introduce it. Appropriate to this is theorem 12.44.

$$
\begin{array}{ll}
 & (\ll \circ R \times S)< \\
= & \qquad \{\ \times \text{ is a relator }\ \} \\
 & (\ll \circ R \times S \circ I \times I)< \\
= & \qquad \{\ \text{theorem 12.44, domains: (10.19)}\ \} \\
 & (\ll \circ R \times S \circ \ll \cup <)< \\
= & \qquad \{\ \text{domains: (10.16)}\ \} \\
 & (\ll \circ R \times S \circ \ll \cup)< \\
= & \qquad \{\ \bullet \quad S \neq \perp\!\!\perp, \text{ theorem 12.37 }\ \} \\
 & R<
\end{array}
$$

□

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 12.47 (Split Right and Left Domain)**

(a) $(P \vartriangle Q)> \;=\; P> \sqcap Q>$

(b) $(P \vartriangle Q)< \;=\; \ll \cup \circ P \circ Q \cup \circ \gg \sqcap I$

**Proof** Simple application of the definition of split and right domain together with axiom (12.23) for part (a) and the definition of split together with property (10.18) for part (b).

□

**Corollary 12.48** $((R \circ S) \vartriangle T)< \;=\; (R \vartriangle (T \circ S\cup))<$

**Proof** By 12.47(b) the left side is equal to

$$\ll \cup \circ R \circ S \circ T\cup \circ \gg \sqcap I$$

Using the same rule and elementary properties of reverse the right side can also be written in the same way.

□

### 12.3.5 Bottom Strictness

In this section we explore circumstances in which a split or product of two specs is equal to $\bot\!\!\bot$. The conclusion of the section is that product and split are both "bottom-strict". That is, if either of their arguments is $\bot\!\!\bot$ then their result is $\bot\!\!\bot$. The key observation is the following:

**Lemma 12.49** $\quad I \vartriangle R \;=\; \bot\!\!\bot \quad\equiv\quad R{>} \;=\; \bot\!\!\bot \quad .$

**Proof**

$$
\begin{array}{ll}
 & I \vartriangle R \;=\; \bot\!\!\bot \\
\equiv & \qquad \{ \text{ domains: (dual of) D9(d) } \} \\
 & (I \vartriangle R){>} \;=\; \bot\!\!\bot \\
\equiv & \qquad \{ \text{ split right domain: 12.47(a) } \} \\
 & I{>} \sqcap R{>} \;=\; \bot\!\!\bot \\
\equiv & \qquad \{ \text{ domains: (10.27) and (10.4) } \} \\
 & R{>} \;=\; \bot\!\!\bot
\end{array}
$$

□

Using lemma 12.49 we can determine exactly when a split is $\bot\!\!\bot$.

**Lemma 12.50** $\quad R \vartriangle S \;=\; \bot\!\!\bot \quad\equiv\quad R \circ S^{\cup} \;=\; \bot\!\!\bot \quad .$

**Proof**

$$
\begin{array}{ll}
 & R \vartriangle S \;=\; \bot\!\!\bot \\
\equiv & \qquad \{ \text{ domains: D9(d) } \} \\
 & (R \vartriangle S){<} \;=\; \bot\!\!\bot \\
\equiv & \qquad \{ \text{ corollary 12.48 } \} \\
 & ((R \circ S^{\cup}) \vartriangle I){<} \;=\; \bot\!\!\bot \\
\equiv & \qquad \{ \text{ domains: D9(d) (twice), and lemma 12.49 } \} \\
 & R \circ S^{\cup} \;=\; \bot\!\!\bot
\end{array}
$$

□

Now, applying 12.50 we get a condition for a product to be $\bot\!\!\bot$.

**Lemma 12.51** $\quad R \times S \;=\; \bot\!\!\bot \quad\equiv\quad R \circ \top\!\!\top \circ S^{\cup} \;=\; \bot\!\!\bot \quad .$

**Proof**

$$
\begin{array}{ll}
 & R \times S \;=\; \perp\!\!\perp \\
\equiv & \qquad \{ \text{ definition: 12.17, lemma 12.50 } \} \\
 & R \circ \ll \circ \gg^{\cup} \circ S^{\cup} \;=\; \perp\!\!\perp \\
\equiv & \qquad \{ \text{ theorem 12.34 } \} \\
 & R \circ \top\!\!\top \circ S^{\cup} \;=\; \perp\!\!\perp
\end{array}
$$

□

If we assume the cone rule then (see lemma 12.1) the right side of lemma 12.51 is equivalent to $R = \perp\!\!\perp \;\vee\; S = \perp\!\!\perp$. Thus we conclude:

**Theorem 12.52 (Strictness of Split and Product)** For all specsa $R$ and $S$,

$$R \vartriangle S = \perp\!\!\perp \;\Leftarrow\; R = \perp\!\!\perp \;\vee\; S = \perp\!\!\perp \quad .$$

Moreover, assuming the cone rule,

$$R \times S = \perp\!\!\perp \;\Leftarrow\; R = \perp\!\!\perp \;\vee\; S = \perp\!\!\perp \quad .$$

**Proof** Straightforward application of 12.50 and 12.51 using the hints given above.

□

### 12.3.6 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 $\pi : \mathcal{F} \stackrel{\cdot}{\longleftarrow} \underline{C}$ such that for every set $D$ and every natural transformation $\sigma : \mathcal{F} \stackrel{\cdot}{\longleftarrow} \underline{D}$ 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 $\pi_X$ , $\pi_Y$ of projections. The terminality of $\pi$ 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, an important theorem.

**Theorem 12.53 (Domain Trading)**

$$P \vartriangle Q \quad = \quad (P \circ Q{>}) \;\vartriangle\; (Q \circ P{>})$$

**Proof**

$$\begin{array}{ll}
 & (P \circ Q{>}) \;\vartriangle\; (Q \circ P{>}) \\
= & \qquad \{ \text{ domains (10.13), monotypes (10.2) } \} \\
 & (P \circ (P{>} \sqcap Q{>})) \;\vartriangle\; (Q \circ (P{>} \sqcap Q{>})) \\
= & \qquad \{ \; P{>} \sqcap Q{>} \text{ is an imp, } \sqcap\text{-distributivity } \} \\
 & (P \vartriangle Q) \circ (P{>} \sqcap Q{>}) \\
= & \qquad \{ \text{ (12.47(a)), (10.13) } \} \\
 & P \vartriangle Q
\end{array}$$

□

**Corollary 12.54** For monotypes $A$ and $B$,

$$A \sqsupseteq Q{>} \;\wedge\; B \sqsupseteq P{>} \quad \Rightarrow \quad P \vartriangle Q \;=\; (P \circ A) \;\vartriangle\; (Q \circ B)$$

**Proof** Follows immediately from the domain trading rule by monotonicity.
□

The significance of theorem 12.53 is that it is *not* dualisable to disjoint sum. As we shall see, a sum of two specs is truly disjoint. Theorem 12.53, on the other hand, says that a split (and hence also a product) is not disjoint; the two operands interact with each other.

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

**Theorem 12.55 (Unique Extension Property)**
Suppose $\trianglelefteq \in \{\sqsubseteq, =, \sqsupseteq\}$. Assume also that

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

Then

$$X \trianglelefteq P \vartriangle Q \;\equiv\; \ll \circ X \trianglelefteq P \circ Q{>} \;\wedge\; \gg \circ X \trianglelefteq Q \circ P{>}$$

**Proof** The $\Rightarrow$-part of the equivalence follows from the computation rule (theorem 12.33) and monotonicity. For the other part we assume the righthand side of the equivalence and prove the validity of the lefthand side:

$$\begin{array}{cl}
& P \vartriangle Q \\
= & \quad \{ \text{ lemma 12.53) } \} \\
& (P \circ Q{\scriptstyle >}) \vartriangle (Q \circ P{\scriptstyle >}) \\
= & \quad \{ \text{ definition of split } \} \\
& \ll^{\cup} \circ\ (P \ \circ\ Q{\scriptstyle >}) \quad \sqcap \quad \gg^{\cup} \circ\ (Q \ \circ\ P{\scriptstyle >}) \\
\unrhd & \quad \{ \text{ rhs is assumed true, monotonicity } \} \\
& \ll^{\cup} \circ \ll \circ\ X \quad \sqcap \quad \gg^{\cup} \circ \gg \circ\ X \\
\unrhd & \quad \{ \text{ assumption } \} \\
& X
\end{array}$$

□

More often than not we apply the theorem with the variable "$\unlhd$" 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 12.55, the cost in terms of burden of proof being typically almost negligible.

The assumption in theorem 12.55 is somewhat unwieldy; however, it is important to note that it is *not* equivalent to $X$ being an imp. (It is however implied by that circumstance when $\unlhd$ is instantiated to equality.) 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 elimination property.

**Theorem 12.56 (Split Elimination)**

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

$$P \vartriangle Q \ \unlhd\ R \vartriangle S \quad \equiv \quad P \circ Q{\scriptstyle >} \ \unlhd\ R \circ S{\scriptstyle >} \quad \wedge \quad Q \circ P{\scriptstyle >} \ \unlhd\ S \circ R{\scriptstyle >}$$

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

$$\begin{array}{cl}
& \ll^{\cup} \circ \ll \circ\ (R \vartriangle S) \quad \sqcap \quad \gg^{\cup} \circ \gg \circ\ (R \vartriangle S) \\
= & \qquad \{ \text{ computation rule for split } \}
\end{array}$$

$$
\begin{array}{ll}
 & \ll^{\cup} \circ\, R \circ S{>} \;\sqcap\; \gg^{\cup} \circ\, S \circ R{>} \\
= & \qquad \{ \text{ definition of split } \} \\
 & (R \circ S{>}) \vartriangle (S \circ R{>}) \\
= & \qquad \{ \text{ lemma 12.53 } \} \\
 & R \vartriangle S
\end{array}
$$

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 12.57 (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.

□

### 12.3.7 Naturality Properties

Part (a) of lemma 12.28 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 12.58 (Naturality of Split)**

For all specs $R$,$S$ and $T$, and all imps $f$,

(a) $\vartriangle \;\in\; (R \times S \stackrel{\cdot}{\leftsquigarrow} T) \stackrel{\cdot}{\leftsquigarrow} (R \stackrel{\cdot}{\leftsquigarrow} T) \times (S \stackrel{\cdot}{\leftsquigarrow} T)$

(b) $\vartriangle \;\in\; (R \times S \stackrel{\cdot}{\leftrightsquigarrow} f) \stackrel{\cdot}{\leftsquigarrow} (R \stackrel{\cdot}{\leftrightsquigarrow} f) \times (S \stackrel{\cdot}{\leftrightsquigarrow} f)$

**Proof** We supply the full details in the case of part (a) only.

$$
\begin{array}{ll}
 & \vartriangle \;\in\; (R \times S \;\dot{\leftsquigarrow}\; T) \;\dot{\leftsquigarrow}\; (R \;\dot{\leftsquigarrow}\; T) \times (S \;\dot{\leftsquigarrow}\; T) \\
\equiv & \qquad \{\ (11.4),\ \ \vartriangle \text{ is a function } \} \\
 & \forall(\underline{U},\underline{V} ::\ \ \vartriangle.\underline{U}\,\langle R \times S \;\dot{\leftsquigarrow}\; T\rangle \vartriangle.\underline{V} \\
 & \qquad \Leftarrow\ \ \underline{U}\,\langle (R \;\dot{\leftsquigarrow}\; T) \times (S \;\dot{\leftsquigarrow}\; T)\rangle\,\underline{V} \\
 & \ ) \\
\equiv & \qquad \{\ \text{definition of } \times \text{ in a higher-order algebra }\ \} \\
 & \forall(U1, U2, V1, V2 :: \\
 & \qquad\qquad U1 \vartriangle U2\,\langle R \times S \;\dot{\leftsquigarrow}\; T\rangle\, V1 \vartriangle V2 \\
 & \qquad \Leftarrow\ \ U1\,\langle R \;\dot{\leftsquigarrow}\; T\rangle\, V1 \;\wedge\; U2\,\langle S \;\dot{\leftsquigarrow}\; T\rangle\, V2 \\
 & \ ) \\
\equiv & \qquad \{\ \text{definition of } \dot{\leftsquigarrow}\ \ \} \\
 & \forall(U1, U2, V1, V2 :: \\
 & \qquad\qquad (R \times S) \circ (V1 \vartriangle V2) \;\sqsupseteq\; (U1 \vartriangle U2) \circ T \\
 & \qquad \Leftarrow\ \ R \circ V1 \;\sqsupseteq\; U1 \circ T \;\wedge\; S \circ V2 \;\sqsupseteq\; U2 \circ T \\
 & \ ) \\
\Leftarrow & \qquad \{\ \text{theorem 12.28(a); split-spec fusion theorem 12.30(a)}\ \} \\
 & \forall(U1, U2, V1, V2 :: \\
 & \qquad\qquad (R \circ V1) \vartriangle (S \circ V2) \;\sqsupseteq\; (U1 \circ T) \vartriangle (U2 \circ T) \\
 & \qquad \Leftarrow\ \ R \circ V1 \;\sqsupseteq\; U1 \circ T \;\wedge\; S \circ V2 \;\sqsupseteq\; U2 \circ T \\
 & \ ) \\
\equiv & \qquad \{\ \text{monotonicity of } \vartriangle\ \ \} \\
 & \mathbf{true}
\end{array}
$$

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

The proof of (b) can be obtained by suitably modifying the above proof, appealing to split-imp fusion (part (c) of theorem 12.30) rather than split-spec fusion.

□

Since product is a (binary) relator we can simply instantiate theorem 11.5 to obtain:

**Theorem 12.59 (Naturality of Product)**

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

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

□

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

**Theorem 12.60 (Naturality of Left and Right Projection)**

For all specs $R$, and all total specs $S$ (i.e. specs $S$ s.t. $S{>} = I$)
(a) $\ll\ \in\ R \dot{\leftrightsquigarrow} R \times S$ and $\gg\ \in\ R \dot{\leftrightsquigarrow} S \times R$
In particular, for all specs $R$,
(b) $\ll\ \in\ R \dot{\leftrightsquigarrow} R \times \top\!\!\top$ and $\gg\ \in\ R \dot{\leftrightsquigarrow} \top\!\!\top \times R$
and
(c) $\ll\ \in\ R \dot{\leftrightsquigarrow} R \times I$ and $\gg\ \in\ R \dot{\leftrightsquigarrow} I \times R$
and, for all specs $R$ and $S$,
(d) $\ll\ \in\ R \dot{\leftsquigarrow} R \times S$ and $\gg\ \in\ R \dot{\leftsquigarrow} S \times R$

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

**Proof**

$$\begin{array}{ll}
 & \ll\ \in\ R \dot{\leftrightsquigarrow} R \times S \\
\equiv & \qquad \{ \text{ definition of } \dot{\leftrightsquigarrow} \ \} \\
 & R \circ \ll\ =\ \ll \circ\ (R \times S) \\
\equiv & \qquad \{ \text{ computation rule (12.36) } \} \\
 & R \circ \ll\ =\ R \circ \ll \circ\ (I \times S{>}) \\
\equiv & \qquad \{ \text{ assumption: } S{>} = I, \text{ theorem 12.44 } \} \\
 & \textbf{true}
\end{array}$$

The corollaries (b) and (c) are just instances of (a). Part (d) follows because $\dot{\leftsquigarrow}$ 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 12.61 (Naturality of Composition)**
For all $\dot{\sim}\ \in \{\dot{\leftsquigarrow},\ \dot{\rightsquigarrow},\ \dot{\leftrightsquigarrow}\}$, and all specs $P$, $Q$ and $R$,

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

□

### 12.3.8 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 12.62 (Split-Cap and Product-Cap Abide Laws)**

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

**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.

$$
\begin{array}{ll}
 & (P \vartriangle Q) \sqcap (R \vartriangle S) \\
= & \quad \{\ (12.15)\ ;\ \text{plat calculus}\ \} \\
 & \ll^{\cup} \circ P \;\sqcap\; \ll^{\cup} \circ R \;\sqcap\; \gg^{\cup} \circ Q \;\sqcap\; \gg^{\cup} \circ S \\
= & \quad \{\ \ll^{\cup} \text{ and } \gg^{\cup} \text{ are co-imps}\ \} \\
 & \ll^{\cup} \circ (P \sqcap R) \;\sqcap\; \gg^{\cup} \circ (Q \sqcap S) \\
= & \quad \{\ (12.15)\ \} \\
 & (P \sqcap R) \vartriangle (Q \sqcap S)
\end{array}
$$

□

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

**Theorem 12.63**

(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 (12.62) and (12.63): the split and product operators are positively $\sqcap$-junctive. They are *not* universally $\sqcap$-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 12.64** Let $\mathcal{V}$ be a non-empty bag of pairs $(V_\lambda, V_\rho)$ of specs, $L = \sqcap(V : V \in \mathcal{V} : V_\lambda)$ and $R = \sqcap(V : V \in \mathcal{V} : V_\rho)$ . Then

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

**Proof**

$$
\begin{array}{ll}
 & \sqcap(V : V \in \mathcal{V} :\ \ll^{\cup} \circ\ V_\lambda\ \sqcap \gg^{\cup} \circ\ V_\rho) \\
= & \quad \{ \text{ quantifier calculus } \} \\
 & \sqcap(V : V \in \mathcal{V} :\ \ll^{\cup} \circ\ V_\lambda)\ \sqcap\ \sqcap(V : V \in \mathcal{V} :\ \gg^{\cup} \circ\ V_\rho) \\
= & \quad \{ \ll \text{ and } \gg \text{ are imps and } \mathcal{V} \text{ is non-empty } \} \\
 & \ll^{\cup} \circ\ L\ \sqcap \gg^{\cup} \circ\ R \\
= & \quad \{ \text{ (12.15) } \} \\
 & L \vartriangle R
\end{array}
$$

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

□

In particular, split and product are $\sqcap$-continuous. Although they are not $\sqcup$-junctive, they are $\sqcup$-continuous:

**Theorem 12.65** Let $\mathcal{V}$ be a linear bag of pairs $(V_\lambda, V_\rho)$ 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**

$$
\begin{array}{ll}
 & \ll^{\cup} \circ\ L\ \sqcap \gg^{\cup} \circ\ R \\
= & \quad \{ \text{ universal } \sqcup\text{-junctivity of composition } \} \\
 & \sqcup(V : V \in \mathcal{V} :\ \ll^{\cup} \circ\ V_\lambda)\ \sqcap\ \sqcup(V : V \in \mathcal{V} :\ \gg^{\cup} \circ\ V_\rho) \\
= & \quad \{ \text{ quantifier calculus } \} \\
 & \sqcup(V, W : V, W \in \mathcal{V} :\ \ll^{\cup} \circ\ V_\lambda\ \sqcap \gg^{\cup} \circ\ W_\rho) \\
= & \quad \{\ \mathcal{V} \text{ is linear, diagonalization } \} \\
 & \sqcup(V : V \in \mathcal{V} :\ \ll^{\cup} \circ\ V_\lambda\ \sqcap \gg^{\cup} \circ\ V_\rho) \\
= & \quad \{ \text{ (12.15) } \} \\
 & \sqcup(V : V \in \mathcal{V} : V_\lambda \vartriangle V_\rho)
\end{array}
$$

□

## 12.4 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 $\sqcup$-junctive but not universally $\sqcap$-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 (when indeed it is possible). The reader should therefore be on their guard as we are on ours.)

### 12.4.1 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 12.66 (Junc-Sum and Sum-Sum Fusion)**

(a) $\quad (P \triangledown Q) \circ (R{+}S) \;=\; (P \circ R) \;\triangledown\; (Q \circ S)$
(b) $\quad (P{+}Q) \circ (R{+}S) \;=\; (P \circ R) \;+\; (Q \circ S)$
$\Box$

**Proof** Transliteration of the proof of theorem 12.28.
$\Box$

**Corollary 12.67** $+$ is a relator.
$\Box$

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

**Theorem 12.68 (Spec-Junc Fusion)**

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

□

Comparison should be made with theorem 12.30 where a restriction to imps had to be made in order to obtain an equality.

### 12.4.2 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 12.31 has a trivial counterpart; the following is the counterpart of lemma 12.32.

**Lemma 12.69**

(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 next theorem (although the property is not really the dual of the strictness of product).

**Theorem 12.70 (Co-strictness of Sum)**

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

□

The proof is elementary.

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

**Theorem 12.71 (Computation Rules for Junc)**

(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 12.69.

**Theorem 12.72**

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

□

### 12.4.3 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 12.73**

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

**Proof** Immediate from the computation rule (12.71) combined with (12.69).
□

**Corollary 12.74** $\hookrightarrow$ 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 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 12.75 (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.) Part (c) follows immediately from the fact that $+$ is a relator.
□

### 12.4.4 Left and Right Domains

Lemma 12.73 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 12.76**

(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 12.77**

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

**Proof**

$$
\begin{array}{ll}
 & \hookrightarrow{<} \sqcap \hookleftarrow{<} \\
= & \quad \{ \text{ domains are monotypes, (10.2) } \} \\
 & \hookrightarrow{<} \circ \hookleftarrow{<} \\
= & \quad \{ \text{ theorem 12.76(b) } \} \\
 & I + \perp\!\!\perp \circ \perp\!\!\perp + I \\
= & \quad \{ \text{ relator.+ } \} \\
 & \perp\!\!\perp + \perp\!\!\perp \\
= & \quad \{ \text{ costrictness of sum: theorem 12.70 } \} \\
 & \perp\!\!\perp
\end{array}
$$
□

For many purposes a weaker form of theorem 12.77 suffices.

**Corollary 12.78**

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

**Proof**

$$
\begin{array}{ll}
 & \hookrightarrow \sqcap \hookleftarrow \quad = \quad \perp\!\!\!\perp \\
\equiv & \qquad \{ \text{ lemma 12.1 } \} \\
 & (\hookrightarrow \sqcap \hookleftarrow)_< \quad = \quad \perp\!\!\!\perp \\
\Leftarrow & \qquad \{ \text{ monotonicity } \} \\
 & \hookrightarrow_< \ \sqcap \ \hookleftarrow_< \quad = \quad \perp\!\!\!\perp
\end{array}
$$

□

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 12.79**

$$
\begin{array}{lrcl}
\text{(a)} & (P{+}Q)_> & = & P_> \ + \ Q_> \\
\text{(b)} & (P{+}Q)_< & = & P_< \ + \ Q_< \\
\text{(c)} & (P \triangledown Q)_< & = & P_< \ \sqcup \ Q_< \\
\text{(d)} & (P \triangledown Q)_> & = & P_> \ + \ Q_>
\end{array}
$$

□

**Proof** The proofs of (a), (b) and (c) can all be obtained by transliterating the proofs of the corresponding properties of cartesian product. By (10.20), (d) follows if we can establish that

$$\top\!\!\!\top \circ (P_> {+} Q_>) \quad = \quad \top\!\!\!\top \circ (P \triangledown Q)$$

This we now do.

$$
\begin{array}{ll}
 & \top\!\!\!\top \circ (P_> {+} Q_>) \\
= & \qquad \{ \text{ definition of sum, theorem 12.68 } \} \\
 & (\top\!\!\!\top \circ \hookrightarrow \circ P_>) \ \triangledown \ (\top\!\!\!\top \circ \hookleftarrow \circ Q_>) \\
= & \qquad \{ \text{ (10.20) } \}
\end{array}
$$

$$
\begin{array}{ll}
 & (\top\!\!\top \circ \hookrightarrow\!> \circ P\!>) \;\triangledown\; (\top\!\!\top \circ \hookleftarrow\!> \circ Q\!>) \\
= & \qquad \{\ (12.76)\ \} \\
 & (\top\!\!\top \circ P\!>) \;\triangledown\; (\top\!\!\top \circ Q\!>) \\
= & \qquad \{\ (10.20),\ \text{theorem } 12.68\ \} \\
 & \top\!\!\top \circ (P \triangledown Q)
\end{array}
$$

□

### 12.4.5 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 12.80 (Cancellation Properties)**

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

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

(b) $\quad P{+}Q \ \trianglelefteq\ R{+}S \ \equiv\ P \trianglelefteq R \ \wedge\ Q \trianglelefteq S$

**Proof**

(a)

$$
\begin{array}{ll}
 & P \triangledown Q \ \trianglelefteq\ R \triangledown S \\
\Rightarrow & \qquad \{\ \text{monotonicity}\ \} \\
 & P \triangledown Q \circ \hookrightarrow \ \trianglelefteq\ R \triangledown S \circ \hookrightarrow \ \wedge\ P \triangledown Q \circ \hookleftarrow \ \trianglelefteq\ R \triangledown S \circ \hookleftarrow \\
\equiv & \qquad \{\ \text{computation rules}\ \} \\
 & P \trianglelefteq R \ \wedge\ Q \trianglelefteq S \\
\Rightarrow & \qquad \{\ \text{monotonicity}\ \} \\
 & P \triangledown Q \ \trianglelefteq\ R \triangledown S
\end{array}
$$

(b)

$$
\begin{array}{ll}
 & P{+}Q \ \trianglelefteq\ R{+}S \\
\equiv & \qquad \{\ \text{definition of sum, (a)}\ \} \\
 & \hookrightarrow \circ P \ \trianglelefteq\ \hookrightarrow \circ R \ \wedge\ \hookleftarrow \circ Q \ \trianglelefteq\ \hookleftarrow \circ S \\
\Rightarrow & \qquad \{\ \text{compose on the left with } \hookrightarrow\!\cup \text{ and } \hookleftarrow\!\cup,\ \text{lemma } 12.73\ \} \\
 & P \trianglelefteq R \ \wedge\ Q \trianglelefteq S \\
\Rightarrow & \qquad \{\ \text{monotonicity}\ \}
\end{array}
$$

$$P{+}Q \;\trianglelefteq\; R{+}S$$

□

**Corollary 12.81 (Junc Initiality)**

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

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

**Proof** By the definition of sum, (12.18) and spec-junc fusion, (12.68),

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

Initiality thus follows immediately.

□

### 12.4.6 Naturality Properties

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

**Theorem 12.82 (Naturality of Left and Right Injection)**
For all specs $R$ and $S$,

(a) $\hookrightarrow \;\in\; R{+}S \;\dot{\leftrightsquigarrow}\; R$
(b) $\hookleftarrow \;\in\; R{+}S \;\dot{\leftrightsquigarrow}\; S$

**Proof** Immediate from the computation rules and the definition of $\dot{\leftrightsquigarrow}$ .

□

The naturality property of junc is also stronger.

**Theorem 12.83 (Naturality of Junc and Sum)**
For all specs $R$, $S$, $T$ and $U$ and all $\dot{\sim} \;\in\; \{\dot{\leftsquigarrow}, \dot{\rightsquigarrow}, \dot{\leftrightsquigarrow}\}$,

(a) $\triangledown \;\in\; (R \mathbin{\dot{\sim}} S{+}T) \;\dot{\leftsquigarrow}\; (R \mathbin{\dot{\sim}} S) \times (R \mathbin{\dot{\sim}} T)$
(b) $+ \;\in\; (R{+}S \mathbin{\dot{\sim}} T{+}U) \;\dot{\leftsquigarrow}\; (R \mathbin{\dot{\sim}} T) \times (S \mathbin{\dot{\sim}} U)$

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

$$
\begin{array}{ll}
 & \triangledown \;\in\; (R \mathrel{\dot{\sim}} S{+}T) \mathrel{\dot{\lesssim}} (R \mathrel{\dot{\sim}} S) \times (R \mathrel{\dot{\sim}} T) \\
\equiv & \qquad \{ \text{ theorem 11.4 } \} \\
 & \forall(U,V,W,X :: \\
 & \quad U \triangledown V \;\langle R \mathrel{\dot{\sim}} S{+}T \rangle\; W \triangledown X \quad \Leftarrow \quad U \,\langle R \mathrel{\dot{\sim}} S \rangle\, W \;\wedge\; V \,\langle R \mathrel{\dot{\sim}} T \rangle\, X \\
 & )
\end{array}
$$

We now continue with the quantified expression.

$$
\begin{array}{ll}
 & U \triangledown V \;\langle R \mathrel{\dot{\sim}} S{+}T \rangle\; W \triangledown X \quad \Leftarrow \quad U \;\langle R \mathrel{\dot{\sim}} S \rangle\; W \;\wedge\; V \;\langle R \mathrel{\dot{\sim}} T \rangle\; X \\
\equiv & \qquad \{ \text{ definition of } \dot{\sim} \text{ } \} \\
 & R \,\circ\, (W \triangledown X) \;\trianglelefteq\; (U \triangledown V) \,\circ\, (S{+}T) \\
 & \quad \Leftarrow \quad R \circ W \;\trianglelefteq\; U \circ S \quad \wedge \quad R \circ X \;\trianglelefteq\; V \circ T \\
\equiv & \qquad \{ \text{ theorems 12.68 and 12.66 } \} \\
 & (R \circ W) \triangledown (R \circ X) \;\trianglelefteq\; (U \circ S) \triangledown (V \circ T) \\
 & \quad \Leftarrow \quad R \circ W \;\trianglelefteq\; U \circ S \quad \wedge \quad R \circ X \;\trianglelefteq\; V \circ T \\
\equiv & \qquad \{ \text{ monotonicity } \} \\
 & \mathbf{true}
\end{array}
$$

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

### 12.4.7 Junctivity Properties

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

**Theorem 12.84 (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** The proof technique is the same in all four cases. We make do, therefore, with a proof of (c) as illustration.

Applying the initiality property (12.81), property (c) reduces to three properties:

$$\begin{array}{lcl} (P \triangledown Q) \sqcap (R \triangledown S) \circ I{+}I & = & (P \triangledown Q) \sqcap (R \triangledown S) \\ (P \triangledown Q) \sqcap (R \triangledown S) \circ \hookrightarrow & = & P \sqcap R \\ (P \triangledown Q) \sqcap (R \triangledown S) \circ \hookleftarrow & = & Q \sqcap S \quad . \end{array}$$

The validity of these three is easily seen by applying theorem 10.30 (noting that $I{+}I$, $\hookrightarrow$ and $\hookleftarrow$ are all imps) followed by junc-sum fusion in the case of the first equation and the computation rule in the case of the second two equations.
□

Again, more can be shown: Both junc and sum are positively $\sqcap$-junctive and universally $\sqcup$-junctive. Hence they are $\sqcup$- and $\sqcap$-continuous.

**Theorem 12.85** Let $\mathcal{V}$ be a bag of pairs $(V_\lambda, V_\rho)$ of specs, $L = \bigsqcup (V : V \in \mathcal{V} : V_\lambda\,)$ and $R = \bigsqcup (V : V \in \mathcal{V} : V_\rho\,)$ . Then

(a) $\quad L \triangledown R \;=\; \bigsqcup (V : V \in \mathcal{V} : V_\lambda \triangledown V_\rho\,)$
(b) $\quad L{+}R \;=\; \bigsqcup (V : V \in \mathcal{V} : V_\lambda{+}V_\rho\,)$
□

**Theorem 12.86** Let $\mathcal{V}$ be a non-empty bag of pairs $(V_\lambda, V_\rho)$ of specs, $L = \bigsqcap (V : V \in \mathcal{V} : V_\lambda\,)$ and $R = \bigsqcap (V : V \in \mathcal{V} : V_\rho\,)$ . Then

(a) $\quad L \triangledown R \;=\; \bigsqcap (V : V \in \mathcal{V} : V_\lambda \triangledown V_\rho\,)$
(b) $\quad L{+}R \;=\; \bigsqcap (V : V \in \mathcal{V} : V_\lambda{+}V_\rho\,)$

**Proof**

$$\begin{array}{cl} & L \triangledown R \\ \sqsubseteq & \quad \{ \ \triangledown \ \text{ is monotonic } \} \\ & \bigsqcap (V : V \in \mathcal{V} : V_\lambda \circ \hookrightarrow_\cup \ \sqcup \ V_\rho \circ \hookleftarrow_\cup\,) \\ \sqsubseteq & \quad \{ \text{ quantifier calculus } \} \\ & \bigsqcap (V : V \in \mathcal{V} : V_\lambda \circ \hookrightarrow_\cup\,) \ \sqcup \ \bigsqcap (V : V \in \mathcal{V} : V_\rho \circ \hookleftarrow_\cup\,) \\ & \quad \sqcup \ (\, \top\!\!\top \circ \hookrightarrow_\cup \ \sqcap \ \top\!\!\top \circ \hookleftarrow_\cup\,) \\ = & \quad \{ \text{ (12.77) } \} \\ & \bigsqcap (V : V \in \mathcal{V} : V_\lambda \circ \hookrightarrow_\cup\,) \ \sqcup \ \bigsqcap (V : V \in \mathcal{V} : V_\rho \circ \hookleftarrow_\cup\,) \\ = & \quad \{ \hookrightarrow_\cup \ \text{ is an imp and } \mathcal{V} \text{ is non-empty } \} \\ & \bigsqcap (V : V \in \mathcal{V} : V_\lambda\,) \circ \hookrightarrow_\cup \ \sqcup \ \bigsqcap (V : V \in \mathcal{V} : V_\rho\,) \circ \hookleftarrow_\cup \\ = & \quad \{ \text{ definitionof } \triangledown : \text{ (12.16) } \} \\ & L \triangledown R \end{array}$$

□

## 12.5 Basic Simulations and 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, $\mathbb{1}+\mathbb{1}$ and $\mathbb{1}\times(\mathbb{1}+\mathbb{1})$ are monotypes, and the functions $\mathbb{1}+$ and $(\mathbb{1}\times\mathbb{1})+$ are unary relators. Relators and monotypes built in this way we call *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 by constructing a series of elementary natural isomorphisms between combinations of the polymorphic relators. One simulation is also calculated.

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.

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

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 $+\mathbb{1}$. (In general any monotype may be used in place of $\mathbb{1}$.)

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

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

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

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

To complete this task we must exhibit a spec, $\gamma$, satisfying three quite strong conditions. We can make progress in this task by temporarily setting aside two of the conditions, *constructing* $\gamma$ 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 $\gamma$:

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

Thus the constructed spec is $\gamma$ where

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

It remains to show that $\gamma$ 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 $\gamma$ 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.

$$\begin{array}{ll}
 & \gamma \quad \text{is a co-imp} \\
\Leftarrow & \qquad \{\ \text{theorem 12.75(b)}\ \} \\
 & (I\times\hookrightarrow \quad\text{and}\quad I\times\hookleftarrow \quad \text{are co-imps}\ ) \\
 & \wedge\ \ (I\times\hookrightarrow)< \;\sqcap\; (I\times\hookleftarrow)< \;=\; \perp\!\!\perp
\end{array}$$

$$
\begin{array}{ll}
\equiv & \quad \{ \text{ theorem } 12.41 \ \} \\
& (I \times \hookrightarrow)_< \sqcap (I \times \hookleftarrow)_< = \bot\!\!\bot \\
\equiv & \quad \{ \text{ theorem } 12.45 \ \} \\
& (I \times \hookrightarrow_<) \sqcap (I \times \hookleftarrow_<) = \bot\!\!\bot \\
\equiv & \quad \{ \text{ lemma } 12.62\text{(b)} \ \} \\
& I \times (\hookrightarrow_< \sqcap \hookleftarrow_<) = \bot\!\!\bot \\
\equiv & \quad \{ \text{ theorem } 12.77,\ I \times \bot\!\!\bot = \bot\!\!\bot \ \} \\
& \textbf{true}
\end{array}
$$

We now calculate the left domain of $\gamma$.

$$
\begin{array}{ll}
& \gamma_< \\
= & \quad \{ \text{ definition of } \gamma\text{, theorem } 12.79\text{(c)} \ \} \\
& (I \times \hookrightarrow)_< \sqcup (I \times \hookleftarrow)_< \\
= & \quad \{ \text{ theorems } 12.45 \text{ and } 12.76 \ \} \\
& I \times (I + \bot\!\!\bot) \sqcup I \times (\bot\!\!\bot + I) \\
= & \quad \{ \text{ lemma } 12.63\text{(e)} \ \} \\
& I \times ((I + \bot\!\!\bot) \sqcup (\bot\!\!\bot + I)) \\
= & \quad \{ \text{ definition of } + \ \} \\
& I \times (I + I)
\end{array}
$$

Finally, we calculate the right domain of $\gamma$.

$$
\begin{array}{ll}
& \gamma_> \\
= & \quad \{ \text{ definition of } \gamma\text{, theorem } 12.79\text{(d)} \ \} \\
& (I \times \hookrightarrow)_> + (I \times \hookleftarrow)_> \\
= & \quad \{ \text{ theorems } 12.45 \text{ and } 12.76 \ \} \\
& (I \times I) + (I \times I)
\end{array}
$$

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 [91], 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 $\gamma$ 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 catamorphisms 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 12.87 (Split-Junc Abide Law)**

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

**Proof** We aim to use the initiality property (theorem 12.81) of junc. First note that

$$\begin{array}{ll} & (R \triangledown S) \vartriangle (T \triangledown U) \ \circ \ I{+}I \\ = & \qquad \{ \ I{+}I \text{ is a monotype and thus an imp, fusion: 12.30(c)} \ \} \\ & (R \triangledown S \ \circ \ I{+}I) \vartriangle (T \triangledown U \ \circ \ I{+}I) \\ = & \qquad \{ \text{ split fusion (12.28) } \} \\ & (R \triangledown S) \vartriangle (T \triangledown U) \end{array}$$

Hence:

$$\begin{array}{ll} & (R \triangledown S) \vartriangle (T \triangledown U) \quad = \quad (R \vartriangle T) \triangledown (S \vartriangle U) \\ \equiv & \qquad \{ \text{ theorem 12.81 combined with the above } \} \\ & (R \triangledown S) \vartriangle (T \triangledown U) \ \circ \hookrightarrow \quad = \quad R \vartriangle T \\ & \wedge \quad (R \triangledown S) \vartriangle (T \triangledown U) \ \circ \hookleftarrow \quad = \quad S \vartriangle U \end{array}$$

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

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

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

The properties of the natural isomorphism $\gamma$ that we anticipated above can now be given.

**Theorem 12.88**

$$
\begin{array}{lll}
\text{(a)} & \gamma \circ (R \vartriangle S){+}(R \vartriangle T) \;=\; (R \circ I \triangledown I) \vartriangle (S{+}T) \\
\text{(b)} & R \times (S \triangledown T) \circ \gamma \;=\; (R \times S) \;\triangledown\; (R \times T)
\end{array}
$$

**Proof**

$$
\begin{array}{ll}
\text{(a)} & \gamma \circ (R \vartriangle S){+}(R \vartriangle T) \\
= & \quad \{ \text{ definition of } \gamma, \text{ junc fusion theorem 12.66(a) } \} \\
 & (I \times \hookrightarrow \circ R \vartriangle S) \;\triangledown\; (I \times \hookleftarrow \circ R \vartriangle T) \\
= & \quad \{ \text{ split fusion theorem 12.28(a) } \} \\
 & (R \vartriangle (\hookrightarrow \circ S)) \;\triangledown\; (R \vartriangle (\hookleftarrow \circ T)) \\
= & \quad \{ \text{ abides law (12.87) } \} \\
 & (R \triangledown R) \;\vartriangle\; ((\hookrightarrow \circ S) \;\triangledown\; (\hookleftarrow \circ T)) \\
= & \quad \{ \text{ junc fusion (12.66a), definition of sum (12.18) } \} \\
 & (R \circ I \triangledown I) \;\vartriangle\; (S{+}T)
\end{array}
$$

$$
\begin{array}{ll}
\text{(b)} & R \times (S \triangledown T) \circ \gamma \\
= & \quad \{ \text{ definition of } \gamma, \text{ spec-junc fusion (12.68) } \} \\
 & (R \times (S \triangledown T) \circ I \times \hookrightarrow) \;\triangledown\; (R \times (S \triangledown T) \circ I \times \hookleftarrow)) \\
= & \quad \{ \times \text{ is a relator, junc computation rules (12.71) } \} \\
 & (R \times S) \;\triangledown\; (R \times T)
\end{array}
$$

□

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.

$$
\begin{array}{lrcl}
(12.89) & \lambda(R ::\;\; \perp\!\!\perp) & \cong & \lambda(R ::\;\; R \times \perp\!\!\perp) \\
(12.90) & \lambda(R ::\;\; R{+}\perp\!\!\perp) & \cong & \lambda(R ::\;\; R) \\
(12.91) & \lambda(R, S ::\;\; R{+}S) & \cong & \lambda(R, S ::\;\; S{+}R) \\
(12.92) & \lambda(R, S, T ::\;\; R{+}(S{+}T)) & \cong & \lambda(R, S, T ::\;\; (R{+}S){+}T)
\end{array}
$$

$$\text{(12.93)} \quad \lambda(R,S :: \; R\times S) \;\cong\; \lambda(R,S :: \; S\times R)$$
$$\text{(12.94)} \quad \lambda(R,S,T :: \; R\times (S\times T)) \;\cong\; \lambda(R,S,T :: \; (R\times S)\times T)$$
$$\text{(12.95)} \quad \lambda(R :: \; R) \;\cong\; \lambda(R :: \; R\times \mathbb{1})$$
$$\text{(12.96)} \quad \lambda(R,S,T :: \; R\times (S{+}T)) \;\cong\; \lambda(R,S,T :: \; (R\times S){+}(R\times T))$$

Of these (12.89) is trivial (the isomorphism is $\perp\!\!\!\perp$ itself) and (12.90) and (12.96) we have already discussed. Hints on how to prove (12.91)-(12.95) are given below. Of course the reader may wish to ignore the hints altogether.

*Hints:* Isomorphisms (12.91) and (12.92) can be constructed using the same startegy as that used to construct (12.96). In the case of (12.91) the very short calculations that are necessary can be made yet shorter by noting that the constructed isomorphism is its own reverse.

Isomorphisms (12.93) and (12.94) require a somewhat different strategy. The reason is that the natural isomorphism properties of split and product only help in the construction of up-formations (see definition 11.9) and not transformations. Moreover, whereas the calculation of the right domain of a split is straightforward, the calculation of its left domain is not (compare 12.47(a) with 12.47(b)). To add to this, split preserves imps but does not preserve co-imps. One may avoid all these difficulties by constructing two up-formations, 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 (12.93) these "two" up-formations 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 (12.95) is a case apart. Note that $\ll$ is an up-formation of the identity relator by $\times\mathbb{1}$. Try restricting $\ll$ so that its right domain is $I\times\mathbb{1}$ and then verify that your conjectured isomorphism meets all the requirements. You may find that theorem 12.46 is helpful. *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.)

$$\text{(12.97)} \quad \perp\!\!\!\perp \;=\; \alpha_1 \circ R \vartriangle \perp\!\!\!\perp$$
$$\text{(12.98)} \quad R \triangledown \perp\!\!\!\perp \circ \alpha_2 \;=\; R$$
$$\text{(12.99)} \quad R \triangledown S \circ \alpha_3 \;=\; S \triangledown R$$
$$\text{(12.100)} \quad R \triangledown (S \triangledown T) \circ \alpha_4 \;=\; (R \triangledown S) \triangledown T$$

$$\begin{array}{lrcl}
(12.101) & R \vartriangle S & = & \alpha_5 \circ S \vartriangle R \\
(12.102) & R \vartriangle (S \vartriangle T) & = & \alpha_6 \circ (R \vartriangle S) \vartriangle T \\
(12.103) & R & = & \alpha_7 \circ R \vartriangle !
\end{array}$$

Rest assured: all have very trivial proofs. Note the pattern: the relators + and $\times$ have been systematically replaced by their corresponding catamorphism and $\mathbb{1}$ has been replaced by its catamorphism !. The $\alpha$'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 $\gamma$ satisfying requirement (a) of a natural simulation (see definition 11.10).

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

We have thus shown that

$$(\hookrightarrow \times \hookrightarrow) \triangledown (\hookleftarrow \times \hookleftarrow) \in (R{+}S) \times (T{+}U) \;\dot{\leftsquigarrow\!\!\rightsquigarrow}\; (R \times T){+}(S \times U)$$

We continue to call the constructed spec $\gamma$.

Clearly, $\gamma$ is an imp; it is also a co-imp although this requires more (routine) effort to establish.

$$
\begin{array}{ll}
& \text{co-imp}.((\hookrightarrow \times \hookrightarrow) \;\triangledown\; (\hookleftarrow \times \hookleftarrow)) \\
\Leftarrow & \qquad \{ \;\; 12.75(\text{b}) \;\; \} \\
& \text{co-imp}.(\hookrightarrow \times \hookrightarrow) \;\wedge\; \text{co-imp}.(\hookleftarrow \times \hookleftarrow) \\
& \wedge \;\; (\hookrightarrow \times \hookrightarrow)_< \sqcap (\hookleftarrow \times \hookleftarrow)_< \;=\; \perp\!\!\perp \\
\equiv & \qquad \{ \;\; \text{relator}.\times,\; \hookrightarrow \text{ and } \hookleftarrow \text{ are both bijections} \;\; \} \\
& (\hookrightarrow \times \hookrightarrow)_< \sqcap (\hookleftarrow \times \hookleftarrow)_< \;=\; \perp\!\!\perp \\
\equiv & \qquad \{ \;\; \text{relator}.\times,\;\; \text{theorem } 12.76 \;\; \} \\
& (I + \perp\!\!\perp) \times (I + \perp\!\!\perp) \;\sqcap\; (\perp\!\!\perp + I) \times (\perp\!\!\perp + I) \;\;=\;\; \perp\!\!\perp \\
\equiv & \qquad \{ \;\; \text{cap-sum and cap-product abide laws: (12.84) and (12.62)} \;\; \} \\
& ((I \sqcap \perp\!\!\perp) + (\perp\!\!\perp \sqcap I)) \;\times\; ((I \sqcap \perp\!\!\perp) + (\perp\!\!\perp \sqcap I)) \;\;=\;\; \perp\!\!\perp \\
\equiv & \qquad \{ \;\; \text{co-strictness of sum, strictness of product} \;\; \} \\
& \mathbf{true}
\end{array}
$$

Calculation of its right domain using (12.79), (12.45) and (12.76) is straightforward. We obtain

$$\gamma_> \;\;=\;\; (I \times I) \;+\; (I \times I)$$

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

We leave it as an exercise for the reader to show that $\gamma$ does not witness an isomorphism between the two relators. To do this first verify that

$$\gamma_< \;\;=\;\; (I + \perp\!\!\perp) \times (I + \perp\!\!\perp) \;\sqcup\; (\perp\!\!\perp + I) \times (\perp\!\!\perp + I)$$

and then use this to show that $\gamma_<$ is properly included in $(I + I) \times (I + I)$.

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 $\gamma$. 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 $\gamma$ for the final calculation which is to verify that

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

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

This concludes our discussion of the elementary properties of the polynomial relators.

# Chapter 13

# Initial Datatypes and Catamorphisms

A fundamental argument for the use of type information in the design of large programs is that the structure of the program is governed by the structure of the data. A well-established example is the use of recursive descent to structure the parsing (and compilation) of strings defined by a context-free grammar; here the structure of the data is defined by its grammar and the structure of the parsing program is identical. The idea is extended in the denotational description of programming languages where a fundamental initial step is the definition of so-called domain equations; those familiar with denotational semantics know that once this step has been taken the later steps are often relatively mundane and straightforward. Users of strongly-typed languages like Pascal will argue strongly that the effective use of type declarations is extremely important for subsequent program development, and even users of untyped languages like Lisp will admit that the programming errors that they make are often caused by type violations. A fundamental goal of our research is therefore to develop calculi of program construction that lay bare the oneness of program and data structure.

An example of a programming formalism in which this oneness plays the rôle of a major design principle is the theory of types developed by Martin-Löf. In this theory each type is defined by four sets of rules one of which is the set of so-called introduction rules and another is a singleton set containing the so-called elimination rule for the type. (The remaining sets are not relevant to the present discussion.) The introduction rules describe the structure of the elements of the type whereas the elimination rule says how to construct functions over the elements of the type. As has been argued elsewhere [8], the introduction rules completely define the type in the sense that all other rules (including the elimination rule) are systematically derived from them. The elimination rule is added in order to express the notion that "nothing else" is in the type other than the elements that can be constructed via the introduction rules by stating that the structure of functions on elements of the type is completely governed by the structure of these rules.

In the algebraic approach that we are currently pursuing a different (although formally equivalent) approach is taken to the definition of data types and in particular to expressing the notion that "nothing else" is in the type other than the elements constructed via its introduction rules. Nevertheless, the underlying principle is that a data type is a structured set of elements that is equipped with a mechanism governed by that structure for defining functions on the elements of the type. For the benefit of readers who may not be familiar with it we now outline this approach as it pertains to functional programming. Other readers will probably wish to skip the next two paragraphs; all they need to know is that we use the term "catamorphism" to refer to $F$-homomorphisms whose domain is an initial $F$-algebra. (We are currently in the process of extending our work to terminal algebras but none of that work is reported here.)

The approach involves several stages building up to the definition of a "universal object" in a category of algebras. First, in place of the introduction rules in Martin-Löf's system the notion of endofunctor is of paramount importance. An endofunctor is (in this context) a pair of functions, one from types to types and the other from functions to functions. Typically, both functions are denoted by the same symbol. Suppose $F$ is an endofunctor, $A$ and $B$ are types and $f$ and $g$ are functions of composable type. Let $I_A$ denote the identity function on the type $A$. Then it is required that

$$\begin{array}{lrcl} & F.f \in F.A \longleftarrow F.B & \Leftarrow & f \in A \longleftarrow B \\ & F.I_A & = & I_{F.A} \\ \text{and} & F.(f \circ g) & = & F.f \circ F.g \end{array}$$

Without seeing some examples it is difficult for the uninitiated to envisage the correspondence between a number of introduction rules and an endofunctor. For the moment let us just remark that typically an endofunctor will take the form of a disjoint sum of other more primitive functors, and that each term in such a sum corresponds to one introduction rule. The next step is to define an $F$-algebra as a pair consisting of a type $A$ and a function $f \in A \longleftarrow F.A$. (Note that if, indeed, the endofunctor $F$ is a disjoint sum of other functors then the function $f$ can be broken down into distinct components each being applicable to elements introduced by one of the corresponding introduction rules.) The data type defined by the endofunctor $F$ is then an $F$-algebra satisfying a so-called "universality property", namely that there is a unique homomorphism from the data type to each $F$-algebra. Such homomorphisms take the place of the eliminators in Martin-Löf's theory. To emphasise their special rôle we shall give them the name "catamorphism".

An example would be the data type natural number. Roughly speaking, $\mathbb{N}$ has the property

$$\mathbb{N} \quad = \quad \{0\}+\mathbb{N}$$

where "+" denotes the disjoint sum of two types. (According to this definition the elements of $\mathbb{N}$ are $\hookrightarrow.0$ and $\hookleftarrow.n$ where $n$ ranges over $\mathbb{N}$ and $\hookrightarrow$ and $\hookleftarrow$ denote the injection functions associated with disjoint sum. You should interpret "$\hookrightarrow.0$" as zero and "$\hookleftarrow$" as the successor function. Our discussion has been phrased in terms of "$\{0\}$" rather than the unit type "$\mathbb{1}$" in order to make the link with standard terminology a little more accessible.) More formally, we recognise in this equation an endofunctor "$\{0\}+$". This is a function that maps the type $A$ to the type $\{0\}+A$. But it may also be extended to map functions to functions by defining $\{0\}+f$ to be that function $g$ such that $g \circ \hookrightarrow$ is the constant function always returning $\hookrightarrow.0$, and $g \circ \hookleftarrow \; = \; \hookleftarrow \circ f$. (Moreover, it satisfies all the properties required of a functor, but that we leave to the reader to verify.) A $\{0\}+$-algebra is a set together with a constant and a unary operator (these being zero and the successor function in the case of the natural numbers), and a $\{0\}+$-homomorphism is just what one would normally understand by a homomorphism of an algebraic structure, in this case a function $\phi$, say, from one $\{0\}+$-algebra $(A, a, \sigma)$, say, to another $(B, b, \tau)$, say, that maps the constant of the first to the constant of the second

i.e. $\qquad \phi.a \quad = \quad b$

and commutes with the unary operator of the first replacing it with that of the second

i.e. $\qquad \phi \circ \sigma \quad = \quad \tau \circ \phi$

That $\mathbb{N}$ is "universal" in the class of $\{0\}+$-algebras just means that for any $\{0\}+$-algebra $(A, a, \sigma)$, say, there is a unique homomorphism mapping $\mathbb{N}$ to $A$. With a suitable definition of the operators it is also easily shown that $\{0\}+\mathbb{N}$ is a $\{0\}+$-algebra satisfying the universality property. Thus, $\mathbb{N}$ is a fixed point of the endofunctor $\{0\}+$ in the sense that there are homomorphisms mapping $\mathbb{N}$ to $\{0\}+\mathbb{N}$ and vice-versa which (on account of their uniqueness) are each others' inverses.

To summarise this discussion: in the framework of functional programming datatypes are fixed points of endofunctors on which are defined what we call "catamorphisms", i.e. homomorphisms satisfying a uniqueness and universality property. This is not the place to discuss the practicality of catamorphisms as a program structuring method, that being something that we intend to address in future publications. We hope however that we have provided sufficient background to motivate the calculations that follow in this section. Specifically, we explore the extension of the notion of a (functional) catamorphism to relations. For this we need the notion of endorelator instead of endofunctor. We begin by discussing the least fixed point of an endorelator and then introduce our definition of a (relational) catamorphism.

From now on we assume that $F$ is an endorelator.

## 13.1 Initial Datatypes

Since endorelators are, by definition, monotonic the Knaster-Tarski theorem asserts the existence of their fixed points, in particular least and greatest. We hope shortly to report on our work on greatest fixed points but in the present paper we restrict our attention to least fixed points. Specifically, the least fixed point of the endorelator $F$, here denoted by $\mu F$, has the defining properties

(13.1) $\mu F \quad = \quad F.\mu F$

and, for all $X$,

(13.2) $X \ \sqsupseteq \ \mu F \quad \Leftarrow \quad X \ \sqsupseteq \ F.X$

We shall refer to (13.2) as the *induction principle*.

The following lemma is about all we can say about $\mu F$ at this stage. Nevertheless, it is a necessary first step.

**Lemma 13.3** $\mu F$ is a monotype.

**Proof**

$$
\begin{array}{ll}
 & \mu F \text{ is a monotype} \\
\equiv & \quad \{ \text{ definition } \} \\
 & I \ \sqsupseteq \ \mu F \\
\Leftarrow & \quad \{ \text{ induction principle (13.2) } \} \\
 & I \ \sqsupseteq \ F.I \\
\equiv & \quad \{ \ F \text{ is a relator (10.33a) } \} \\
 & \textbf{true}
\end{array}
$$

□

## 13.2 Catamorphisms Defined

**Definition 13.4** For endorelator $F$ we define a function, denoted by $(\![F;\ \_\,]\!)$, by the properties that, for all specs $R$,

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

and for all specs $R$ and $X$,

(b) $\quad X \ \sqsupseteq \ (\![F;\ R]\!) \ \Leftarrow \ X \ \sqsupseteq \ R \circ F.X$

□

In other words, $(\![F;\ R]\!)$ is the least solution to the equation

$$X :: \qquad X \ = \ R \circ F.X$$

Its well-definedness is thus guaranteed by the Knaster-Tarski theorem.

We call specs of the form $(\![F;\ R]\!)$ *catamorphisms* (or $F$-*catamorphisms* when we particularly wish to be explicit about $F$) and we verbalise $(\![F;\ R]\!)$ as "($F$-)catamorphism $R$", omitting the qualification "$F$" when there is no doubt about the relator in question.

For reasons that will only become clear later, we call 13.4(a) the *computation rule* for catamorphisms. We call 13.4(b) the *induction principle* for catamorphisms. (The relationship with the induction principle for $\mu F$ will become evident after we have established that $\mu F$ is itself a catamorphism — see theorem 13.20.)

One may well raise one's eyebrows at the unconventional "banana brackets" we have chosen to denote catamorphisms. The reasoning behind this choice is based on envisaged applications: typically, the relators one encounters in programming problems are formed using the disjoint-sum operator. Consequently, the catamorphism constructor will be applied to a junc of specs having the same number of components as that of the associated relator. The use of special brackets thus avoids the otherwise inevitable pair of parentheses. For several examples of such applications see [53] and [80]. The notation was first introduced by Malcolm in [63] and [64].

The catamorphism $(\!|F;\ I|\!)$ is of particular importance since it is clearly the least fixed point of $F$. Thus, we have:

$$(13.5)\quad \mu F \;=\; (\!|F;\ I|\!)$$

From now on we omit the argument "$F$" within the catamorphism brackets and write just "$(\!|R|\!)$" instead of "$(\!|F;\ R|\!)$".

## 13.3 The Unique Extension Property

The definition of a catamorphism is clear enough but with its two distinct parts it is not well suited to calculational purposes. We proceed now to prove two properties that predict a single-statement definition of catamorphism. The first is simple enough.

**Theorem 13.6**

$$(\!|R|\!) \;=\; (\!|R|\!) \circ \mu F$$

**Proof**

$$\begin{array}{ll} & (\!|R|\!) \;=\; (\!|R|\!) \circ \mu F \\ \equiv & \quad \{ \text{ lemma 13.3 } \} \\ & (\!|R|\!) \circ \mu F \;\sqsupseteq\; (\!|R|\!) \end{array}$$

$$
\begin{array}{ll}
\Leftarrow & \quad \{ \text{ induction principle for catamorphisms: (13.4b) } \} \\
& ([R]) \circ \mu F \;\; \sqsupseteq \;\; R \circ F.(([R]) \circ \mu F) \\
\equiv & \quad \{ \; F \text{ is a relator (10.33c), } \mu F \text{ is a fixed point of } F \; \} \\
& ([R]) \circ \mu F \;\; \sqsupseteq \;\; R \circ F.([R]) \circ \mu F \\
\equiv & \quad \{ \text{ computation rule for catamorphisms: (13.4a) } \} \\
& \mathbf{true}
\end{array}
$$

□

Note that the theorem could equally well have been formulated as

$$\mu F \;\; \sqsupseteq \;\; ([R])\triangleright$$

We have now established that $([R])$ satisfies two equations, namely,

$$
\begin{array}{llccl}
& X :: & X & = & R \circ F.X \\
\text{and} & X :: & X & = & X \circ \mu F
\end{array}
$$

Obviously, therefore, it also satisfies the third equation

$$(13.7) \quad X :: \qquad X \;\; = \;\; R \circ F.X \circ \mu F$$

The important insight contained in the next theorem is that the set of specs simultaneously solving the first two equations is identical to the set of solutions of the third equation.

**Theorem 13.8**

$$X = R \circ F.X \circ \mu F \quad \equiv \quad X = X \circ \mu F \;\; \wedge \;\; X = R \circ F.X$$

**Proof**

$$
\begin{array}{ll}
& X \circ \mu F = X \;\; \wedge \;\; X = R \circ F.X \\
\equiv & \quad \{ \text{ substitution } \} \\
& X \circ \mu F = X \;\; \wedge \;\; X = R \circ F.(X \circ \mu F) \\
\equiv & \quad \{ \; F \text{ is a relator (10.33c), } \mu F \text{ is a fixed point of } F \; \} \\
& X \circ \mu F = X \;\; \wedge \;\; X = R \circ F.X \circ \mu F \\
\equiv & \quad \{ \Rightarrow \text{ is obvious; } \Leftarrow \text{ by (13.3) and (10.1) } \} \\
& X = R \circ F.X \circ \mu F
\end{array}
$$

□

More significantly, equation (13.7) has a *unique* solution, which we shall now prove. It will come as no surprise that a goal in our proof is to invoke the induction principle (13.2). How we do so is, in our view, particularly elegant and offers an excellent illustration of the benefits to be gained from a systematic development of a theory taking account of clearly stated calculational rules, in this case the Galois connection between factors and composition.

Suppose $P$ and $Q$ are two solutions to (13.7). I.e.

$$P = R \circ F.P \circ \mu F \tag{13.9}$$

$$Q = R \circ F.Q \circ \mu F \tag{13.10}$$

Since $P$ and $Q$ are completely symmetrical our task reduces to showing that $Q \sqsupseteq P$. We use factor theory and the induction principle to prove this property as follows.

$$\begin{array}{ll}
 & Q \sqsupseteq P \\
\equiv & \quad \{ \ P = \{ \ (13.9), \text{theorem } 13.8 \ \} \ P \circ \mu F \ \} \\
 & Q \sqsupseteq P \circ \mu F \\
\equiv & \quad \{ \ (9.1a) \ \} \\
 & P \backslash Q \sqsupseteq \mu F \\
\Leftarrow & \quad \{ \ \text{induction principle } (13.2) \ \} \\
 & P \backslash Q \sqsupseteq F.(P \backslash Q) \\
\equiv & \quad \{ \ (9.1a) \ \} \\
 & Q \sqsupseteq P \circ F.(P \backslash Q) \\
\equiv & \quad \{ \ (13.9), (13.10), \text{theorem } 13.8 \ \} \\
 & R \circ F.Q \sqsupseteq R \circ F.P \circ F.(P \backslash Q) \\
\Leftarrow & \quad \{ \ F \text{ is a relator, monotonicity of composition} \ \} \\
 & F.Q \sqsupseteq F.(P \circ P \backslash Q) \\
\equiv & \quad \{ \ (9.2a), \text{monotonicity of relators} \ \} \\
 & \mathbf{true}
\end{array}$$

In conclusion we have:

**Corollary 13.11 (Unique Extension Property)**
For all specs $X$ and $R$,

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

□

## 13.4 Consequences of the UEP

Were we obliged to refer to one theorem in the paper that is the most important of all then it would be the above unique extension property. It will be used so often below that we will refer to it within proof hints simply as "uep". A first example is the simple but nevertheless useful identity rule:

**Lemma 13.12 (Identity Rule)**

$$\mu F \;=\; (\!|\mu F|\!)$$

**Proof**

$$\begin{array}{ll} & \mu F \;=\; (\!|\mu F|\!) \\ \equiv & \qquad \{ \text{ uep: (13.11) } \} \\ & \mu F \;=\; \mu F \circ F.\mu F \circ \mu F \\ \equiv & \qquad \{ \text{ (13.1), (13.3) and (10.1) } \} \\ & \textbf{true} \end{array}$$

□

(Later — see theorem 13.20 — we shall see various other ways in which $\mu F$ can be expressed as a catamorphism.)

Also, the coincidence in $(\!|R|\!)$ of the least and greatest solutions of (13.7) together with the Knaster-Tarski theorem gives:

**Theorem 13.13**

(a) $\quad X = (\!|R|\!) \;\Leftarrow\; X = R \circ F.X \circ \mu F$

(b) $\quad X \sqsupseteq (\!|R|\!) \;\Leftarrow\; X \sqsupseteq R \circ F.X \circ \mu F$

(c) $\quad X \sqsubseteq (\!|R|\!) \;\Leftarrow\; X \sqsubseteq R \circ F.X \circ \mu F$

□

A corollary of the above that figures very prominently in program calculations is

**Corollary 13.14 (Catamorphism Fusion)**

(a) $\quad U \circ (\!|V|\!) = (\!|R|\!) \;\Leftarrow\; U \circ V = R \circ F.U$

(b) $\quad U \circ (\!|V|\!) \sqsupseteq (\!|R|\!) \;\Leftarrow\; U \circ V \sqsupseteq R \circ F.U$

(c) $\quad U \circ (\!|V|\!) \sqsubseteq (\!|R|\!) \;\Leftarrow\; U \circ V \sqsubseteq R \circ F.U$

**Proof** Let $\trianglelefteq \in \{=, \sqsupseteq, \sqsubseteq\}$. Then

$$
\begin{array}{ll}
 & U \circ (\!|V|\!) \;\trianglelefteq\; (\!|R|\!) \\
\Leftarrow & \quad \{\ (13.13)\ \} \\
 & U \circ (\!|V|\!) \;\trianglelefteq\; R \circ F.(U \circ (\!|V|\!)) \circ \mu F \\
\equiv & \quad \{\ \text{uep: } (13.11);\ (10.33c)\ \} \\
 & U \circ V \circ F.(\!|V|\!) \circ \mu F \;\trianglelefteq\; R \circ F.U \circ F.(\!|V|\!) \circ \mu F \\
\Leftarrow & \quad \{\ \text{invariance of } \trianglelefteq \text{ under composition}\ \} \\
 & U \circ V \;\trianglelefteq\; R \circ F.U
\end{array}
$$

□

The importance of fusion laws has already been stressed in our discussion of the polynomial relators. In earlier publications [4, 62] we used the term "promotion" property, this term having been used by Bird to name a technique for improving the efficiency of programs [15] and which our notion captured and generalised. In so doing it was our explicit intention to "promote" the recognition and use of such laws in program transformation. Maarten Fokkinga suggested the more descriptive term "fusion" property, and we have been glad to adopt his suggestion. Our use of the term here is yet another generalisation that we have no doubt will prove to be just as important.

**Theorem 13.15 (Monotonicity)**

$$(\!|R|\!) \sqsupseteq (\!|S|\!) \;\Leftarrow\; R \sqsupseteq S$$

**Proof**

$$
\begin{array}{ll}
 & (\!|R|\!) \sqsupseteq (\!|S|\!) \\
\Leftarrow & \quad \{\ \text{fusion } (13.14b),\ U := I\ \} \\
 & I \circ R \sqsupseteq S \circ F.I \\
\Leftarrow & \quad \{\ (10.33a)\ \} \\
 & R \sqsupseteq S
\end{array}
$$

□

## 13.5 Further Properties of Catamorphisms

Our next series of calculations is motivated by the wish to determine the dependency of the left and right domain of $(\!|R|\!)$ on the left and right domain of $R$. We also wish to determine to what extent "functionality", "injectivity", "surjectivity" and "totality" properties are maintained by catamorphism construction.

An alternative motivation might be that we wish to verify the type inference rules

$$(\![R]\!) \in A \sim \mu F \quad \Leftarrow \quad R \in A \sim F.A$$

and

$$(\![R]\!) \in A \longleftarrow \mu F \quad \Leftarrow \quad R \in A \longleftarrow F.A$$

We do indeed verify these rules but would stress once again that they are included primarily to enable the reader to relate theorems about "<" and ">" to conventional mechanisms for expressing type properties. Statements about "<" and ">" involve one fewer dummy (the universally quantified monotype $A$ in the rules above is not needed) and separate properties of the left from those of the right domain; either of these is sufficient grounds to justify their use in preference to the conventional modes of expression.

We begin with imp and co-imp preservation.

**Lemma 13.16**

$$R \;\sqsupseteq\; (\![S]\!) \circ (\![T]\!)^{\cup} \quad \Leftarrow \quad R \;\sqsupseteq\; S \circ F.R \circ T^{\cup}$$

**Proof**

$$
\begin{array}{ll}
 & R \;\sqsupseteq\; (\![S]\!) \circ (\![T]\!)^{\cup} \\
\equiv & \qquad \{ \text{ left factors (9.1b) } \} \\
 & R/(\![T]\!)^{\cup} \;\sqsupseteq\; (\![S]\!) \\
\Leftarrow & \qquad \{ \text{ induction principle: 13.4(b) } \} \\
 & R/(\![T]\!)^{\cup} \;\sqsupseteq\; S \circ F.(R/(\![T]\!)^{\cup}) \\
\equiv & \qquad \{ \text{ left factors (9.1b) } \} \\
 & R \;\sqsupseteq\; S \circ F.(R/(\![T]\!)^{\cup}) \circ (\![T]\!)^{\cup} \\
\equiv & \qquad \{ \text{ computation rule (13.4a), reverse } \} \\
 & R \;\sqsupseteq\; S \circ F.(R/(\![T]\!)^{\cup}) \circ F.(\![T]\!)^{\cup} \circ T^{\cup} \\
\equiv & \qquad \{ \text{ relators (10.33c) } \} \\
 & R \;\sqsupseteq\; S \circ F.(R/(\![T]\!)^{\cup} \circ (\![T]\!)^{\cup}) \circ T^{\cup} \\
\Leftarrow & \qquad \{ \text{ left factors (9.2b), monotonicity } \} \\
 & R \;\sqsupseteq\; S \circ F.R \circ T^{\cup}
\end{array}
$$

□

**Theorem 13.17 (Imp and co-imp preservation)**
The function $(\!|\_|\!)$ respects (a) imps, (b) co-imps and (c) bijections.

**Proof** Part (a) is a corollary of lemma 13.16 obtained by instantiating $R$ to $I$ and applying monotonicity. Specifically, we have:

$$
\begin{array}{ll}
 & (\!|S|\!) \text{ is an imp} \\
\equiv & \quad \{ \text{ definition } \} \\
 & I \;\sqsupseteq\; (\!|S|\!) \circ (\!|S|\!)^{\cup} \\
\Leftarrow & \quad \{ \text{ lemma 13.16(a) } \} \\
 & I \;\sqsupseteq\; S \circ F.I \circ S^{\cup} \\
\Leftarrow & \quad \{ \text{ (10.33a), monotonicity } \} \\
 & I \;\sqsupseteq\; S \circ S^{\cup} \\
\equiv & \quad \{ \text{ definition } \} \\
 & S \text{ is an imp.}
\end{array}
$$

Co-imp preservation is established by the following argument:

$$
\begin{array}{ll}
 & I \;\sqsupseteq\; (\!|S|\!)^{\cup} \circ (\!|S|\!) \\
\Leftarrow & \quad \{ \text{ transitivity: 13.3 } \} \\
 & \mu F \;\sqsupseteq\; (\!|S|\!)^{\cup} \circ (\!|S|\!) \\
\Leftarrow & \quad \{ \text{ (13.5), catamorphism fusion: 13.14(c) } \} \\
 & I \circ F.(\!|S|\!)^{\cup} \;\sqsupseteq\; (\!|S|\!)^{\cup} \circ S \\
\equiv & \quad \{ \text{ calculus } \} \\
 & F.(\!|S|\!) \;\sqsupseteq\; S^{\cup} \circ (\!|S|\!) \\
\equiv & \quad \{ \text{ computation rule: 13.4(a) } \} \\
 & F.(\!|S|\!) \;\sqsupseteq\; S^{\cup} \circ S \circ F.(\!|S|\!) \\
\Leftarrow & \quad \{ \text{ monotonicity } \} \\
 & I \;\sqsupseteq\; S^{\cup} \circ S
\end{array}
$$

Part (c) is, of course, just the conjunction of (a) and (b).
□

It is interesting to note that the complete proof of theorem 13.17 (that is including the proof of lemma 13.16) uses all the properties of a relator. (Uses of (10.33a) and (10.33c) are explicitly mentioned; use of (10.33b) — the monotonicity of $F$ — occurs along with the use of other monotonicity properties in the last step in the proof of lemma 13.16, and use of (10.33d) is hidden in the preceding step by our use of ambiguous notation.) Indeed, this theorem is crucial to potential uses of our calculus and provides, on its own, much support for the chosen definition of a relator.

We turn now to the relationship between the right and left domains of $(\!|R|\!)$ and those of $R$. We begin with some relatively straightforward observations.

**Theorem 13.18 (Type of catamorphisms)**

(a) $\quad R< \;\sqsupseteq\; (\!|R|\!)<$
(b) $\quad \mu F \;\sqsupseteq\; (\!|R|\!)>$

**Proof** Both parts follow immediately from $(\!|R|\!) = R \circ F.(\!|R|\!) \circ \mu F$ and, respectively, properties (10.17) and (10.22).
□

In the next theorem we show how, without loss of generality, the argument of a catamorphism can be restricted to specs whose domains satisfy certain criteria.

**Theorem 13.19 (Domain Trading)**

For all monotypes $A$ and specs $R$,
(a) $\quad (\!|R|\!) = (\!|R \circ F.A|\!) = (\!|A \circ R|\!) \quad \Leftarrow \quad A \sqsupseteq (R \circ F.A)<$
(b) $\quad (\!|R|\!) = (\!|R \circ F.R<|\!)$
(c) $\quad (\!|R|\!) = (\!|R \circ A|\!) \quad \Leftarrow \quad A \sqsupseteq F.R<$

**Proof** Part (b) is a particular instance of (a) obtained by instantiating $A$ to $R<$. (The antecedent of (a) is easily verified to be true.) Part (c) follows from (b) by monotonicity. So only (a) needs to be proved. We begin by proving equality of the first and second terms.

$$
\begin{array}{ll}
 & (\!|R|\!) = (\!|R \circ F.A|\!) \\
\Leftarrow & \qquad \{ \text{ calculus } \} \\
 & (\!|R|\!) = A \circ (\!|R \circ F.A|\!) \;\wedge\; A \circ (\!|R \circ F.A|\!) = (\!|R \circ F.A|\!) \\
\Leftarrow & \qquad \{ \text{ fusion (13.14), applied twice } \} \\
 & A \circ R \circ F.A = R \circ F.A \;\wedge\; A \circ R \circ F.A = R \circ F.A \circ F.A \\
\equiv & \qquad \{ \; F.A \circ F.A = F.A, \text{ calculus } \} \\
 & A \circ R \circ F.A = R \circ F.A \\
\equiv & \qquad \{ \text{ (10.12) } \} \\
 & A \sqsupseteq (R \circ F.A)<
\end{array}
$$

Now, equality between the second and third terms is straightforward:

$$
\begin{array}{ll}
 & (\![R \circ F.A]\!) \\
= & \qquad \{ \text{ assumption: } A \sqsupseteq (R \circ F.A)_< \ \} \\
 & (\![A \circ R \circ F.A]\!) \\
= & \qquad \{ \ A \sqsupseteq (A \circ R \circ F.A)_<, \text{ theorem above } \} \\
 & (\![A \circ R]\!)
\end{array}
$$

□

A consequence of this domain trading rule is that we can now generalise lemma 13.12 to a very flexible and useful form.

**Theorem 13.20 (Identity Rule)**

(a) $\mu F = (\![A]\!) \ \Leftarrow \ I \sqsupseteq A \sqsupseteq \mu F$
In particular,
(b) $\mu F = (\![\mu F]\!) = (\![I]\!) = (\![F.I]\!)$

**Proof** First we show that $(\![I]\!) = \mu F$.

$$
\begin{array}{ll}
 & (\![I]\!) = \mu F \\
\equiv & \qquad \{ \text{ lemma 13.12 } \} \\
 & (\![I]\!) = (\![\mu F]\!) \\
\Leftarrow & \qquad \{ \text{ domain trading: 13.19(c) } \} \\
 & \mu F \sqsupseteq F.(\mu F)_< \\
\equiv & \qquad \{ \ \mu F = F.\mu F = (\mu F)_< \ \} \\
 & \textbf{true}
\end{array}
$$

Hence

$$
\begin{array}{ll}
 & \mu F = (\![A]\!) \\
\equiv & \\
 & (\![I]\!) = (\![A]\!) = (\![\mu F]\!) \\
\Leftarrow & \qquad \{ \text{ monotonicity: theorem 13.15 } \} \\
 & I \sqsupseteq A \sqsupseteq \mu F
\end{array}
$$

This proves part (a). Part (b) lists some particular instances that we use most commonly.
□

Properties 13.18(a) and (b) may be jointly rephrased as

$$([R]) \in A\sim\mu F \quad \Leftarrow \quad R \in A\sim B$$

In particular,

$$([R]) \in A\sim\mu F \quad \Leftarrow \quad R \in A\sim F.A$$

Moreover, property 13.19(b) implies that catamorphisms may be restricted, quite without loss of generality, to relations $R \in A\sim F.A$ for some monotype $A$. In conventional accounts this restriction is indeed imposed — with the consequence that catamorphisms are no longer total functions. It is precisely these burdensome and highly undesirable type restrictions that our theory tries to avoid!

Theorem 13.18 raises the question as to when the inclusions in (a) and (b) may be strengthened to equalities. This is an important question because the statement

$$([R])< \;=\; R<$$

is interpreted as the statement that $([R])$ maintains any surjectivity property of $R$, whilst the statement

$$([R])> \;=\; \mu F$$

is interpreted as the statement that $([R])$ is total on $\mu F$. Part (a) of the following lemma seems to be the strongest statement that can be made about the surjectivity of catamorphisms; the dual statement (part (b) below) acts as a stepping stone to the desired theorem on totality.

**Lemma 13.21**

(a) $\quad ([R])< \;=\; R< \quad \Leftarrow \quad F.([R])< \;\sqsupseteq\; R>$

(b) $\quad ([R])> \;=\; \mu F \quad \Leftarrow \quad R> \;\sqsupseteq\; F.([R])<$

**Proof**

$$
\begin{array}{ll}
\text{(a)} & R< \;=\; ([R])< \\
\equiv & \quad \{ \text{ computation rule, (13.4a) } \} \\
 & R< \;=\; (R \circ F.([R]))< \\
\equiv & \quad \{ \text{ (10.16) } \} \\
 & R< \;=\; (R \circ F.([R])<)< \\
\Leftarrow & \quad \{ \text{ (10.13) } \} \\
 & F.([R])< \;\sqsupseteq\; R>
\end{array}
$$

(b)
$$
\begin{array}{ll}
 & \llparenthesis R \rrparenthesis{>} \;=\; \mu F \\
\equiv & \qquad \{ \text{ (13.18b) } \} \\
 & \llparenthesis R \rrparenthesis{>} \;\sqsupseteq\; \mu F \\
\Leftarrow & \qquad \{ \text{ induction principle (13.2 } \} \\
 & \llparenthesis R \rrparenthesis{>} \;\sqsupseteq\; F.\llparenthesis R \rrparenthesis{>} \\
\equiv & \qquad \{ \text{ computation rule, (13.4a) } \} \\
 & (R \circ F.\llparenthesis R \rrparenthesis){>} \;\sqsupseteq\; F.\llparenthesis R \rrparenthesis{>} \\
\Leftarrow & \qquad \{ \text{ (10.21) and (10.12) } \} \\
 & R{>} \;\sqsupseteq\; F.\llparenthesis R \rrparenthesis{<}
\end{array}
$$

□

A requirement for totality of $\llparenthesis R \rrparenthesis$ is now easy to derive.

**Theorem 13.22 (Totality)**

$$\llparenthesis R \rrparenthesis{>} \;=\; \mu F \quad \Leftarrow \quad R{>} \sqsupseteq F.R{<}$$

**Proof**

$$
\begin{array}{ll}
 & \llparenthesis R \rrparenthesis{>} \;=\; \mu F \\
\Leftarrow & \qquad \{ \text{ lemma 13.21b } \} \\
 & R{>} \sqsupseteq F.\llparenthesis R \rrparenthesis{<} \\
\Leftarrow & \qquad \{ \text{ theorem 13.18(a), monotonicity } \} \\
 & R{>} \sqsupseteq F.R{<}
\end{array}
$$

□

**Corollary 13.23**

$$\llparenthesis f \rrparenthesis \in A \longleftarrow \mu F \quad \Leftarrow \quad f \in A \longleftarrow F.A$$

**Proof** This is a matter of expanding the definition of the antecedent and consequent and applying the appropriate lemma or theorem. Thus, assume $f \in A \longleftarrow F.A$. Then, by definition and the monotonicity of relators, $f$ is an imp and $f{>} = F.A \sqsupseteq F.f{<}$. Thus, by theorem 13.17, $\llparenthesis f \rrparenthesis$ is an imp, by theorem 13.22, $\llparenthesis f \rrparenthesis{>} = \mu F$, and, by theorem 13.18(a), $A \sqsupseteq \llparenthesis f \rrparenthesis{<}$. I.e. $\llparenthesis f \rrparenthesis \in A \longleftarrow \mu F$.

□

## 13.6 Naturality of Catamorphisms

**Theorem 13.24 (Naturality of catamorphisms)** If $F$ is an endorelator then for all specs $R$

| | |
|---|---|
| (a) | $(\!\|F;\_\|\!) \in (R \overset{\cdot}{\leftsquigarrow} \mu F) \overset{\cdot}{\leftsquigarrow} (R \overset{\cdot}{\leftsquigarrow} F.R)$ |
| (b) | $(\!\|F;\_\|\!) \in (R \overset{\cdot}{\rightsquigarrow} \mu F) \overset{\cdot}{\leftsquigarrow} (R \overset{\cdot}{\rightsquigarrow} F.R)$ |
| (c) | $(\!\|F;\_\|\!) \in (R \overset{\cdot}{\leftrightsquigarrow} \mu F) \overset{\cdot}{\leftsquigarrow} (R \overset{\cdot}{\leftrightsquigarrow} F.R)$ |

**Proof** We prove part (a) only. The proof is very similar to that of theorem 11.5 and the reader should be able to see how to extend the proof of (a) to prove (b) and (c).

$$
\begin{array}{ll}
 & (\!|F;\_|\!) \in (R \overset{\cdot}{\leftsquigarrow} \mu F) \overset{\cdot}{\leftsquigarrow} (R \overset{\cdot}{\leftsquigarrow} F.R) \\
\equiv & \quad \{ \text{ theorem 11.4, definition of } \_{\in}\_{\overset{\cdot}{\leftsquigarrow}}\_ \ \} \\
 & \forall (U,V :: (\!|F;U|\!) \langle R \overset{\cdot}{\leftsquigarrow} \mu F\rangle (\!|F;V|\!) \quad \Leftarrow \quad U \langle R \overset{\cdot}{\leftsquigarrow} F.R\rangle V) \\
\equiv & \quad \{ \text{ definition of } \overset{\cdot}{\leftsquigarrow} \ \} \\
 & \forall (U,V :: R \circ (\!|V|\!) \sqsupseteq (\!|U|\!) \circ \mu F \quad \Leftarrow \quad R \circ V \sqsupseteq U \circ F.R) \\
\equiv & \quad \{ \text{ lemma 13.6; fusion, theorem 13.14 } \} \\
 & \mathbf{true}
\end{array}
$$

□

The reader who has diligently followed through the proofs of theorems 11.5 and 13.24 (and filled in the missing elements) will realise that the theorems combine a number of important properties of relators and catamorphisms — they preserve imps (and co-imps although that didn't play any rôle above), they are monotonic, relators distribute through composition and catamorphisms obey the fusion properties 13.14 (a) to (c).

It is no accident that this is the case. Indeed, it can be said that theorems 11.5 and 13.24 were the initial inspiration for all the research reported here. That is to say, some time after becoming aware of the notion of "natural polymorphism" we specifically set out to develop a theory of datatypes with these two theorems as primary "healthiness criteria". As our work developed we realised that they could be decomposed into more elementary requirements — exactly the theorems presented prior to this subsection. Most important of all we realised that the "naturality" of functors amounted precisely to the definition of "relator" that we have given. (We now run the risk of being criticised for not having included this discussion at a much earlier stage in the presentation. That argument we would counter by saying that a research paper and a logbook are not the same thing.)

Theorem 13.24 reformulates various fusion properties as naturality properties. We conclude this section with a useful lemma whose statement involves a combination of a naturality property and a fusion property.

**Lemma 13.25 (Catamorphism-catamorphism fusion)**
For all $(\trianglelefteq, \dot{\sim}) \in \{(\sqsupseteq, \dot{\leftsquigarrow}),\ (=, \dot{\leftrightsquigarrow}),\ (\sqsubseteq, \dot{\rightsquigarrow})\}$,

$$\llparenthesis F;\ R \rrparenthesis \circ \llparenthesis G;\ S \rrparenthesis \trianglelefteq \llparenthesis G;\ R \circ S \rrparenthesis \quad \Leftarrow \quad S \in F \mathrel{\dot{\sim}} G$$

**Proof**

$$\begin{array}{ll}
 & \llparenthesis F;\ R \rrparenthesis \circ \llparenthesis G;\ S \rrparenthesis \trianglelefteq \llparenthesis G;\ R \circ S \rrparenthesis \\
\Leftarrow & \qquad \{ \text{ catamorphism fusion: theorem 13.14 } \} \\
 & \llparenthesis F;\ R \rrparenthesis \circ S \trianglelefteq R \circ S \circ G.\llparenthesis F;\ R \rrparenthesis \\
\Leftarrow & \qquad \{ \text{ computation rule: 13.4(a) } \} \\
 & R \circ F.\llparenthesis F;\ R \rrparenthesis \circ S \trianglelefteq R \circ S \circ G.\llparenthesis F;\ R \rrparenthesis \\
\Leftarrow & \qquad \{ \text{ transitivity and monotonicity } \} \\
 & S \in F \mathrel{\dot{\sim}} G
\end{array}$$

□

## 13.7 Isomorphic Monotypes and Initial Algebras

In this section we turn to the consideration of monotypes that are isomorphic to $\mu F$. We have two reasons to do so. The first is in order to relate our own theory to other theories, in particular to those based on category theory. Since categorical approaches characterise types only "up to isomorphism" we are obliged to pitch the discussion at this more general level. The second, and more important reason, is that types are often represented by a variety of isomorphic algebraic structures. The natural numbers, for example, can be represented by unary numerals (zero and the successor operator of Peano arithmetic), binary numerals, decimal numerals etc. Lists may be *cons* lists (an algebraic structure having a constant *nil* and a binary "cons" operator that appends new elements to the front of a list), *snoc* lists ( a similar structure but with a "snoc" operator that appends new elements to the end of a list) or join lists (a structure having a constant *nil*, a unary singleton-list-forming operation, and an associative join operation on pairs of lists). Such isomorphic instances of a type arise through the use of different relators and so have different associated catamorphisms. Choosing the right instance can be the key to the design of efficient programs, but it is also necessary to be able to relate the catamorphisms of isomorphic types.

### 13.7.1 Initial $F$-Algebras Defined

In category theory types are defined by means of initial algebras (see e.g. [73]). In order to set the scene we introduce the definitions of an "$F$-algebra" and an "initial" $F$-algebra.

**Definition 13.26** An $F$-*algebra* is a pair $(C,\tau)$ such that:

(a) $C$ is a monotype, and
(b) $\tau \in C \longleftarrow F.C$
□

**Definition 13.27** An *initial* $F$-algebra is a triple $(C,\tau,\eta)$ such that

(a) the pair $(C,\tau)$ is an $F$-algebra.

Furthermore, $\eta$ is a function from imps to imps with the property that for all $F$-algebras $(A,f)$:

(b) $\eta.f \in A \longleftarrow C$

and is the unique solution in $A \longleftarrow C$ of the equation

$$g :: \quad g \circ \tau = f \circ F.g$$

That is, for all $g \in A \longleftarrow C$,

(c) $g = \eta.f \equiv g \circ \tau = f \circ F.g$

Condition (c) is referred to as the *initiality condition.*
□

In definition 13.27 we have been at pains to reproduce the conventional definition of an initial $F$-algebra as closely as possible within our system whilst nevertheless giving a definition that is amenable to calculation. It is an ugly definition because of the provisos on the rules pertaining to the use of $\eta.f$. A rephrasing of the definition helps in later calculations: Note the similarity between (c) and the unique extension property. The similarity increases if we can prove:

(13.28) $g \circ \tau = f \circ F.g \quad\equiv\quad g = f \circ F.g \circ \tau^{\cup}$

for $g \in A \longleftarrow C$ and $f \in A \longleftarrow F.A$.

$$
\begin{array}{ll}
 & g \circ \tau = f \circ F.g \;\equiv\; g = f \circ F.g \circ \tau^{\cup} \\
\Leftarrow & \quad \{ \text{ mutual implication and Leibniz } \} \\
 & g \circ \tau \circ \tau^{\cup} = g \;\wedge\; F.g \circ \tau^{\cup} \circ \tau = F.g \\
\Leftarrow & \quad \{ \; g \circ C = g, \text{ relator.}F \; \} \\
 & \tau \circ \tau^{\cup} = C \;\wedge\; \tau^{\cup} \circ \tau = F.C
\end{array}
$$

So (13.28) holds if $\tau$ is a bijection to $C$ from $F.C$. To prove this fact involves constructing the inverse of $\tau$.

Simple type inference gives us a potential candidate. Specifically:

(13.29) $\eta.(F.\tau) \in F.C \longleftarrow C$

Thus, we now proceed to verify that $\tau^{\cup} = \eta.(F.\tau)$.

The basis for the verification is theorem (D21) in the appendix which, given the type information that we already have, asserts that it suffices to verify the two properties:

(13.30) $\tau \circ \eta.(F.\tau) \quad = \quad C$

(13.31) $\eta.(F.\tau) \circ \tau \quad = \quad F.C$

Before proceeding it is useful to interpose a minor observation.

(13.32) $C \quad = \quad \eta.\tau$

**Proof**

$$
\begin{array}{ll}
 & C \;=\; \eta.\tau \\
\equiv & \quad \{ \text{ initiality condition: 13.27(c) } \} \\
 & C \circ \tau \;=\; \tau \circ F.C \\
\equiv & \quad \{ \text{ (13.26b) } \} \\
 & \mathbf{true}
\end{array}
$$

□

Now we can continue.

**Proof of (13.30)**

$$
\begin{array}{ll}
 & \tau \circ \eta.(F.\tau) \;=\; C \\
\equiv & \quad \{ \text{ (13.32) } \} \\
 & \tau \circ \eta.(F.\tau) \;=\; \eta.\tau \\
\equiv & \quad \{ \text{ initiality condition: 13.27(c) } \} \\
 & \tau \circ \eta.(F.\tau) \circ \tau \;=\; \tau \circ F.(\tau \circ \eta.(F.\tau)) \\
\Leftarrow & \quad \{ \text{ substitution, relators } \} \\
 & \eta.(F.\tau) \circ \tau \;=\; F.\tau \circ F.(\eta.(F.\tau)) \\
\equiv & \quad \{ \text{ initiality condition: 13.27(c) with } g \;:=\; \eta.(F.\tau) \; \} \\
 & \mathbf{true}
\end{array}
$$

□

**Proof of (13.31)**

$$
\begin{array}{ll}
 & \eta.(F.\tau) \circ \tau \;=\; F.C \\
\equiv & \quad \{ \text{ initiality condition: 13.27(c) } \} \\
 & F.\tau \circ F.(\eta.(F.\tau)) \;=\; F.C \\
\Leftarrow & \quad \{ \text{ substitution, relators } \} \\
 & \tau \circ \eta.(F.\tau) \;=\; C \\
\equiv & \quad \{ \text{ (13.30) } \} \\
 & \mathbf{true}
\end{array}
$$

□

As already explained we conclude by (D21) that $\tau$ is a bijection to $C$ from $F.C$ with:

(13.33) $\tau^{\cup} \;=\; \eta.(F.\tau)$

All together we now have the ingredients of an equivalent definition of initial $F$-algebras:

**Definition 13.34** $(C, \tau, \eta)$ is an initial $F$-algebra equivales the conjunction of

(a) $C$ is a monotype,
(b) $\tau$ is a bijection to $C$ from $F.C$.

and $\eta$ is a function from imps to imps with the properties that, for all $F$-algebras $(A, f)$ and all imps $g \in A \longleftarrow C$,

(c) $\eta.f \in A \longleftarrow C$, and
(d) $g = \eta.f \equiv g = f \circ F.g \circ \tau\cup$

**Proof** We have shown that

$$\begin{array}{ll} & \tau \text{ is a bijection to } C \text{ from } F.C \\ \Rightarrow & \\ & \forall(A, f, g :\ (A, f) \text{ is an } F\text{-algebra} \wedge g \in A \longleftarrow C \\ & \quad :\ g \circ \tau = f \circ F.g \equiv g = f \circ F.g \circ \tau\cup) \end{array}$$

and

$$\begin{array}{ll} & (C, \tau, \eta) \text{ is an initial } F\text{-algebra} \\ \Rightarrow & \\ & \tau \text{ is a bijection to } C \text{ from } F.C \end{array}$$

Simple predicate calculus completes the proof.
□

Since we also have

- $(\mu F, \mu F)$ is an $F$-algebra
- $(\!| f |\!) \in A \longleftarrow \mu F \Leftarrow f \in A \longleftarrow F.A$

(see corollary 13.23) we conclude:

**Corollary 13.35** $(\mu F, \mu F, (\!| F; \_ |\!))$ is an initial $F$-algebra.
□

Our next objective in this section is to prove the more general statement that a monotype $C$ is the first component of an initial $F$-algebra if and only if $C$ is *isomorphic* to $\mu F$. Just as for $\mu F$, we shall, in so doing, characterise such monotypes by somewhat broader properties than the initiality condition; in particular, we establish the existence of "$C$-catamorphisms", i.e. the existence of a function that is total on all specs (rather than just imps), is imp-preserving, and obeys a certain "unique extension property" that when restricted to imps agrees with the required initiality property of "$\eta$" in definition 13.34.

The proof is by mutual implication, the next two subsections being devoted to each part.

### 13.7.2 Isomorphic monotypes

Suppose $C$ is a monotype that is isomorphic to $\mu F$. By definition there is a bijection, $\varepsilon$ say, satisfying:

(13.36) $\varepsilon \circ \varepsilon\cup \;=\; C$

(13.37) $\varepsilon\cup \circ \varepsilon \;=\; \mu F$

Simple consequences of (13.36) and (13.37) are that $\varepsilon< \;=\; C$ and $\varepsilon> \;=\; \mu F$. In particular,

(13.38) $C \circ \varepsilon \;=\; \varepsilon \;=\; \varepsilon \circ \mu F$

We use definition 13.34 to show that $C$ is the first component in an initial $F$-algebra. Our first task is to construct a bijection $\tau$ to $C$ from $F.C$. This we do by type considerations.

By construction of $\tau$:

$\tau$ is a bijection to $C$ from $F.C$

$\equiv$ { $\varepsilon$ is a bijection to $C$ from $\mu F$,
• $\tau := \varepsilon \circ \gamma$ }

$\gamma$ is a bijection to $\mu F$ from $F.C$

$\Leftarrow$ { $\mu F = F.\mu F$ }

$\gamma$ is a bijection to $F.\mu F$ from $F.C$

$\Leftarrow$ { $\varepsilon\cup$ is a bijection to $\mu F$ from $C$,
• $\gamma := F.\varepsilon\cup$ }

**true**

The constructed bijection is thus

(13.39) $\tau \;=\; \varepsilon \circ F.\varepsilon^{\cup}$

Our next (and final) task is to construct a function $\eta$ from imps to imps satisfying 13.34(c) and (d). It is at this point that we make a more general claim. Specifically, our claim is that $C$ enjoys its own form of catamorphism with comparable properties to $F$-catamorphisms, including that of satisfying the initiality condition.

Such a "$C$-catamorphism" is required to be the unique solution to the equation:

$$X :: \quad X \;=\; R \circ F.X \circ \tau^{\cup}$$

(In other words we generalise (d) in definition 13.34 to all specs $R$ and not just imps $f$ of a certain type.) To show that this equation always has a unique solution and simultaneously derive the definition of a $C$-catamorphism we proceed as follows (the goal of the calculation being to remove $X$ from the rhs of the initial equation):

$$
\begin{array}{ll}
 & X \;=\; R \circ F.X \circ \tau^{\cup} \\
\equiv & \quad \{\ (13.39),\ \text{reverse}\ \} \\
 & X \;=\; R \circ F.X \circ F.\varepsilon \circ \varepsilon^{\cup} \\
\equiv & \quad \{\ \text{by } (13.36),\ \varepsilon^{\cup} \circ C \;=\; \varepsilon^{\cup}\ \} \\
 & X \;=\; R \circ F.X \circ F.\varepsilon \circ \varepsilon^{\cup} \quad \wedge \quad X \circ C \;=\; X \\
\equiv & \quad \{\ \text{relators},\ (13.36),\ (13.37)\ \} \\
 & X \circ \varepsilon \;=\; R \circ F.(X \circ \varepsilon) \circ \mu F \quad \wedge \quad X \circ C \;=\; X \\
\equiv & \quad \{\ \text{uep: } (13.11)\ \} \\
 & X \circ \varepsilon \;=\; (\!|F;\ R|\!) \quad \wedge \quad X \circ C \;=\; X \\
\equiv & \quad \{\ (13.36),\ (13.37),\ \text{substitution}\ \} \\
 & X \;=\; (\!|F;\ R|\!) \circ \varepsilon^{\cup} \quad \wedge \quad X \circ C \;=\; X \\
\equiv & \quad \{\ \text{by } (13.36),\ \varepsilon^{\cup} \circ C \;=\; \varepsilon^{\cup}\ \} \\
 & X \;=\; (\!|F;\ R|\!) \circ \varepsilon^{\cup}
\end{array}
$$

In conclusion, we define

(13.40) $(\!|F,\varepsilon;\ R|\!) \;=\; (\!|F;\ R|\!) \circ \varepsilon^{\cup}$

and we have established the unique extension property

(13.41) $R \;=\; (\!|F,\varepsilon;\ S|\!) \;\equiv\; R \;=\; S \circ F.R \circ \tau^{\cup}$

Note that we have introduced yet another parameter into the definition of a catamorphism. This should, however, cause no confusion since the number of parameters clearly identifies the intended definition. Moreover, the multiple use of the catamorphism brackets is justified by the identity

$$(\!|F;\ \ R|\!) \quad = \quad (\!|F, \mu F;\ \ R\,|\!)$$

which identity is easily derived from the properties of the monotype $\mu F$ and the right domain of $(\!|F;\ \ R|\!)$.

Property 13.41 holds the key to many additional properties of such catamorphisms. All the properties of $F$-catamorphisms established in sections 13.3 and 13.5 can be generalised. The generalised properties are almost verbatim repetitions of the originals, only minor modifications being necessary to replace $\mu F$ by $C$, $\tau$ or $\tau^\cup$. Without further ado, therefore, we shall quickly summarise the properties. Note that the order of presentation remains the same as in sections 13.3 and 13.5.

To begin, the computation rule is

$$(\!|F,\varepsilon;\ \ R|\!) \circ \tau \quad = \quad R \circ F.(\!|F,\varepsilon;\ \ R|\!)$$

(Some readers may find this rule more familiar than the earlier one; it is the rule that appears frequently in, for example, [62].)

Second, we have the identity rule:

$$(\!|F,\varepsilon;\ \ \tau\,|\!) \quad = \quad C$$

Third, by invoking the Knaster-Tarski theorem we have:

$$X \;=\; (\!|F,\varepsilon;\ \ R|\!) \quad \Leftarrow \quad X \;=\; R \circ F.X \circ \tau^\cup$$

$$X \;\sqsupseteq\; (\!|F,\varepsilon;\ \ R|\!) \quad \Leftarrow \quad X \;\sqsupseteq\; R \circ F.X \circ \tau^\cup$$

$$X \;\sqsubseteq\; (\!|F,\varepsilon;\ \ R|\!) \quad \Leftarrow \quad X \;\sqsubseteq\; R \circ F.X \circ \tau^\cup$$

from which we may derive just as before the fusion properties:

$$U \circ (\!|F,\varepsilon;\ \ V\,|\!) = (\!|F,\varepsilon;\ \ R|\!) \quad \Leftarrow \quad U \circ V \;=\; R \circ F.U$$

$$U \circ (\!|F,\varepsilon;\ \ V\,|\!) \sqsupseteq (\!|F,\varepsilon;\ \ R|\!) \quad \Leftarrow \quad U \circ V \;\sqsupseteq\; R \circ F.U$$

$$U \circ (\![F,\varepsilon;\ \ V]\!) \sqsubseteq (\![F,\varepsilon;\ \ R]\!) \quad \Leftarrow \quad U \circ V \sqsubseteq R \circ F.U$$

Monotonicity is now an easy consequence:

$$(\![F,\varepsilon;\ \ R]\!) \sqsupseteq (\![F,\varepsilon;\ \ S]\!) \quad \Leftarrow \quad R \sqsupseteq S$$

That the function $(\![F,\varepsilon;\ \ \_]\!)$ respects (a) imps, (b) co-imps and (c) bijections follows from the fact that it is the composition of two functions, namely $(\![\_\,]\!)$ and $(\circ(\varepsilon\cup))$, that themselves respect imps, co-imps and bijections. (That $(\circ(\varepsilon\cup))$ respects all three is a consequence of $\varepsilon$ being a bijection. That function composition preserves the property of being imp- (respectively co-imp-, bijection-) respecting is easily verified.)
Finally, we have the following properties of the left and right domains of such catamorphisms:

$$R< \ \sqsupseteq \ (\![F,\varepsilon;\ \ R]\!)<$$

$$C \ \sqsupseteq \ (\![F,\varepsilon;\ \ R]\!)>$$

$$(\![F,\varepsilon;\ \ R]\!) \ = \ (\![F,\varepsilon;\ \ R \circ F.R<]\!)$$

$$(\![F,\varepsilon;\ \ R]\!)< \ = \ R< \quad \Leftarrow \quad F.(\![F,\varepsilon;\ \ R]\!)< \ \sqsupseteq \ R>$$

$$(\![F,\varepsilon;\ \ R]\!)> \ = \ C \quad \Leftarrow \quad R> \ \sqsupseteq \ F.R<$$

$$(\![F,\varepsilon;\ \ f]\!) \in A \longleftarrow C \quad \Leftarrow \quad f \in A \longleftarrow F.A$$

All of these properties can be verified by minor editing of the proofs given in sections 13.3 and 13.5. In some cases we have outlined an alternative (and preferable) proof strategy. The complete details are left to the industrious reader.

It remains for us to remark that $(C,\tau)$ is obviously an $F$-algebra, and its initiality is guaranteed by the last property above together with the unique extension property, property (13.41).

### 13.7.3 Initial algebras

We suppose now that $(C, \tau, \eta)$ is an initial $F$-algebra. Our goal is to prove that $C$ is isomorphic to $\mu F$. We intend to achieve this goal by exhibiting $\varepsilon$ and $\gamma$ such that

- $\varepsilon \in C \longleftarrow \mu F$
- $\gamma \in \mu F \longleftarrow C$
- $\gamma = \varepsilon^{\cup}$

Type inference again gives us potential candidates $\varepsilon$ and $\gamma$:

By construction of $\varepsilon$:

$$
\begin{array}{ll}
 & \varepsilon \in C \longleftarrow \mu F \\
\Leftarrow & \quad \{ \bullet \quad \varepsilon = (\![f]\!), \text{ initiality of } (\mu F, \mu F, (\![\_\_]\!)) \quad \} \\
 & f \in C \longleftarrow F.C \\
\Leftarrow & \quad \{ \ (C, \tau) \text{ is an } F\text{-algebra} \ \} \\
 & f = \tau
\end{array}
$$

By construction of $\gamma$:

$$
\begin{array}{ll}
 & \gamma \in \mu F \longleftarrow C \\
\Leftarrow & \quad \{ \bullet \quad \gamma = \eta.f, \text{ initiality of } (C, \tau, \eta) \ \} \\
 & f \in \mu F \longleftarrow F.\mu F \\
\Leftarrow & \quad \{ \ (\mu F, \mu F) \text{ is an } F\text{-algebra} \ \} \\
 & f = \mu F
\end{array}
$$

So choose $\varepsilon = (\![\tau]\!)$ and $\gamma = \eta.\mu F$. Finally we verify that one is the reverse of the other:

$$
\begin{array}{ll}
 & \gamma = \varepsilon^{\cup} \\
\equiv & \quad \{ \ \text{choice of } \varepsilon \text{ and } \gamma \ \} \\
 & \eta.\mu F = (\![\tau]\!)^{\cup} \\
\equiv & \quad \{ \ \text{initiality of } \eta\text{: 13.34(d)} \ \} \\
 & (\![\tau]\!)^{\cup} = \mu F \circ F.(\![\tau]\!)^{\cup} \circ \tau^{\cup} \\
\equiv & \quad \{ \ \text{reverse, } \mu F = \mu F^{\cup} \ \} \\
 & (\![\tau]\!) = \tau \circ F.(\![\tau]\!) \circ \mu F \\
\equiv & \quad \{ \ \text{uep: (13.11)} \ \} \\
 & \mathbf{true}
\end{array}
$$

### 13.7.4 An Application to Isomorphic Relators

Under what circumstances might one construct two isomorphic monotypes? One possibility that suggests itself is when constructing the fixed point of two naturally isomorphic endorelators. That, after all, is how one obtains naturally isomorphic representations of the natural numbers. Suppose $F$ is an endorelator and $\gamma$ is a bijection with $\gamma< = F.I$. Then, the claim is:

**Theorem 13.42**

$$(\![F;\ \gamma\cup]\!) \;=\; (\![F^{\gamma};\ \gamma]\!)\cup \text{ is a bijection to } \mu(F^{\gamma}) \text{ from } \mu F.$$

(See (11.14) in section 11.4 for the definition of $F^{\gamma}$.)

**Proof** Let $G = F^{\gamma}$. We must construct a bijection with right domain $\mu F$ and left domain $\mu G$. To satisfy the first condition an $F$-catamorphism is the obvious thing to construct. To satisfy the second condition the reverse of a $G$-catamorphism is more appropriate. Let us therefore try both. First,

$$\begin{array}{ll}
\multicolumn{2}{l}{\text{by construction of } \alpha} \\
 & (\![F;\ \alpha]\!)> = \mu F \\
\Leftarrow & \qquad \{ \text{ lemma 13.21(b) } \} \\
 & \alpha> \sqsupseteq F.(\![F;\ \alpha]\!)< \\
\Leftarrow & \qquad \{ \bullet \;\; \alpha \;:=\; \gamma\cup,\ \gamma\cup> \;=\; F.I \;\} \\
 & F.I \sqsupseteq F.(\![F;\ \gamma\cup]\!)< \\
\equiv & \qquad \{ \text{ domains are monotypes, monotonicity } \} \\
 & \mathbf{true}
\end{array}$$

Similarly,

$$(\![G;\ \gamma]\!)> = \mu G$$

Since $\gamma$ is a bijection and catamorphisms preserve bijections, both of $(\![F;\ \gamma\cup]\!)$ and $(\![G;\ \gamma]\!)$ are bijections. We now conjecture that

$$(\![F;\ \gamma\cup]\!) = (\![G;\ \gamma]\!)\cup$$

The proof goes as follows:

$$
\begin{array}{ll}
& (\!|F;\ \gamma^{\cup}|\!) = (\!|G;\ \gamma|\!)^{\cup} \\
\equiv & \qquad \{\ \text{uep: (13.11)}\ \} \\
& (\!|G;\ \gamma|\!)^{\cup} \ = \ \gamma^{\cup} \circ F.(\!|G;\ \gamma|\!)^{\cup} \circ \mu F \\
\equiv & \qquad \{\ \text{reverse,}\ \ \mu F \ = \ \mu F^{\cup}\ \} \\
& (\!|G;\ \gamma|\!) \ = \ \mu F \circ F.(\!|G;\ \gamma|\!) \circ \gamma \\
\equiv & \qquad \{\ \gamma \in F {\leftrightsquigarrow} G\ \} \\
& (\!|G;\ \gamma|\!) \ = \ \mu F \circ \gamma \circ G.(\!|G;\ \gamma|\!) \\
\Leftarrow & \qquad \{\ \text{computation rule: 13.4(a)}\ \} \\
& (\!|G;\ \gamma|\!) = (\!|G;\ \mu F \circ \gamma|\!) \\
\Leftarrow & \qquad \{\ \text{13.19(a)}\ \} \\
& \mu F \sqsupseteq (\gamma \circ G.\mu F)< \\
\equiv & \qquad \{\ \gamma \in F {\leftrightsquigarrow} G\ \} \\
& \mu F \sqsupseteq (F.\mu F \circ \gamma)< \\
\equiv & \qquad \{\ \mu F \ = \ F.\mu F \ = \ F.\mu F<,\ \text{domains: (10.17)}\ \} \\
& \mathbf{true}
\end{array}
$$

In conclusion

$$(\!|F;\ \gamma^{\cup}|\!) = (\!|G;\ \gamma|\!)^{\cup}$$

is a bijection to $\mu G$ from $\mu F$.
□

We now seek a relationship between $F$-catamorphisms and $F^{\gamma}$-catamorphisms. Formally we prove:

**Theorem 13.43**

$$(\!|F^{\gamma};\ R\circ\gamma|\!) \ = \ (\!|F;\ R|\!) \circ (\!|F^{\gamma};\ \gamma|\!) \ = \ (\!|F, (\!|F;\ \gamma^{\cup}|\!);\ R|\!)$$

**Proof**

By construction of $S$ and $\tau$:

$$
\begin{array}{ll}
& (\!|F^{\gamma};\ S|\!) \ = \ (\!|F;\ R|\!) \circ \tau \\
\equiv & \qquad \{\ \text{uep: (13.11)},\ F^{\gamma}\ \text{is a relator}\ \} \\
& (\!|F;\ R|\!) \circ \tau \ = \ S \circ F^{\gamma}.(\!|F;\ R|\!) \circ F^{\gamma}.\tau \circ \mu F^{\gamma} \\
\equiv & \qquad \{\ \text{computation rule: 13.4(a)}\ \} \\
& R \circ F.(\!|F;\ R|\!) \circ \tau \ = \ S \circ F^{\gamma}.(\!|F;\ R|\!) \circ F^{\gamma}.\tau \circ \mu F^{\gamma} \\
\Leftarrow & \qquad \{\ \bullet\ \ S := R\circ\gamma,\ \text{aiming to invoke}\ \gamma \in F {\leftrightsquigarrow} F^{\gamma}\ \} \\
& R \circ F.(\!|F;\ R|\!) \circ \tau \ = \ R \circ \gamma \circ F^{\gamma}.(\!|F;\ R|\!) \circ F^{\gamma}.\tau \circ \mu F^{\gamma}
\end{array}
$$

$$
\begin{array}{ll}
\equiv & \quad \{\ \gamma \in F \leftrightsquigarrow F^{\gamma}\ \} \\
& R \circ F.(\![F;\ R]\!) \circ \tau \ = \ R \circ F.(\![F;\ R]\!) \circ \gamma \circ F^{\gamma}.\tau \circ \mu F^{\gamma} \\
\Leftarrow & \quad \{\ \text{Leibniz}\ \} \\
& \tau \ = \ \gamma \circ F^{\gamma}.\tau \circ \mu F^{\gamma} \\
\equiv & \quad \{\ \text{uep: } (13.11)\ \} \\
& \tau = (\![F^{\gamma};\ \gamma]\!)
\end{array}
$$

□

Dual to this we expect

**Theorem 13.44**

$$(\![F;\ R \circ \gamma^{\cup}]\!) \ = \ (\![F^{\gamma};\ R]\!) \circ (\![F;\ \gamma^{\cup}]\!) \ = \ (\![F^{\gamma}, (\![F^{\gamma};\ \gamma]\!);\ R]\!)$$

which we verify as follows:

$$
\begin{array}{ll}
& (\![F^{\gamma};\ R]\!) \\
= & \quad \{\ 13.19(\text{a})\ \} \\
& (\![F^{\gamma};\ R \circ F^{\gamma}.I]\!) \\
= & \quad \{\ \gamma^{\cup} \circ \gamma \ = \ F^{\gamma}.I\ \} \\
& (\![F^{\gamma};\ R \circ \gamma^{\cup} \circ \gamma]\!) \\
= & \quad \{\ \text{theorem 13.43}\ \} \\
& (\![F;\ R \circ \gamma^{\cup}]\!) \circ (\![F^{\gamma};\ \gamma]\!)
\end{array}
$$

Hence

$$
\begin{array}{ll}
& (\![F;\ R \circ \gamma^{\cup}]\!) \\
= & \quad \{\ \mu F^{\gamma} \ = \ (\![F^{\gamma};\ \gamma]\!) \circ (\![F;\ \gamma^{\cup}]\!)\ \} \\
& (\![F;\ R \circ \gamma^{\cup}]\!) \circ (\![F^{\gamma};\ \gamma]\!) \circ (\![F;\ \gamma^{\cup}]\!) \\
= & \quad \{\ \text{above}\ \} \\
& (\![F^{\gamma};\ R]\!) \circ (\![F;\ \gamma^{\cup}]\!)
\end{array}
$$

□

# Chapter 14

# Parameterised Types

## 14.1 New relators from old

The theorems in the earlier sections are all well and good but a major concern is to build new relators from existing ones. The achievement of this goal is delightfully simple. Suppose $\otimes$ is a binary relator. Fix one of its arguments to spec $R$, say, and then consider $\mu(R\otimes)$. Finally, abstracting from $R$ we have constructed a function from specs to specs. In this section we prove the beautiful and remarkable result that this function is a relator.

(Note: Up until now we have used the operator "$\mu$" only in the context of a relator. In general $R\otimes$ is not a relator (although $I\otimes$ is), but it is monotonic and so has a least fixed point.)

Before embarking on the proof let us recall also the defining property of $\mu(R\otimes)$:

(14.1) $\mu(R\otimes) \;=\; R \otimes \mu(R\otimes)$

(14.2) $X \sqsupseteq \mu(R\otimes) \;\Leftarrow\; X \sqsupseteq R \otimes X$

The form of (14.1) and (14.2) is highly reminiscent of the definition of a catamorphism, leading us to the following:

**Definition 14.3**

Suppose $\otimes$ is a binary relator. It is easy to verify that $I\otimes$ is a relator (where $(I\otimes).R = I \otimes R$). Its catamorphisms therefore exist and we may define:

$$\varpi R \quad \hat{=} \quad (\!|I\otimes;\;\; R \otimes I|\!)$$

□

Our initial goal is to show that $\mu(R\otimes) \;=\; \varpi R$.

In the following calculations we adopt the convention that composition has lower precedence than "$\otimes$". We also drop the argument "$I\otimes$" within the catamorphism brackets since our discussion will be confined to just this one relator.

For ease of reference it is useful to instantiate the unique extension property, computation rule and fusion properties of section 13 with $F := I\otimes$ and the definition of $\varpi R$. After some simplification, using in particular the assumed compositionality of $\otimes$, these become:

(Unique extension property)

$$(14.4)\quad X \;=\; \varpi R \;\;\equiv\;\; X \;=\; R\otimes X \circ \mu(I\otimes)$$

(Computation rules)

$$(14.5)\quad (\!|R|\!) \;=\; R \circ I\otimes(\!|R|\!) \;=\; (\!|R|\!) \circ \mu(I\otimes)$$
$$(14.6)\quad \varpi R \;=\; R\otimes\varpi R \;=\; \varpi R \circ \mu(I\otimes)$$

(Fusion laws)

$$(14.7)\quad U \circ (\!|V|\!) \;\trianglelefteq\; (\!|R|\!) \;\;\Leftarrow\;\; U \circ V \;\trianglelefteq\; R \circ I\otimes U$$

$$(14.8)\quad U \circ \varpi V \;\trianglelefteq\; \varpi R \;\;\Leftarrow\;\; U \circ V\otimes I \;\trianglelefteq\; R\otimes U$$

where "$\trianglelefteq$" is any of "$=$", "$\sqsupseteq$", "$\sqsubseteq$".

To achieve our goal the obvious first step is to invoke the unique extension property.

$$\begin{array}{ll}
 & \mu(R\otimes) = \varpi R \\
\equiv & \qquad \{\ \text{uep: (14.4)}\ \} \\
 & \mu(R\otimes) \;=\; R\otimes I \circ I\otimes\mu(R\otimes) \circ \mu(I\otimes) \\
\equiv & \qquad \{\ \text{relator.}\otimes\ \} \\
 & \mu(R\otimes) \;=\; R\otimes\mu(R\otimes) \circ \mu(I\otimes) \\
\equiv & \qquad \{\ \text{induction principle: (14.2)}\ \} \\
 & \mu(R\otimes) \;=\; \mu(R\otimes) \circ \mu(I\otimes)
\end{array}$$

This last equality is established by proving two inclusions. First:

(14.9) $I \;\sqsupseteq\; \mu(I\otimes)$

**Proof** Immediate from the induction principle (14.2) and $I \sqsupseteq I \otimes I$.
□
Thus, by monotonicity:

(14.10) $\mu(R\otimes) \sqsupseteq \mu(R\otimes) \circ \mu(I\otimes)$

For the other inclusion another appeal to the induction principle is required:

$$
\begin{array}{ll}
 & \mu(R\otimes) \circ \mu(I\otimes) \sqsupseteq \mu(R\otimes) \\
\Leftarrow & \qquad \{ \text{ induction principle: (14.2) } \} \\
 & \mu(R\otimes) \circ \mu(I\otimes) \sqsupseteq R \otimes (\mu(R\otimes) \circ \mu(I\otimes)) \\
\equiv & \qquad \{ \text{ relator.}\otimes \} \\
 & \mu(R\otimes) \circ \mu(I\otimes) \sqsupseteq R \otimes \mu(R\otimes) \circ I \otimes \mu(I\otimes) \\
\equiv & \qquad \{ \text{ (14.1) } \} \\
 & \mu(R\otimes) \circ \mu(I\otimes) \sqsupseteq \mu(R\otimes) \circ \mu(I\otimes) \\
\equiv & \\
 & \mathbf{true}
\end{array}
$$

We have thus established

(14.11) $\mu(R\otimes) \sqsubseteq \mu(R\otimes) \circ \mu(I\otimes)$

and the combination of (14.10) and (14.11) completes the proof of:

**Theorem 14.12** $\mu(R\otimes) = \varpi R$

□
It is a straightforward matter to verify that $\varpi$ is a relator. Here are the proofs of the four properties.

**Lemma 14.13**

$$I \;\sqsupseteq\; \varpi I$$

**Proof** Immediate from the conjunction of (14.9) and theorem 14.12.
□

In order to show that $\varpi$ distributes over composition we prove first a more general lemma, called the *map fusion* lemma. The lemma is very useful in its own right because it states that a catamorphism and a map can always be fused into a single catamorphism. Although the lemma is just a special case of the fusion laws in section 13.4 (see corollary 13.14) its significance is that two specs of which the *left* operand is a catamorphism are fused into one catamorphism.

**Lemma 14.14 (Map Fusion)**

$$(\!|R|\!) \circ \varpi S \quad = \quad (\!|R \circ S \otimes I|\!)$$

**Proof**

$$
\begin{array}{ll}
& (\!|R|\!) \circ \varpi S \quad = \quad (\!|R \circ S \otimes I|\!) \\
\Leftarrow & \qquad \{ \text{ fusion — (14.7) and definition 14.3 } \} \\
& (\!|R|\!) \circ S \otimes I \quad = \quad R \circ S \otimes I \circ I \otimes (\!|R|\!) \\
\equiv & \qquad \{ \text{ binary relators abide with composition } \} \\
& (\!|R|\!) \circ S \otimes I \quad = \quad R \circ I \otimes (\!|R|\!) \circ S \otimes I \\
\equiv & \qquad \{ \text{ computation rule, (14.5) } \} \\
& \mathbf{true}
\end{array}
$$

□

**Lemma 14.15**

$$\varpi R \circ \varpi S = \varpi(R \circ S)$$

**Proof**

$$
\begin{array}{ll}
& \varpi R \circ \varpi S \\
= & \qquad \{ \text{ defn. of } \varpi \ \} \\
& (\!|R \otimes I|\!) \circ \varpi S \\
= & \qquad \{ \text{ lemma 14.14 } \} \\
& (\!|R \otimes I \circ S \otimes I|\!) \\
= & \qquad \{ \text{ compositionality of relators } \} \\
& (\!|(R \circ S) \otimes I|\!) \\
= & \qquad \{ \text{ defn. of } \varpi \ \} \\
& \varpi(R \circ S)
\end{array}
$$

□

**Lemma 14.16 (Monotonicity)**

$$\varpi R \sqsupseteq \varpi S \quad \Leftarrow \quad R \sqsupseteq S$$

**Proof**
Immediate from the definition of $\varpi$ and the monotonicity of catamorphisms and relators.
□

**Lemma 14.17 (Revertability)**

$$(\varpi R)\cup \quad = \quad \varpi(R\cup)$$

**Proof**

$$\begin{array}{ll}
 & (\varpi R)\cup \quad = \quad \varpi(R\cup) \\
\equiv & \qquad \{ \text{ uep: } (14.4) \ \} \\
 & (\varpi R)\cup \quad = \quad R\cup \otimes (\varpi R)\cup \circ \mu(I\otimes) \\
\equiv & \qquad \{ \text{ reverse, } \otimes \text{ is a relator and } \mu(I\otimes) \text{ a monotype } \} \\
 & \varpi R \quad = \quad \mu(I\otimes) \circ R \otimes \varpi R \\
\equiv & \qquad \{ \text{ theorem 14.12; computation rule (14.6) } \} \\
 & \varpi R \quad = \quad \varpi I \circ \varpi R \\
\equiv & \qquad \{ \text{ lemma 14.15, } I \text{ is the unit of composition } \} \\
 & \mathbf{true}
\end{array}$$

□

**Theorem 14.18** $\varpi$ is a relator.

**Proof** Lemmas 14.13, 14.15, 14.16 and 14.17.
□

The map fusion law and general fusion law for catamorphisms, specialised appropriately, are conveniently combined into one naturality law.

**Theorem 14.19 (Naturality of Map Relators)**
For all $\sim \ \in \{\leftsquigarrow, \leftrightsquigarrow, \rightsquigarrow\}$

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

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

$$
\begin{array}{ll}
 & (\!|\_|\!) \ \in \ (S \sim \varpi.R) \ \dot{\leftsquigarrow} \ (S \sim R \otimes S) \\
\Leftarrow & \qquad \{ \ (\!|\_|\!) \text{ is a total function: theorem 11.4 } \} \\
 & \forall (U,V :: \ S \circ (\!|U|\!) \quad \trianglelefteq \quad (\!|V|\!) \circ \varpi.R \\
 & \qquad\qquad\qquad \Leftarrow \quad S \circ U \quad \trianglelefteq \quad V \ \circ \ R \otimes S \\
 & \ )
\end{array}
$$

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

$$
\begin{array}{ll}
 & S \circ (\!|U|\!) \quad \trianglelefteq \quad (\!|V|\!) \circ \varpi.R \\
\equiv & \qquad \{ \ \text{map fusion: 14.14} \ \} \\
 & S \circ (\!|U|\!) \quad \trianglelefteq \quad (\!|V \circ R \otimes I|\!) \\
\Leftarrow & \qquad \{ \ \text{catamorphism fusion: 13.14} \ \} \\
 & S \ \circ \ U \quad \trianglelefteq \quad V \ \circ \ R \otimes I \ \circ \ I \otimes S \\
\equiv & \qquad \{ \ \otimes \text{ is a relator } \} \\
 & S \ \circ \ U \quad \trianglelefteq \quad V \ \circ \ R \otimes S \quad .
\end{array}
$$

□

****comment***

**Theorem 14.20**

$$
(\!|A\otimes; R|\!) \ = \ (\!|I\otimes; R|\!) \circ \varpi A
$$

**Proof**

$$
\begin{array}{ll}
 & (\!|A\otimes; R|\!) \ = \ (\!|I\otimes; R|\!) \circ \varpi A \\
\equiv & \qquad \{ \ \text{uep: (13.11)} \ \} \\
 & (\!|I\otimes; R|\!) \circ \varpi A \quad = \quad R \ \circ \ A \ \otimes \ ((\!|I\otimes; R|\!) \circ \varpi A) \ \circ \ \mu(A\otimes) \\
\equiv & \qquad \{ \ \text{computation rule: (14.5); } \otimes \text{ is a relator; (14.12) } \} \\
 & R \ \circ \ I \ \otimes \ (\!|I\otimes; R|\!) \ \circ \ \varpi A \ = \ R \ \circ \ I \ \otimes \ (\!|I\otimes; R|\!) \ \circ \ A \otimes \varpi A \ \circ \ \varpi A \\
\equiv & \qquad \{ \ \text{(14.6)}, \varpi A \ = \ \varpi A \circ \varpi A \ \} \\
 & \textbf{true}
\end{array}
$$

□

## 14.2 Junctivity properties

In [36], chapter 8, Dijkstra and Scholten tread a similar path to our own: they first consider equations in $X$ of the form $X = \theta.X$, for monotonic function $\theta$, and then introduce a parameter $Y$ by supposing that $\theta = Y\oplus$ for some binary operator $\oplus$. (Their notation is, however, different.) By way of two, what they call "beautiful", theorems they establish that the least and greatest solutions of the equation in $X$, $X = Y \oplus X$, are remarkably well behaved with respect to the parameter $Y$. In the present contexts their theorems become one, which we shall call the junctivity theorem.

As remarked elsewhere, the functions $(R\circ)$ and $(\circ R)$ are universally $\sqcup$-junctive for all specs $R$. Moreover, $(f\circ)$ is positively $\sqcap$-junctive for all co-imps $f$, and $(\circ f)$ is positively $\sqcap$-junctive for all imps $f$. These are two of the most important ingredients in the proof that follows.

**Theorem 14.21 (Junctivity)**

(a) If $\otimes$ is $\mathcal{I}$-$\sqcup$-junctive then so too is $\varpi$.
(b) For non-empty $\mathcal{I}$, if $\otimes$ is $\mathcal{I}$-$\sqcap$-junctive then so too is $\varpi$.
(c) If $\otimes$ is $\mathcal{I}$-$\sqcup$-continuous, then so too is $\varpi$.
(d) For non-empty $\mathcal{I}$, if $\otimes$ is $\mathcal{I}$-$\sqcap$-continuous then so too is $\varpi$.

**Proof**
(a) Suppose $\mathcal{R}$ is an $\mathcal{I}$-bag.

$$
\begin{array}{ll}
 & \varpi(\sqcup_{\mathcal{I}}\mathcal{R}) \;=\; \sqcup_{\mathcal{I}}(\varpi \bullet \mathcal{R}) \\
\equiv & \quad \{ \text{ uep } \} \\
 & \sqcup_{\mathcal{I}}(\varpi \bullet \mathcal{R}) \;=\; \sqcup_{\mathcal{I}}\mathcal{R} \otimes \sqcup_{\mathcal{I}}(\varpi \bullet \mathcal{R}) \circ \mu(I\otimes) \\
\Leftarrow & \quad \{ \quad \sqcup_{\mathcal{I}}(\varpi \bullet \mathcal{R}) \\
 & \qquad = \qquad \{ \text{ definition } \} \\
 & \qquad\quad \sqcup(i : i \in \mathcal{I} : \varpi(\mathcal{R}.i)) \\
 & \qquad = \qquad \{ \ (14.6) \ \} \\
 & \qquad\quad \sqcup(i : i \in \mathcal{I} : \varpi(\mathcal{R}.i) \circ \mu(I\otimes)) \\
 & \qquad = \qquad \{ \circ(\mu(I\otimes)) \text{ is universally } \sqcup\text{-junctive } \} \\
 & \qquad\quad \sqcup(i : i \in \mathcal{I} : \varpi(\mathcal{R}.i)) \circ \mu(I\otimes) \\
 & \qquad = \qquad \{ \text{ definition } \} \\
 & \qquad\quad \sqcup_{\mathcal{I}}(\varpi \bullet \mathcal{R}) \circ \mu(I\otimes) \\
 & \quad \} \\
 & \sqcup_{\mathcal{I}}(\varpi \bullet \mathcal{R}) \;=\; \sqcup_{\mathcal{I}}\mathcal{R} \otimes \sqcup_{\mathcal{I}}(\varpi \bullet \mathcal{R}) \\
\equiv & \quad \{ \text{ computation rule } \} \\
 & \sqcup_{\mathcal{I}}(\otimes \bullet \langle \mathcal{R}, \varpi\mathcal{R} \rangle) \;=\; \sqcup_{\mathcal{I}}\mathcal{R} \otimes \sqcup_{\mathcal{I}}(\varpi \bullet \mathcal{R}) \\
\Leftarrow & \quad \{ \text{ definition } \} \\
 & \otimes \text{ is } \mathcal{I}\text{-}\sqcup\text{-junctive}
\end{array}
$$

The proof of part (b) is the exact dual except for the step that appeals to the universal $\sqcup$-junctivity of $\circ(\mu(I\otimes))$. Since $\mu(I\otimes)$ is a monotype, it is an imp; hence (see our preliminary remarks) $\mu(I\otimes)$ is positively $\sqcap$-junctive. The justification of the step may thus be replaced by an appeal to this fact under the assumption that $\mathcal{I}$ is non-empty.

For parts (c) and (d) all we have to remark is that, if $\mathcal{R}$ is linear, monotonicity of $\varpi$ guarantees that the $\mathcal{I}$-bag $\langle \mathcal{R}, \varpi\mathcal{R}\rangle$ is linear too.
□
Interesting consequences of theorem 14.21 are obtained by universally quantifying over all $\mathcal{I}$-bags of a certain type. Examples include the theorem that $\varpi$ is positively $\sqcap$-junctive if $\otimes$ is, and $\varpi$ is ($\sqcup$-or $\sqcap$-) continuous if $\otimes$ is too.

## 14.3 Preservation of Isomorphisms

An obvious and important question to ask is whether the construction of $\varpi$ from binary relator $\otimes$ preserves natural isomorphisms between relators. The answer is, of course, yes!

**Theorem 14.22** Let $\oplus$ and $\otimes$ be binary relators and suppose $\gamma \in \oplus \cong \otimes$. Let $\dagger$ and $\ddagger$ be the relators defined by

$$\begin{array}{rcl} \dagger R & = & (\!|I\oplus;\ R\oplus I|\!) \\ \ddagger R & = & (\!|I\otimes;\ R\otimes I|\!) \end{array}$$

Then $\dagger$ and $\ddagger$ are naturally isomorphic.

**Proof** For binary relators $\oplus$ and $\otimes$ the statement $\gamma \in \oplus \cong \otimes$ means that $\gamma$ is a bijection with

(14.23) $\gamma< \;=\; I\oplus I$

(14.24) $\gamma> \;=\; I\otimes I$

and for all specs $R$ and $S$,

(14.25) $R\oplus S \circ \gamma \;=\; \gamma \circ R\otimes S$

Equation (14.25) is easily shown to be equivalent to the conjunction of

(14.26) $R\oplus I \circ \gamma \;=\; \gamma \circ R\otimes I$

for all specs $R$, and

(14.27) $I \oplus S \circ \gamma \quad = \quad \gamma \circ I \otimes S$

for all specs $S$.

Hence, in combination with (14.23) and (14.24) we have:

$$\gamma \in \oplus \cong \otimes \quad \equiv \quad \gamma \in (I\oplus) \cong (I\otimes) \quad \wedge \ \gamma \in (\oplus I) \cong (\otimes I)$$

That $\gamma \in (I\oplus) \cong (I\otimes)$ means we can invoke theorem 13.44 with $F$ instantiated to $(I\oplus)$ and $G$ to $(I\otimes)$. For brevity let $\overline{\gamma} \ = \ (\!|I\oplus;\ \gamma^{\cup}|\!)$. Then, instantiating theorems 13.42, 13.43 and 13.44 we have the following:

(14.28) $\overline{\gamma}$ is a bijection to $\dagger I$ from $\ddagger I$

(14.29) $\overline{\gamma} \quad = \quad (\!|I\oplus;\ \gamma^{\cup}|\!) \quad = \quad (\!|I\otimes;\ \gamma|\!)^{\cup}$

(14.30) $(\!|I\otimes;\ R \circ \gamma|\!) \quad = \quad (\!|I\oplus;\ R|\!) \circ \overline{\gamma}^{\cup}$

(14.31) $(\!|I\oplus;\ R \circ \gamma^{\cup}|\!) \quad = \quad (\!|I\otimes;\ R|\!) \circ \overline{\gamma}$

Because of (14.28) we conjecture that $\overline{\gamma}$ witnesses an isomorphism between $\dagger$ and $\ddagger$. To verify the conjecture it suffices to show that $\ddagger R \circ \overline{\gamma} \quad = \quad \overline{\gamma} \circ \ \dagger R$ for all $R$. Here goes!

$$
\begin{array}{cl}
 & \ddagger R \circ \overline{\gamma} \\
= & \qquad \{ \ \text{definition of } \ddagger \ \} \\
 & (\!|I\otimes;\ R \otimes I|\!) \circ \overline{\gamma} \\
= & \qquad \{ \ (14.31) \ \} \\
 & (\!|I\oplus;\ R \otimes I \circ \gamma^{\cup}|\!) \\
= & \qquad \{ \ (14.26) \ \} \\
 & (\!|I\oplus;\ \gamma^{\cup} \circ R \oplus I|\!) \\
= & \qquad \{ \ \text{lemma 14.14} \ \} \\
 & (\!|I\oplus;\ \gamma^{\cup}|\!) \circ \ \dagger R \\
= & \qquad \{ \ (14.29) \ \} \\
 & \overline{\gamma} \circ \ \dagger R
\end{array}
$$

□

## 14.4 A Simulation Property

In this section we present what is, at the time of writing, an isolated result but which may prove to be much more significant in the future.

Recall that relator $F$ simulates relator $G$ is denoted by $F \gtrsim G$. Simulation is a preordering on relators and so can easily be extended to a partial ordering. Now, one of the most powerful tools for reasoning about partial orderings is the Knaster-Tarski theorem. The question thus arises whether it is possible to extend the theorem to simulations between relators. This is highly desirable because one can then demonstrate that one relator simulates another without having to explicitly construct the "witness", i.e. the simulation itself, — this is obtained mechanically as a by-product of the extended Knaster-Tarski theorem.

In order to set up such a theorem we first need the notion of a monotonic function on relators. The definition is obvious: a function $\Phi$ is a monotonic function on relators if it maps relators to relators and is such that, for all relators $F$ and $G$,

$$F \gtrsim G \quad \Rightarrow \quad \Phi.F \gtrsim \Phi.G$$

The revised "Knaster-Tarski theorem" would then take the form: if $\Phi$ is a monotonic function from relators to relators then the equation

$$F :: \qquad F \cong \Phi.F$$

has a least solution $\mu\Phi$ with the properties that

$$\begin{array}{ll} & \mu\Phi \cong \Phi.\mu\Phi \\ \text{and} & F \gtrsim \Phi.F \quad \Rightarrow \quad F \gtrsim \mu\Phi \end{array}$$

The proof of the theorem would have to be constructive, otherwise the whole purpose of establishing the theorem (establishing simulations without explicitly constructing the witness) would be lost. It is likely, therefore, that its proof would necessarily be by induction on the prescribed methods for constructing relators. Here we present one lemma in such an inductive proof.

Suppose $\otimes$ is a binary relator and $G$ is a unary relator. Define the function $\Phi$ from relators to relators by, for all specs $R$,

$$\begin{array}{lrcl} & (\Phi.F).R & = & G.R \otimes F.R \\ \text{I.e.} & \Phi.F & = & G \hat{\otimes} F \quad . \end{array}$$

It is easy to verify that if $\Phi$ is monotonic in the sense just defined.

Now the theorem we prove is the following:

$$F \gtrsim G\hat{\otimes}F \quad \Rightarrow \quad F \gtrsim \varpi \bullet G$$

The introduction of the relator $G$ into the statement of the theorem makes it slightly more general than the generalisation of the Knaster-Tarski theorem outlined above but the extra complication proves useful.

Three lemmas lead the way to the theorem's proof.

**Lemma 14.32** $\quad (\!|(G.I)\otimes;\ T|\!) \;=\; (\!|I\otimes;\ T \circ G.I \otimes I|\!) \quad .$

**Proof**

$$\begin{array}{ll} & (\!|(G.I)\otimes;\ T|\!) \\ = & \quad \{ \quad \text{theorem 14.20} \quad \} \\ & (\!|I\otimes;\ T|\!) \circ \varpi.G.I \\ = & \quad \{ \quad \text{map fusion: theorem 14.14} \quad \} \\ & (\!|I\otimes;\ T \circ G.I \otimes I|\!) \quad . \end{array}$$

□

**Lemma 14.33**

$$R> \;=\; G.I \otimes F.I \quad \wedge \quad R< \;\sqsubseteq\; F.I \quad \Rightarrow \quad (\!|I\otimes;\ R|\!)> \;=\; \varpi.G.I \quad .$$

**Proof** Assume $R> \;=\; G.I \otimes F.I \quad \wedge \quad R< \;\sqsubseteq\; F.I$ . Then,

$$\begin{array}{ll} & (\!|I\otimes;\ R|\!) \\ = & \quad \{ \quad R> \;=\; G.I \otimes F.I \text{ , domains} \quad \} \\ & (\!|I\otimes;\ R \circ G.I \otimes I|\!) \\ = & \quad \{ \quad \text{lemma 14.32} \quad \} \\ & (\!|(G.I)\otimes;\ R|\!) \quad . \end{array}$$

Hence,

$$\begin{array}{ll} & (\!|I\otimes;\ R|\!) \;=\; \varpi.G.I \\ \Leftarrow & \quad \{ \quad \text{above, totality: theorem 13.22} \quad \} \\ & R> \;\sqsupseteq\; G.I \otimes R< \\ \Leftarrow & \quad \{ \quad R> \;=\; G.I \otimes F.I \quad \} \\ & F.I \;\sqsupseteq\; R< \quad . \end{array}$$

□

**Corollary 14.34** $\quad R \in F \overset{>}{\sim} G \otimes F \quad \Rightarrow \quad (\![ I\otimes;\ R ]\!)_{>} \ = \ \varpi.G.I \quad .$

□

**Lemma 14.35** For all $\dot{\sim} \ \in \ \{\dot{\leftsquigarrow}, \dot{\leftrightsquigarrow}, \dot{\rightsquigarrow}\}$

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

**Proof**

$$
\begin{array}{ll}
 & (\![ I\otimes;\ R ]\!) \in F \mathrel{\dot{\sim}} \varpi \bullet G \\
\equiv & \qquad \{ \quad \text{definition} \quad \} \\
 & \forall(S ::\ (\![ I\otimes;\ R ]\!) \in F.S \ \leftrightsquigarrow\ \varpi.G.S) \\
\Leftarrow & \qquad \{ \quad \text{naturality of map relators: 14.19} \quad \} \\
 & \forall(S ::\ R \in F.S \ \leftrightsquigarrow\ G.S \otimes F.S) \\
\equiv & \qquad \{ \quad \text{definition} \quad \} \\
 & R \in F \mathrel{\dot{\sim}} G \otimes F
\end{array}
$$

□

**Theorem 14.36**

$$(\![ I\otimes;\ T ]\!) \in F \overset{>}{\sim} \varpi \bullet G \quad \Leftarrow \quad T \circ I \otimes F.I \in F \overset{>}{\sim} G \otimes F \quad .$$

**Proof** By combining lemmas 14.34 and 14.35, with $R$ instantiated to $T \circ I \otimes F.I$, we obtain:

$$(\![ I\otimes;\ T ]\!) \in F \overset{>}{\sim} \varpi \bullet G \quad \Leftarrow \quad T \circ I \otimes F.I \in F \overset{>}{\sim} G \otimes F \quad .$$

But, by a simple application of the domain trading rule for catamorphisms (theorem 13.19),

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

□

**Theorem 14.37**

$$(\![ (G.I)\otimes;\ T ]\!) \in F \overset{>}{\sim} \varpi \bullet G \quad \Leftarrow \quad T \circ G.I \otimes F.I \in F \overset{>}{\sim} G \otimes F \quad .$$

**Proof** Immediate from lemma 14.32 together with theorem 14.36.
□

*Remark*: In connection with our earlier uncertainty about the best definition of "simulates" it is worth pointing out that all theorems so far stated with respect to the current definition are equally valid if $\leftrightsquigarrow$ is replaced by $\leftsquigarrow$ in the definition. *End of Remark*

This is an excellent point at which to conclude this section: such a powerful theorem proved with so little effort!

# Chapter 15

# Complemented Domains and Conditionals

Conditionals (**if-then-else** statements) are a well-established feature of programming languages, and our own theory would be incomplete if they were not included. In this section we show how they are expressed and we explore in some detail their algebraic properties.

## 15.1 Domain Complement

For the purpose of defining conditionals (**if-then-else** statements) it is useful to have a *total* operator that has the properties of a complement operator when restricted to monotypes. We call this operator the *complemented right domain* operator.

We specify the complemented right domain of $R$, denoted $R{>}\!\!\bullet$, by the requirement that it is the greatest monotype $A$ satisfying $\bot\!\!\bot \sqsupseteq R \circ A$. I.e.

$$(15.1) \quad R{>}\!\!\bullet \sqsupseteq A \ \equiv\ \bot\!\!\bot \sqsupseteq R \circ A$$

As always, such a requirement imposes on us the burden of showing that it can indeed be fulfilled. To this end we first observe several expressions equivalent to the right side of equation (15.1). Two of these give a closed form for $R{>}\!\!\bullet$ thus establishing the existence (and uniqueness) of the operator.

**Lemma 15.2** The following are all equivalent:

(a) $\perp\!\!\perp \;\sqsupseteq\; R \circ A$
(b) $\perp\!\!\perp \;\sqsupseteq\; R{>} \circ A$
(c) $I \sqcap \neg(R{>}) \;\sqsupseteq\; A$
(d) $\perp\!\!\perp \;\sqsupseteq\; \top\!\!\top \circ R \circ A$
(e) $R \;\sqsubseteq\; \neg(\top\!\!\top \circ A)$
(f) $\neg(\top\!\!\top \circ R) \;\sqsupseteq\; \top\!\!\top \circ A$
(g) $(\neg(\top\!\!\top \circ R)){>} \;\sqsupseteq\; A$

**Proof** We leave the details to the reader. The equivalence of properties (a), (b), (c) and (d) is a consequence of the properties of domains combined in the case of (c) with simple plat calculus and in the case of (d) with the identity $\top\!\!\top \circ \perp\!\!\perp \;=\; \perp\!\!\perp$. The equivalence of (d), (e) and (f) is a consequence of the middle exchange rule. Finally the equivalence of (f) and (g) is once more a property of domains.
□
From the equivalence of (a), (c) and (g) we infer

(15.3) $R{\succ\!\!\bullet} \;=\; I \sqcap \neg(R{>}) \;=\; (\neg(\top\!\!\top \circ R)){>}$ .

The latter two formulae are clumsy; exhibiting them serves the purpose of showing that $R{\succ\!\!\bullet}$ does exist. Both are formulae that are suggested by the intended interpretation of the complemented right domain and might have been proposed as definitions. We prefer, however, the form of (15.1) on the grounds that it is closer to our view of a specification and is easier to calculate with.

The steps used to reach (c) and (g) suggest several properties, specifically:

**Lemma 15.4**

(a) $R{>} \sqcup R{\succ\!\!\bullet} \;=\; I$ and $R{>} \sqcap R{\succ\!\!\bullet} \;=\; \perp\!\!\perp$
(b) $R{\succ\!\!\bullet}{\succ\!\!\bullet} \;=\; R{>}$
(c) $R{\succ\!\!\bullet} \;=\; R{\succ\!\!\bullet}{>}$
(d) $R{\succ\!\!\bullet} \;=\; R{>}{\succ\!\!\bullet} \;=\; (\top\!\!\top \circ R){\succ\!\!\bullet}$
(e) $R{\succ\!\!\bullet} \trianglelefteq S{>} \;\equiv\; S{\succ\!\!\bullet} \trianglelefteq R{>}$ for $\trianglelefteq \in \{\sqsubseteq, =, \sqsupseteq\}$.

**Proof** Part (a) follows from $R{\succ\!\!\bullet} = I \sqcap \neg(R{>})$ and simple plat calculus, as does (b). Part (c) follows from the specification of $R{\succ\!\!\bullet}$ (in particular that it is a monotype). Part (d) follows from the equivalence of (a), (b) and (d) in lemma 15.2. From the symmetry of (e) in $R$ and $S$ it suffices to establish just the case that $\trianglelefteq$ is $\sqsupseteq$. For this case we have:

$$
\begin{array}{ll}
 & R{\succ\!\bullet} \sqsupseteq S{>} \\
\equiv & \quad \{ \ (15.1) \text{ and lemma 15.2(b)} \ \} \\
 & \perp\!\!\perp \ \sqsupseteq \ R{>} \circ S{>} \\
\equiv & \quad \{ \ \text{monotypes commute} \ \} \\
 & \perp\!\!\perp \ \sqsupseteq \ S{>} \circ R{>} \\
\equiv & \quad \{ \ (15.1) \text{ and lemma 15.2(b)} \ \} \\
 & S{\succ\!\bullet} \sqsupseteq R{>}
\end{array}
$$

(This little proof illustrates beautifully our preference for (15.1) as the definition of the complemented right domain.)

□

The importance of 15.4(c) has to do with the fact that we have defined a *total* complement operator. One is tempted to make do with the complement operator in the monotype lattice — for monotype $A$ its complement is $I \sqcap \neg A$ — or in the lattice of right (or left) conditions — for right condition $p$ its complement $\neg p$ in the spec lattice coincides with its complement in the lattice of right conditions. However this creates a dilemma as to which to choose, a dilemma which it is better to circumvent. Lemma 15.4(c) indicates that the choice is irrelevant. (We return to this matter when we introduce to the definition of conditionals.)

The equivalence of (a) and (e) in lemma 15.2 together with the specification (15.1) of the complemented domain operator predict that the complemented domain operator is one adjoint of a Galois connection. It follows that the complemented domain operator is universally $\sqcup$-junctive. To be precise we have:

**Theorem 15.5** For all sets of specs $\mathcal{V}$,

(a) $\quad (\sqcup\mathcal{V}){\succ\!\bullet} \;=\; \sqcap_{\mathcal{M}}\,(\mathcal{V}{\succ\!\bullet})$

where $\sqcap_{\mathcal{M}}$ denotes the infimum operator in the lattice of monotypes. (I.e. $\sqcap_{\mathcal{M}}\mathcal{B} \;=\; I$ when set of monotypes $\mathcal{B}$ is empty, otherwise $\sqcap_{\mathcal{M}}\mathcal{B} \;=\; \sqcap\,\mathcal{B}$.)
In particular, for all specs $R$ and $S$,

(b) $\quad (R \sqcup S){\succ\!\bullet} \;=\; R{\succ\!\bullet} \sqcap S{\succ\!\bullet}$

□

In contrast, but not unexpectedly, the complemented domain operator is not universally $\sqcap$-junctive. Its $\sqcap$-junctivity properties are inextricably linked, however, to those of the normal domain operator.

**Theorem 15.6** For all sets of specs $\mathcal{V}$,

$$\text{(a)} \qquad (\sqcap\mathcal{V}){>\!\!\bullet} \;=\; \sqcup(\mathcal{V}{>\!\!\bullet}) \qquad \equiv \qquad (\sqcap\mathcal{V}){>} \;=\; \sqcap_{\mathcal{M}}(\mathcal{V}{>})$$

In particular, for all specs $R$ and $S$,

$$\text{(b)} \qquad (R \sqcap S){>\!\!\bullet} \;=\; R{>\!\!\bullet} \sqcup S{>\!\!\bullet} \qquad \equiv \qquad (R \sqcap S){>} \;=\; R{>} \sqcap S{>}$$

(Note that the right side of (b) is true if $R$ and $S$ are both monotypes or both right conditions. These are two situations in which the lemma proves useful.)

**Proof**

$$\begin{array}{ll}
 & (\sqcap\mathcal{V}){>\!\!\bullet} \;=\; \sqcup(\mathcal{V}{>\!\!\bullet}) \\
\equiv & \qquad \{ \;\; \sqcup(\mathcal{V}{>\!\!\bullet}) \;=\; (\sqcup(\mathcal{V}{>\!\!\bullet})){>}, \;\; \text{lemma 15.4(e)} \;\; \} \\
 & (\sqcap\mathcal{V}){>} \;=\; (\sqcup(\mathcal{V}{>\!\!\bullet})){>\!\!\bullet} \\
\equiv & \qquad \{ \;\; \text{corollary 15.5} \;\; \} \\
 & (\sqcap\mathcal{V}){>} \;=\; \sqcap_{\mathcal{M}}(\mathcal{V}{>\!\!\bullet}{>\!\!\bullet}) \\
\equiv & \qquad \{ \;\; \text{lemma 15.4(b)} \;\; \} \\
 & (\sqcap\mathcal{V}){>} \;=\; \sqcap_{\mathcal{M}}(\mathcal{V}{>})
\end{array}$$

□

We now turn our attention to the behaviour of the operator with respect to relators. Idealistically it would commute with them (like the ordinary domain operators) but we are out of luck. Nevertheless the next lemma proves to be good enough in most cases.

**Lemma 15.7** If relator $F$ is $\sqcup$-junctive and strict then, for all specs $R$,

$$F.I \circ (F.R){>\!\!\bullet} \;=\; F.(R{>\!\!\bullet})$$

**Proof**

$$\begin{array}{ll}
 & F.I \circ (F.R){>\!\!\bullet} \\
= & \qquad \{ \;\; \text{lemma 15.4(a)} \;\; \} \\
 & F.(R{>} \sqcup R{>\!\!\bullet}) \;\circ\; (F.R){>\!\!\bullet} \\
= & \qquad \{ \;\; \bullet \quad F \text{ is } \sqcup\text{-junctive} \;\; \} \\
 & (F.R{>} \;\sqcup\; F.(R{>\!\!\bullet})) \;\circ\; (F.R){>\!\!\bullet} \\
= & \qquad \{ \;\; \text{lemma 15.4(a) with } R \;:=\; F.R \;\; \}
\end{array}$$

$$
\begin{array}{ll}
 & F.(R\succ\!\bullet) \;\circ\; (F.R)\succ\!\bullet \\
= & \quad \{ \;\bullet\quad \perp\!\!\perp = F.\perp\!\!\perp,\ \text{lemma 15.4(a)} \;\} \\
 & F.(R\succ\!\bullet) \;\circ\; F.R\succ \;\sqcup\; F.(R\succ\!\bullet) \;\circ\; (F.R)\succ\!\bullet \\
= & \quad \{ \;\text{distributivity} \;\} \\
 & F.(R\succ\!\bullet) \;\circ\; (F.R\succ \;\sqcup\; (F.R)\succ\!\bullet) \\
= & \quad \{ \;\text{lemma 15.4(a) with } R \;:=\; F.R \;\} \\
 & F.(R\succ\!\bullet)
\end{array}
$$

□

## 15.2 Domain Translation

We now come to the first of several *translation* rules.

**Lemma 15.8 (Domain Translation)** For all specs $R$ and imps $f$, we have:

$$R\succ \;\circ\; f \;\;=\;\; f \;\circ\; (R \circ f)\succ$$

**Proof**

$$
\begin{array}{ll}
 & R\succ \;\circ\; f \\
= & \quad \{ \;\text{domains: (10.20)} \;\} \\
 & (I \;\sqcap\; \top\!\!\top \;\circ\; R) \;\circ\; f \\
= & \quad \{ \;\bullet\quad \text{imp}.f \;\} \\
 & f \;\sqcap\; \top\!\!\top \;\circ\; R \;\circ\; f \\
= & \quad \{ \;\text{domains: (10.20)} \;\} \\
 & f \;\circ\; (R \;\circ\; f)\succ
\end{array}
$$

□

The above domain translation rule is the embryonic form of the so-called "range translation rule" in the quantifier calculus [3]. The rule provides a mechanism for translating a restriction ($R\succ$) on the left domain of imp $f$ into a restriction ( $(R \circ f)\succ$ ) on its right domain.

Our next goal is to show that there is also a translation rule for the complemented domain operator. Three lemmas are necessary.

**Lemma 15.9** For all specs $R$ and imps $f$,

$$R\succ\!\bullet \;\circ\; f \;\;=\;\; f \;\circ\; (R\succ\!\bullet \;\circ\; f)\succ$$

**Proof**

$$
\begin{array}{ll}
 & R{>}{\bullet} \circ f \\
= & \quad \{ \text{ lemma 15.4(c) } \} \\
 & R{>}{\bullet}{>} \circ f \\
= & \quad \{ \text{ domain translation: lemma 15.8 } \} \\
 & f \circ (R{>}{\bullet} \circ f){>}
\end{array}
$$

□

**Lemma 15.10**

$$S \circ (R{>}{\bullet} \circ S){>} \;\sqsupseteq\; S \circ (R \circ S){>}{\bullet}$$

**Proof**

$$
\begin{array}{ll}
 & S \circ (R{>}{\bullet} \circ S){>} \sqsupseteq S \circ (R{>} \circ S){>}{\bullet} \\
\Leftarrow & \quad \{ \text{ monotonicity, domains: } S \circ S{>} \;=\; S \ \} \\
 & (R{>}{\bullet} \circ S){>} \sqsupseteq S{>} \circ (R{>} \circ S){>}{\bullet} \\
\equiv & \quad \{ \ A \circ B \;=\; A \sqcap B \text{, lemma 15.4(a) } \} \\
 & (R{>}{\bullet} \circ S){>} \sqcup (R{>} \circ S){>} \sqsupseteq S{>} \\
\equiv & \quad \{ \text{ domains: (10.8) } \} \\
 & ((R{>}{\bullet} \sqcup R{>}) \circ S){>} \sqsupseteq S{>} \\
\equiv & \quad \{ \text{ lemma 15.4(a) } \} \\
 & \mathit{true}
\end{array}
$$

□

**Lemma 15.11** For all specs $R$ and imps $f$

$$(R{>}{\bullet} \circ f){>} \;\sqsubseteq\; (R \circ f){>}{\bullet}$$

**Proof**

$$
\begin{array}{ll}
 & (R{>}{\bullet} \circ f){>} \sqsubseteq (R{>} \circ f){>}{\bullet} \\
\equiv & \quad \{ \text{ definition complemented domains (15.1) } \} \\
 & R{>} \circ f \circ (R{>}{\bullet} \circ f){>} \sqsubseteq \perp\!\!\perp \\
\equiv & \quad \{ \text{ lemma 15.9 } \} \\
 & R{>} \circ R{>}{\bullet} \circ f \sqsubseteq \perp\!\!\perp \\
\equiv & \quad \{ \text{ lemma 15.4(a) } \} \\
 & \mathit{true}
\end{array}
$$

□

**Corollary 15.12 (Complemented-Domain Translation)** For all specs $R$ and imps $f$

$$R{\succ\!\bullet} \circ f \;=\; f \circ (R \circ f){\succ\!\bullet}$$

**Proof** By mutual inclusion. The combination of lemmas 15.9 and 15.10 gives one inclusion. Lemma 15.9 combined with 15.11 gives the other.
$\Box$

## 15.3 Conditionals

Several publications have already appeared documenting the algebraic properties of conditionals, the most comprehensive account that we know of being by Hoare *et al* [39]. We shall therefore compare the rules given here with the list that they supply. Their notation for conditionals will also be used, its vital characteristic being that it promotes the Boolean condition to an *infix* operator. Some of the rules presented here were included in Backus's [11] Turing award lecture but his account is less comprehensive and spoiled by the choice of the multifix notation used in the language Lisp.

We take the liberty of omitting most proofs about conditionals on the grounds that the properties are (or should be) unsurprising and their proofs involve only the plat calculus plus a few extra rules to be stated (and proven) shortly. (Some less straightforward proofs are given nonetheless.)

**Definition 15.13 (Conditional)** For all specs $P$ we define the binary operator $\lhd P \rhd$ by:

$$R {\lhd P \rhd} S \;=\; R \circ P{>} \;\sqcup\; S \circ P{\succ\!\bullet}$$
$\Box$

The conditional $R {\lhd P \rhd} S$ can be viewed as a spec which applies $R$ to those elements for which condition $P$ holds and applies $S$ to the other ones.

Note that conditionals are defined for *all* specs but that for all specs $P$, $R$ and $S$,

$$R {\lhd P \rhd} S \;=\; R {\lhd (P{>}) \rhd} S \;=\; R {\lhd (\top\!\!\top \circ P) \rhd} S \quad .$$

Totality of operators is something we strive for at all times: the alternative in this case would have been to restrict $P$ either to monotypes or to right conditions. Had we done so then we would have imposed on ourselves the obligation to determine for every other operator in the calculus whether it preserves monotypes and/or right conditions. In the cases that that is not so the laws relating those operators to conditionals would inevitably have taken on much clumsier forms.

Guards are usually formed by composing primitive guards with the boolean operators. We apply the same design principle to the definition of the booleans: we seek definitions that are total on all specs but are indifferent to the choice of monotypes or right conditions as representations of sets. This leads to the following definition.

**Definition 15.14 (Boolean Operators)** The operators $\bigvee$, $\bigwedge$ and $\sim$, and constants *true* and *false* are defined by, for all sets of specs $\mathcal{P}$ and specs $R$,

$$\begin{array}{lrcl}
\text{(a)} & \bigvee\mathcal{P} & = & (\sqcup\mathcal{P}){>} \\
\text{(b)} & \bigwedge\mathcal{P} & = & \sqcap_{\mathcal{M}}(\mathcal{P}{>}) \\
\text{(c)} & {\sim}R & = & R{>\!\bullet} \\
\text{(d)} & \mathit{true} & = & I \\
\text{(e)} & \mathit{false} & = & \bot\!\!\bot
\end{array}$$

□

In the last section we saw two translation rules, one for the right domain operator, one for the complemented-right-domain operator. Combining these with the fact that imps distribute over both cup and cap we obtain:

**Theorem 15.15 (Predicate Translation)** For all specs $R$, imps $f$ and sets (possibly empty) of specs $\mathcal{P}$, we have:

$$\begin{array}{lrcl}
\text{(a)} & \bigvee\mathcal{P} \circ f & = & f \circ \bigvee(\mathcal{P}\circ f) \\
\text{(b)} & \bigwedge\mathcal{P} \circ f & = & f \circ \bigwedge(\mathcal{P}\circ f) \\
\text{(c)} & {\sim}R \circ f & = & f \circ {\sim}(R\circ f)
\end{array}$$

Hence, for all propositional functions $\theta$ (i.e. functions from specs to specs built from the identity function, constant functions and the boolean operators $\wedge$, $\vee$, $\sim$) and all vectors of specs $\underline{P}$ of the appropriate arity,

$$\begin{array}{lrcl}
\text{(d)} & \theta.\underline{P} \circ f & = & f \circ \theta.(\underline{P}\circ f)
\end{array}$$

□

**Theorem 15.16** The binary operator $\lhd P \rhd$ respects imps. I.e.

$$\text{imp}.(f \lhd P \rhd g) \quad \Leftarrow \quad \text{imp}.f \ \wedge \ \text{imp}.g$$

**Proof**

$$
\begin{array}{ll}
 & \text{imp}.(f \lhd P \rhd g) \\
\equiv & \qquad \{ \ \text{definition 10.28(a)} \ \} \\
 & I \ \sqsupseteq \ f \lhd P \rhd g \ \circ \ (f \lhd P \rhd g)\cup \\
\equiv & \qquad \{ \ \text{definition 15.13, properties of reverse} \ \} \\
 & I \ \sqsupseteq \ (f \circ P{>} \ \sqcup \ g \circ P{\succ\!\bullet}) \ \circ \ (P{>} \circ f\cup \ \sqcup \ P{\succ\!\bullet} \circ g\cup) \\
\equiv & \qquad \{ \ \circ \text{ distributes over } \sqcup, \ \ P{>} \circ P{\succ\!\bullet} \ = \ \perp\!\!\perp \ \} \\
 & I \ \sqsupseteq \ f \circ P{>} \circ f\cup \ \sqcup \ g \circ P{\succ\!\bullet} \circ g\cup \\
\Leftarrow & \qquad \{ \ \text{right domains are monotypes, monotonicity} \ \} \\
 & I \ \sqsupseteq \ f \circ f\cup \ \wedge \ I \ \sqsupseteq \ g \circ g\cup \\
\equiv & \qquad \{ \ \text{definition 10.28(a)} \ \} \\
 & \text{imp}.f \ \wedge \ \text{imp}.g
\end{array}
$$

□

Theorem 15.16 corresponds to the theorem

$$x \ := \ E \lhd P \rhd F \quad = \quad (x \ := \ E) \ \lhd P \rhd \ (x \ := \ F)$$

in the set of properties listed by Hoare *et al* [39]. For them the most primitive implementation (thus, "imp") is an assignment and the content of their rule is that a conditional respects assignments. Their rule is thus at a lower level of abstraction than ours, and more detailed.

The theorem illustrates the sort of proof burden one encounters when type restrictions are imposed on laws. We are obliged to document this theorem because, for example, all the translation rules are restricted to translation by imps. Should we ever wish to translate a domain (say) via a conditional then we need to know in advance that the conditional is an imp.

One final lemma is necessary before we can list the laws obeyed by conditionals.

**Lemma 15.17**

(a) $(R \lhd P \rhd S){>} = R{>} \lhd P \rhd S{>}$

(b) $(R \lhd P \rhd S){\succ\!\bullet} = R{\succ\!\bullet} \lhd P \rhd S{\succ\!\bullet}$

**Proof** Part (a) is easily proved using the definition of conditionals. For (b) we have

$$
\begin{array}{cl}
 & (R\lhd P\rhd S){>\!\!\bullet} \\
= & \qquad \{ \text{ lemma 15.4(d), (a) } \} \\
 & ((R{>} \sqcap P{>}) \sqcup (S{>} \sqcap P{>\!\!\bullet})){>\!\!\bullet} \\
= & \qquad \{ \text{ theorem 15.5(a) } \} \\
 & (R{>} \sqcap P{>}){>\!\!\bullet} \sqcap (S{>} \sqcap P{>\!\!\bullet}){>\!\!\bullet} \\
= & \qquad \{ \text{ theorem 15.6, monotypes } \} \\
 & (R{>\!\!\bullet} \sqcup P{>\!\!\bullet}) \sqcap (S{>\!\!\bullet} \sqcup P{>}) \\
= & \qquad \{ \text{ calculus, lemma 15.4(a) } \} \\
 & (R{>\!\!\bullet} \sqcap S{>\!\!\bullet}) \sqcup (R{>\!\!\bullet} \sqcap P{>}) \sqcup (S{>\!\!\bullet} \sqcap P{>\!\!\bullet}) \\
= & \qquad \{ \; A \sqcap B \sqsubseteq (A \sqcap P{>}) \sqcup (B \sqcap P{>\!\!\bullet}) \; \} \\
 & (R{>\!\!\bullet} \sqcap P{>}) \sqcup (S{>\!\!\bullet} \sqcap P{>\!\!\bullet}) \\
= & \qquad \{ \; R{>\!\!\bullet},\ S{>\!\!\bullet} \text{ are monotypes, definition conditionals } \} \\
 & R{>\!\!\bullet}\lhd P\rhd S{>\!\!\bullet}
\end{array}
$$

□

The set of "unsurprising" laws that we announced earlier can now be given:

**Theorem 15.18** For all specs $P, Q, R, S, T$, imps $f$, and non-empty sets of specs $\mathcal{V}$:

| | |
|---|---|
| (a) | $R\lhd \mathit{true}\rhd S \;=\; R$ |
| (b) | $R\lhd \mathit{false}\rhd S \;=\; S$ |
| (c) | $R\lhd P\rhd R \;=\; R$ |
| (d) | $R\lhd {\sim}P\rhd S \;=\; S\lhd P\rhd R$ |
| (e) | $R\lhd P\rhd (S\lhd P\rhd T) \;=\; R\lhd P\rhd T \;=\; (R\lhd P\rhd S)\lhd P\rhd T$ |
| (f) | $R\lhd (P\wedge Q)\rhd S \;=\; (R\lhd P\rhd S)\lhd Q\rhd S$ |
| (g) | $R\lhd (P\vee Q)\rhd S \;=\; R\lhd P\rhd (R\lhd Q\rhd S)$ |
| (h) | $(\sqcup\mathcal{V})\lhd P\rhd S \;=\; \sqcup(\mathcal{V}\lhd P\rhd S)$ |
| (i) | $(\sqcap\mathcal{V})\lhd P\rhd S \;=\; \sqcap(\mathcal{V}\lhd P\rhd S)$ |
| (j) | $S\lhd (P\lhd Q\rhd R)\rhd T \;=\; (S\lhd P\rhd T)\lhd Q\rhd (S\lhd R\rhd T)$ |
| (k) | $(R\lhd P\rhd S) \sqcup T \;=\; (R\sqcup T)\lhd P\rhd (S\sqcup T)$ |
| (l) | $(R\lhd P\rhd S) \sqcap T \;=\; (R\sqcap T)\lhd P\rhd (S\sqcap T)$ |
| (m) | $(R\lhd P\rhd S) \lhd Q\rhd T \;=\; (R\lhd Q\rhd T)\lhd P\rhd (S\lhd Q\rhd T)$ |
| (n) | $T \circ R\lhd P\rhd S \;=\; (T\circ R)\lhd P\rhd (T\circ S)$ |
| (o) | $R\lhd P\rhd S \circ f \;=\; (R\circ f)\lhd (P\circ f)\rhd (S\circ f)$ |

Moreover, for all propositional functions $\theta$ and all vectors of specs $\underline{P}$ of the appropriate arity,

(p) $R \lhd \theta.\underline{P} \rhd S \circ f = (R \circ f) \lhd \theta.(\underline{P} \circ f) \rhd (S \circ f)$

$\Box$

Little needs to be said about properties (a) through (g) except perhaps to note that (e) asserts that the binary operator $\lhd P \rhd$ is associative. Properties (h) and (i) assert that the function $(X \mapsto X \lhd P \rhd S)$ is positively $\sqcup$- and $\sqcap$-junctive. This is more general than the rules stated by Hoare *et al.* (They claimed only finite, positive $\sqcup$- and $\sqcap$-junctivity.) Property (j) is equivalent to the combination of both parts of lemma 15.17. It is used to construct canonical forms of conditionals (see [51]) but otherwise has marginal value.

Properties (k), (l) and (m) are all distributivity properties of the form

$$\theta.(R \lhd P \rhd S) \quad = \quad (\theta.R) \lhd P \rhd (\theta.S)$$

for some function $\theta$. The function $\theta$ has moreover the form $(X \mapsto X \oslash T)$ for some binary operator $\oslash$. Each rule has a dual whereby $\theta$ is replaced by the function $(X \mapsto T \oslash X)$. These duals have not been listed because they can all be deduced from a combination of the properties (k), (l) and (m) and properties already given. Thus the duals of (k) and (l) follow because $\sqcup$ and $\sqcap$ are both symmetric. The dual of (m) follows from (d) and the fact that $P{\succ\!\!\bullet}{\succ\!\!\bullet} = P{>}$. Property (o) is also a distributivity property of the same form; its dual is obtained by replacing the assumption that $f$ is an imp with the assumption that $f$ is a co-imp and reversing all compositions.

As forewarned we omit all proofs — with one exception. We prove parts (o) and (p) in order to explain why we gave lemmas 15.8 and 15.15.

**Proof** of (o).

$$
\begin{array}{ll}
 & R \lhd P \rhd S \;\circ\; f \\
= & \quad \{ \text{ definition (15.13) } \} \\
 & (R \circ P{>} \sqcup S \circ P{\succ\!\!\bullet}) \;\circ\; f \\
= & \quad \{ \circ \text{ distributes over } \sqcup \} \\
 & R \circ P{>} \circ f \;\sqcup\; S \circ P{\succ\!\!\bullet} \circ f \\
= & \quad \{ \text{ lemma 15.8, corollary 15.12 } \} \\
 & R \circ f \circ (P \circ f){>} \;\sqcup\; S \circ f \circ (P \circ f){\succ\!\!\bullet} \\
= & \quad \{ \text{ definition (15.13) } \} \\
 & (R \circ f) \lhd (P \circ f) \rhd (S \circ f)
\end{array}
$$

Part (p) is proved in the same way: replace $P$ everywhere by $\theta.\underline{p}$ and apply 15.15(d) instead of corollary 15.12.
□

Part (p) is the *translation rule for conditionals*. Given a spec $R \lhd P \rhd S$ with right domain $A$ and an imp $f \in A \longleftarrow B$ one may always translate it to a spec with right domain (at most) $B$ by translating the condition at the level of its primitive components. It takes the place of the law

$$\begin{array}{ll} & (x \ := \ E) \ ; \ (R \lhd P(x) \rhd S) \\ = & ((x \ := \ E) \ ; \ R) \ \lhd P(E) \rhd \ ((x \ := \ E) \ ; \ S) \end{array}$$

in the paper by Hoare *et al* [39]. Parts (n) and (o) of the theorem are also well-documented in the form that we have given here, for example by Backus [11] and Meertens [68] (— at least up to the level of imps in the case of part (n)).

A glaring omission — in the present context — in theorem 15.18 is any mention of relators or catamorphisms. A partial remedy is provided by the next theorem.

**Theorem 15.19** For all specs $P$, $R$ and $S$ and all strict, $\sqcup$-junctive relators $F$,

(a) $\quad F.(R \lhd P \rhd S) \ = \ (F.R) \lhd (F.P) \rhd (F.S)$

In particular, for all specs $P$, $Q$, $R$, $S$, $T$ and $U$,

(b) $\quad (R{+}S) \lhd (P{+}Q) \rhd (T{+}U)$
$\quad = \ (R \lhd P \rhd T) + (S \lhd Q \rhd U)$

Furthermore, for all specs $P$, $R$, $S$ and $T$,

(c) $\quad (R \lhd P \rhd S) \ \vartriangle \ T \ = \ (R \vartriangle T) \lhd P \rhd (S \vartriangle T)$
□

We leave the proof of this theorem as an exercise for the reader. In the case of part (a) the relevant lemma is lemma 15.7. Part (b) is a special case because disjoint sum is universally $\sqcup$-junctive, which is rather more than is required to apply (a). Part (c) involves a simple expansion of the definition of $\vartriangle$ and the application of theorem 15.18(l).

Just as several of the distributivity properties listed in theorem 15.18 had a dual, part (c) has a dual in which the conditional is the righthand argument of the split. The dual follows from (c) by precomposing both sides with the natural isomorphism $\alpha_5$ between the relators $(R, S \mapsto R \times S)$ and $(R, S \mapsto S \times R)$ and applying (12.101).

(Theorem 15.19 was not included by Hoare *et al* because their investigation did not extend to type structures.)

Other properties of conditionals have been omitted where they can be derived by combining elements of theorems 15.18 and 15.19. For example, the reader may wish to verify that the binary operators $\lhd P \rhd$ abide with each other and with $\sqcup$, $\sqcap$ and $\vartriangle$. Another interesting property that can be proved in a few steps with the toolkit now present is, for all specs $P$, $Q$, $R$, $S$, $T$ and $U$,

$$(15.20)\quad (R \lhd P \rhd S) \;\triangledown\; (T \lhd Q \rhd U) \quad = \quad (R \triangledown T) \lhd (P \triangledown Q) \rhd (S \triangledown U)$$

Try it and see!

# Chapter 16

# A Hierarchy of Freebies

## 16.1 The Bird-Meertens Formalism

One of the hardest tasks faced by the theoretician is the assessment of the practicality of one's work. The task is not made any easier by the immense breadth of programming problems to which any useful programming calculus should be applicable. The traditional apology for such an assessment is the presentation of a few, inevitably worn and tired, case studies. We shall not follow such a course.

The course we do follow is to pass the buck: we ask the reader not to assess the practicality of our theory but to assess the practicality of the so-called "Bird-Meertens formalism", and to combine that assessment with an evaluation of the way the formalism is rendered within our theory.

The "Bird-Meertens formalism" (to be more precise, our own conception of it) is a calculus of total functions based on a small number of primitives and a hierarchy of types including trees and lists. The theory was set out in an inspiring paper by Meertens [68] and has been further refined and applied in a number of papers by Bird and Meertens [16, 17, 20, 18, 21]. Its beauty derives from the small scale of the theory itself compared with the large scale of applications.

Essentially there are just three primitive operators in the theory - "reduce", "map" and "filter". (Actually, the names used by Meertens for the first two of these operators were "inserted-in" and "applied-to-all". Moreover, just the first two are primitive since filter is defined in terms of reduce and map.) These operators are defined at each level of a hierarchy of types called the "Boom hierarchy" [1] after H.J. Boom to whom Meertens attributes the concept.

The basis of this hierarchy is given by what Meertens calls "$D$-structures". A $D$-structure, for given type $D$, is formed in one of two ways: there is an embedding function that maps an element of $D$ into a $D$-structure, and there is a binary join operation that combines two $D$-structures into one. Thus, a $D$-structure is a full binary tree with elements of $D$ at the leaves. (By "full" we mean that every interior node has exactly two children.) The embedding function and the join operation are called the *constructors* of the type. Other types in the hierarchy are obtained by adding extra algebraic structure. Trees — binary but non-full — are obtained by assuming that the base type $D$ contains a designated **nil** element which is a left and right unit of the join operation. Lists, bags and sets are obtained by successively introducing the requirements that join is associative, symmetric and idempotent.

Meertens describes the $D$-structures as "about the poorest (i.e., in algebraic laws) possible algebra" and trees as "about the poorest-but-one possible algebra". Nevertheless, in this section we exploit the power of abstraction afforded by the notion of a relator to add several more levels to the Boom hierarchy each of which is "poorer" than those considered by Meertens. Each level is characterised by a class of relators that specialises the class at the level below it. In decreasing order of abstraction these are the "sum" relators, "grounded" and "polymorphically grounded" relators, "monadic" relators and "pointed" relators. ("Grounded" and "polymorphically grounded" relators are formally indistinguishable but it helps to introduce an artificial distinction for a first introduction.) The reason for introducing these extra levels is organisational: the goal is to pin down as clearly as possible the minimum algebraic structure necessary to be able to, first, define the three operators of the Bird-Meertens formalism and, second, establish each of the basic properties of the operators. The conciseness and systematic nature of the development about to be presented, and the fact that it can be conducted at a level yet poorer than "the poorest possible algebra" is for us the most satisfying aspect of this work.

The unconventional nature (and perhaps also the conciseness) of the notations used in the Bird-Meertens formalism makes the formalism difficult to

[1] For the record: Doaitse Swierstra appears to have been responsible for coining the name "Bird-Meertens Formalism" when he cracked a joke comparing "BMF" to "BNF" — Backus-Naur Form — at a workshop in Nijmegen in April, 1988. The name "Boom hierarchy" was suggested to Roland Backhouse by Richard Bird at the same workshop.

comprehend for many groups. The program calculations carried out within the formalism are, however, strongly related to calculations within other systems. In particular there is a strong link between a certain combination of the three basic operators of the formalism and the quantifier expressions used for many years in the Eindhoven school of program development, this link being expressed via a correspondence between the basic laws of the two systems. For the benefit of those familiar with the Eindhoven calculus we use the opportunity to point out elements of this correspondence. What emerges is that there are typically more laws in the Bird-Meertens formalism than the quantifier calculus but the Bird-Meertens formalism exhibits a much better developed-separation of concerns. Note, however, that this section only covers a small part of the correspondence. To complete the picture the extra structure introduced at the different levels of the (original) Boom hierarchy is necessary. For a full account the reader is referred to [53].

The theorems presented in this section are more general than those in the publications of Bird and Meertens since their work is restricted to total functions. (Meertens [68] does discuss the issue of indeterminacy but this part of his paper — we regret to have to say — is in our view the least satisfactory.) A danger of generalisation is that it brings with it substantial overhead making a theory abstruse and unworkable. At this stage in our work, however, the generalisation from (total) functions to relations has been very positive bringing to mind a parallel with the extension of the domain of real numbers to complex numbers. The fact of the matter is that we are rarely aware of working with relations rather than functions. The following pages are intended to provide some justification for that claim.

## 16.2 Sum Relators

We begin our discussion with the so-called "sum" relators. Specifically, $F$ is a *sum relator* if for some relators $G$ and $H$ and for all specs $X$,

(16.1) $\quad F.X \;=\; G.X + H.X$

In words, $F$ is the (lifted) sum of $G$ and $H$.

The class of sum relators is very broad but, in spite of its generality, there is surprisingly much that we can say about the class. The most important aspect of such a relator $F$ is that we can identify the "constructors" of $\mu F$ bringing the notion of relator somewhat closer to the notion of polymorphic type as it would be defined in a conventional programming language. An additional technical aspect that proves to be very useful is that $F$-catamorphisms can be restricted without loss of generality to arguments that are the junc of two specs. These two aspects are considered in turn below. Throughout the remainder of this subsection we assume that equation (16.1) is in force.

### 16.2.1 Constructors

Let us consider what consequences equation (16.1) has on $\mu F$. We have the following simple calculation:

$$\begin{array}{ll} & \mu F \\ = & \qquad \{ \ \mu F \text{ is a fixpoint of } F \ \} \\ & F.\mu F \\ = & \qquad \{ \ \text{definition of } F\text{: (16.1)} \ \} \\ & G.\mu F \ + \ H.\mu F \\ = & \qquad \{ \ \text{definition of } +\text{: (12.18)} \ \} \\ & (\hookrightarrow \circ G.\mu F) \ \triangledown \ (\hookleftarrow \circ H.\mu F) \end{array}$$

Continuing with just the first component of this junc expression, we calculate:

$$\begin{array}{ll} & \hookrightarrow \circ G.\mu F \\ = & \qquad \{ \ \text{computation rule: theorem 12.71(c)} \ \} \\ & G.\mu F + H.\mu F \ \circ \ \hookrightarrow \\ = & \qquad \{ \ \text{definition of } F\text{: (16.1), } \mu F \ = \ F.\mu F \ \} \\ & \mu F \ \circ \ \hookrightarrow \end{array}$$

Similarly,

$$\hookleftarrow \circ H.\mu F \ = \ \mu F \circ \hookleftarrow$$

Thus, introducing names $\tau$ and $\eta$ for the two components of the above junc, we have established:

**Theorem 16.2 (Constructors)** For relators $F$, $G$ and $H$ such that $F \ = \ G + H$,

$$\mu F \ = \ \tau \triangledown \eta$$

where

$$\tau \;=\; \hookrightarrow \circ\; G.\mu F \;=\; \mu F \circ \hookrightarrow$$

and

$$\eta \;=\; \hookleftarrow \circ\; H.\mu F \;=\; \mu F \circ \hookleftarrow$$

□

A paraphrase of theorem 16.2 might be that all elements of $\mu F$ are constructed by injections of elements of $G.\mu F$ or elements of $H.\mu F$. For this reason we call $\tau$ and $\eta$ the *constructors* of $\mu F$.

Note that the constructors are bijections (since they are restrictions of the two bijections $\hookrightarrow$ and $\hookleftarrow$). For their domains we have:

$$\begin{array}{cl}
 & \tau{>} \\
= & \qquad \{ \text{ definition of } \tau\text{: theorem 16.2 } \} \\
 & (\hookrightarrow \circ\; G.\mu F){>} \\
= & \qquad \{ \text{ domains: (10.21) } \} \\
 & (\hookrightarrow{>} \circ\; G.\mu F){>} \\
= & \qquad \{ \hookrightarrow{>} \;=\; I \;\text{ : theorem 12.76(a) } \} \\
 & G.\mu F{>} \\
= & \qquad \{ \mu F \text{ is a monotype: (10.27) } \} \\
 & G.\mu F
\end{array}$$

and

$$\begin{array}{cl}
 & \tau{<} \\
= & \qquad \{ \text{ definition of } \tau\text{: theorem 16.2 } \} \\
 & (\mu F \circ \hookrightarrow){<} \\
= & \qquad \{ \text{ domains: (10.16) } \} \\
 & (\mu F \circ \hookrightarrow{<}){<} \\
= & \qquad \{ \hookrightarrow{<} \;=\; I + \perp\!\!\perp \;\text{ : theorem 12.76(b) } \} \\
 & (\mu F \circ I + \perp\!\!\perp){<} \\
= & \qquad \{ \mu F \;=\; G.\mu F + H.\mu F, + \text{ abides with composition } \} \\
 & (G.\mu F \;+\; \perp\!\!\perp){<} \\
= & \qquad \{ \text{ domains: (12.79) and (10.27) } \} \\
 & G.\mu F \;+\; \perp\!\!\perp
\end{array}$$

By a completely symmetrical argument we have:

$$\eta> \;=\; H.\mu F$$

and

$$\eta< \;=\; \perp\!\!\perp + H.\mu F$$

Combining these four domain calculations with the cup and cap abide properties and co-strictness of sum (see theorems 12.84 and 12.70) and summarising we have established:

**Theorem 16.3** The constructors $\tau$ and $\eta$ are both bijections with the following domain properties:

(a) $\tau> \;=\; G.\mu F$
(b) $\tau< \;=\; G.\mu F + \perp\!\!\perp$
(c) $\eta> \;=\; H.\mu F$
(d) $\eta< \;=\; \perp\!\!\perp + H.\mu F$
(e) $\tau< \sqcup \eta< \;=\; \mu F$
(f) $\tau< \sqcap \eta< \;=\; \perp\!\!\perp$
$\Box$

Interpretating these statements in the relational model we have proved that the constructors $\tau$ and $\eta$ establish a (1-1) correspondence between the elements of $\mu F$ and the elements of the union of $G.\mu F$ and $H.\mu F$ in such a way that elements constructed by $\tau$ are distinct from those constructed by $\eta$.

### 16.2.2 Sum-relator Catamorphisms

Let us now investigate the structure of the catamorphisms of a sum relator. We have:

$$\begin{array}{ll}
 & (\!|R|\!) \\
= & \qquad \{ \text{ domain trading: theorem 13.19(b) } \} \\
 & (\!|R \circ (G.R + H.R)<|\!) \\
= & \qquad \{ \text{ } +,\ G,\ H \text{ are relators: theorem 10.34 } \} \\
 & (\!|R \circ G.R< + H.R<|\!) \\
= & \qquad \{ \text{ definition } +\text{: (12.18) } \} \\
 & (\!|R \circ (\hookrightarrow \circ G.R<) \triangledown (\hookleftarrow \circ H.R<)|\!)
\end{array}$$

$$
\begin{array}{ll}
= & \{ \text{ spec-junc fusion: theorem 12.68 } \} \\
& (\!|(R \circ \hookrightarrow \circ G.R_<) \triangledown (R \circ \hookleftarrow \circ H.R_<)|\!)
\end{array}
$$

This calculation shows that we may assume without loss of generality that for every $R$ there exist specs $S$ and $T$ such that

$$(\!|R|\!) \;=\; (\!|S \triangledown T|\!)$$

Specifically,

$$S \;=\; R \circ \hookrightarrow \circ G.R_<$$

and

$$T \;=\; R \circ \hookleftarrow \circ H.R_<$$

Note that from $(\!|R|\!) \;=\; (\!|R|\!) \circ \mu F$ and the fact that $\mu F$ can be expressed as a junc it follows that every catamorphism can also be expressed as a junc. This observation is most useful when combined with the cancellation property of junc (see theorem 12.80). To see why let us first observe the following instantiation of the junc-cancellation property:

**Lemma 16.4** For $\trianglelefteq \in \{\sqsupseteq, =, \sqsubseteq\}$,

$$X \circ \mu F \trianglelefteq Y \circ \mu F \quad\equiv\quad X \circ \tau \trianglelefteq Y \circ \tau \;\wedge\; X \circ \eta \trianglelefteq Y \circ \eta$$

**Proof**

$$
\begin{array}{ll}
& X \circ \mu F \;\trianglelefteq\; Y \circ \mu F \\
\equiv & \{ \text{ theorem 16.2 } \} \\
& X \circ \tau \triangledown \eta \;\trianglelefteq\; Y \circ \tau \triangledown \eta \\
\equiv & \{ \text{ spec-junc fusion: theorem 12.68 } \} \\
& (X \circ \tau) \triangledown (X \circ \eta) \;\trianglelefteq\; (Y \circ \tau) \triangledown (Y \circ \eta) \\
\equiv & \{ \text{ junc cancellation: theorem 12.80 } \} \\
& X \circ \tau \trianglelefteq Y \circ \tau \;\wedge\; X \circ \eta \trianglelefteq Y \circ \eta
\end{array}
$$

□

Combining lemma 16.4 with the unique extension property of catamorphisms we derive a characterisation of $F$-catamorphisms (for sum relators $F$, of course), namely:

**Theorem 16.5 (UEP for Sum Relators)**

$$\begin{array}{ll} & X \circ \mu F \;=\; (\!|R \triangledown S|\!) \\ \equiv & \\ & X \circ \tau \;=\; R \circ G.(X \circ \mu F) \quad \wedge \quad X \circ \eta \;=\; S \circ H.(X \circ \mu F) \end{array}$$

**Proof**

$$\begin{array}{ll} & X \circ \mu F = (\!|R \triangledown S|\!) \\ \equiv & \quad \{ \text{ catamorphism uep: theorem 13.11 } \} \\ & X \circ \mu F = R \triangledown S \circ F.(X \circ \mu F) \circ \mu F \\ \equiv & \quad \{ \text{ lemma 16.4 } \} \\ & \quad X \circ \tau = R \triangledown S \circ F.(X \circ \mu F) \circ \tau \\ & \wedge \quad X \circ \eta = R \triangledown S \circ F.(X \circ \mu F) \circ \eta \end{array}$$

Proceeding further with just the first of the conjuncts on the right hand side of the equivalence (the other being completely symmetrical) we have:

$$\begin{array}{ll} & R \triangledown S \circ F.(X \circ \mu F) \circ \tau \\ = & \quad \{ \text{ definition of } \tau\text{: theorem 16.2 } \} \\ & R \triangledown S \circ F.(X \circ \mu F) \circ \mu F \circ \hookrightarrow \\ = & \quad \{ \ \mu F = F.\mu F,\ \mu F = \mu F \circ \mu F \ \} \\ & R \triangledown S \circ F.(X \circ \mu F) \circ \hookrightarrow \\ = & \quad \{ \text{ definition of } F\text{: (16.1), junc-sum fusion: (12.66) } \} \\ & (R \circ G.(X \circ \mu F)) \triangledown (S \circ H.(X \circ \mu F)) \circ \hookrightarrow \\ = & \quad \{ \text{ junc computation: theorem 12.71(a) } \} \\ & R \circ G.(X \circ \mu F) \end{array}$$

Back-substituting the desired theorem is obtained.
□

Compared with the general uep property (theorem 13.11) theorem 16.5 splits the task of deriving a catamorphism realising a given spec into two separate components, one for each of the constructors. This separation is further reflected in the *computation rules* for $\tau$ and $\eta$:

**Theorem 16.6 (Computation Rule)**

$$\begin{array}{lllll} \text{(a)} & (\!|R \triangledown S|\!) \circ \tau & = & R \circ G.(\!|R \triangledown S|\!) \\ \text{(b)} & (\!|R \triangledown S|\!) \circ \eta & = & S \circ H.(\!|R \triangledown S|\!) \end{array}$$

**Proof** Instantiate theorem 16.5 with $X = (\![R \triangledown S]\!)$ and simplify using the fact that $(\![R \triangledown S]\!) = (\![R \triangledown S]\!) \circ \mu F$.
□

Several other properties of sum relators can be derived simply by instantiating the more general properties of catamorphisms listed in section 13, in particular the fusion and monotonicity properties of catamorphisms (theorems 13.14 and 13.15). The benefit that is gained is that, in each case, the premise in the theorem can be expressed as a conjunction of two simpler premises, thus decomposing the proof obligations. We postpone performing this exercise, however, until we have added more structure to our class of relators.

## 16.3 Polymorphically Grounded Relators

A typical characteristic of monotypes occurring in programming problems is that their elements are generated from a base (mono)type by application of one or more operations. For example, the Peano numbers are generated from the set containing just zero by the successor operation. Polymorphic types, such as list or tree, are families of monotypes parameterised by some base (mono)type. We call such types *polymorphically grounded* types (or rather we call their defining relators polymorphically grounded), the word "grounded" referring to the existence of a base monotype. In this section we abstract a definition of "polymorphically grounded" relator. We do this in two steps. First, we abstract what it means for a relator to be grounded. Then, in order to capture the "polymorphic" element, we abstract sufficient conditions for the existence of a "map" operator. We conclude the section with some consequences of the obtained definition.

### 16.3.1 Grounded Relators

The mechanism needed to introduce the notion of a ground monotype into our class of relators is straightforward: we consider a sum relator and choose the left component of the sum to be a constant relator, i.e. we consider the case that $G.X = A$ for some monotype $A$ and all specs $X$, thereby specializing $F$ to the form:

(16.7) $F.X \;=\; A + H.X$

Using this the constructors are

$$(16.8)\quad \tau \;=\; \mu F \circ \hookrightarrow \;=\; \hookrightarrow \circ A$$

$$(16.9)\quad \eta \;=\; \mu F \circ \hookleftarrow \;=\; \hookleftarrow \circ H.\mu F$$

The form of the constructors provides some motivation for the chosen restriction on $F$. Specifically, suppose we interpret monotypes as sets and $f \circ B$, for monotype $B$ and imp $f$, also as a set, namely the set obtained by applying the function $f$ to the elements of $B$. Then the set $\mu F$ is formed by "juncing" two sorts of sets, the set of "ground" elements, i.e. those elements formed by $\tau$, i.e. by applying $\hookrightarrow$ to elements of $A$, or "non-ground" elements, i.e. those built by $\eta$ from existing elements of $\mu F$. We call relators $F$ satisfying (16.7) "grounded" relators.

In the case that the relator $H$ is $\sqcup$-continuous we can apply a well-known fixpoint theorem to deduce that the elements of $\mu F$ are *finitely* generated. More interestingly, if $H$ is denumerably $\sqcup$-junctive and strict we can express $\mu F$ as the cup of a sequence of monotypes generated from $A$. Specifically, letting

$$(16.10)\; B \;=\; A + \perp\!\!\perp$$

and

$$(16.11)\; \theta.X \;=\; \perp\!\!\perp + H.X$$

for all specs $X$, we have

**Theorem 16.12** If relator $H$ is denumerably $\sqcup$-junctive then

$$\mu F \;=\; \sqcup(i:\; i \geq 0 :\; \theta^i.B)$$

Moreover, $\{i:\; i \geq 0:\; \theta^i.B\}$ is a set of monotypes and, if $H$ is strict (i.e. $H.\perp\!\!\perp \;=\; \perp\!\!\perp$) and $\sqcap$-junctive, the elements of the set are mutually disjoint.
□

(The notation $\theta^i$ in the statement of the theorem denotes the $i$-fold composition of function $\theta$. That is,

$$\begin{array}{lcll} \theta^0.X & = & X, & \text{and} \\ \theta^{i+1} & = & \theta.\theta^i.X & \text{for all natural numbers } i. \end{array}$$

)

The import of this theorem is that, if $H$ is denumerably $\sqcup$-junctive, the monotype $\mu F$ is the cup of a set of monotypes, and, if in addition $H$ is $\sqcap$-junctive and strict, there is a "size" function defined on its elements. Elements of size $i$, for natural number $i$, are the elements of $\theta^i.B$. Elements of size zero are thus the elements of $B$ $(= \theta^0.B)$ which are, in turn, elements of the ground type $A$ "tagged" by $+\perp\!\!\!\perp$. Elements of size $i+1$ are generated by application of $H$ to elements of size $i$ and then "tagging" these elements by $\perp\!\!\!\perp+$. Essentially, therefore, the elements of $\mu F$ are generated by a finite number of applications of $H$ accompanied by a tagging process that ensures that the number of times $H$ has been applied can always be recovered by inspection of the element itself. (Note, however, that if $H.X$ is constantly $\perp\!\!\!\perp$ the sets $\theta^{i+1}.B$ are all $\perp\!\!\!\perp$.)

Although the proof is quite long it is very straightforward. It is, however, worth studying at least briefly as a good illustration of the use of the abide laws of disjoint sum given in section 12.4.7.

**Proof** The "well-known" theorem that we referred to above (appropriately instantiated) says that if $F$ is $\sqcup$-continuous then

(16.13) $\mu F \quad = \quad \sqcup(i: \ i \geq 0: \ F^i.\perp\!\!\!\perp)$

(The theorem is sometimes called "Kleene's theorem", sometimes "Tarski's theorem". See [60] for a discussion of its origin.)

In order to apply the theorem in a form more suited to our purposes we need to break down the proof obligations into separate parts. First, we observe that the denumerable $\sqcup$-junctivity of $F$ follows from the denumerable $\sqcup$-junctivity of $H$. Second, we show that the right side of (16.13) can be rewritten in the form stated in the theorem. Third and fourth, we observe that the specs $\theta^i.B$ (for $i$ ranging over the natural numbers) are monotypes and, with the given assumption, mutually disjoint.

That $\sqcup$-continuity of $F$ follows from the denumerable $\sqcup$-junctivity of $H$ is clear from the definition of $F$ (see (16.7)): the function $F$ is the composition of the function $A+$ after the function $H$ and the former is universally $\sqcup$-junctive (see theorem 12.85). Thus $F$ is denumerably $\sqcup$-junctive. But, denumerable $\sqcup$-junctivity of a function is equivalent to its being both $\sqcup$-continuous and $\sqcup$-junctive. (See [36] for a proof.) So $F$ is $\sqcup$-continuous. For later use we remark that, by the same argument, $\theta$ also inherits denumerable $\sqcup$-junctivity from $H$. Moreover, from the co-strictness of sum (theorem 12.70), it is strict if $H$ is strict.

Now we proceed to rewrite the right side of (16.13). First, we rewrite the definition of $F$ in a way that introduces $\theta$.

$$\begin{array}{ll} & F.X \\ = & \quad \{ \text{ definition: (16.7) } \} \\ & A + H.X \\ = & \quad \{ \text{ plat calculus } \} \\ & (A \sqcup \perp\!\!\perp) \; + \; (\perp\!\!\perp \; \sqcup \; H.X) \\ = & \quad \{ \; + \text{ and } \sqcup \text{ abide: theorem 12.84(b) } \} \\ & (A + \perp\!\!\perp) \sqcup (\perp\!\!\perp \; + \; H.X) \\ = & \quad \{ \text{ definitions: (16.10) , (16.11) } \} \\ & B \sqcup \theta.X \end{array}$$

Summarising,

(16.14) $F.X \quad = \quad B \sqcup \theta.X$

In particular,

(16.15) $F.X \quad \sqsupseteq \quad \theta.X$

Next, we claim that

(16.16) $\sqcup(i : \; i \geq 0 : \; F.\theta^i.B) \quad = \quad \sqcup(i : \; i \geq 0 : \; \theta^i.B) \quad \sqsubseteq \quad \mu F$

from which it follows that, if $F$ is denumerably $\sqcup$-junctive,

(16.17) $F.\sqcup(i : \; i \geq 0 : \; \theta^i.B) \quad = \quad \sqcup(i : \; i \geq 0 : \; \theta^i.B) \quad \sqsubseteq \quad \mu F$

That is, $\sqcup(i : \; i \geq 0 : \; \theta^i.B)$ is a fixed point of $F$ that is at most $\mu F$. Since $\mu F$ is the least fixed point of $F$ the two are equal.

The proof of (16.16) proceeds as follows: we have:

$$\begin{array}{ll} & \sqcup(i : \; i \geq 0 : \; F.\theta^i.B) \\ = & \quad \{ \text{ (16.14) } \} \\ & \sqcup(i : \; i \geq 0 : \; B \; \sqcup \; \theta^{i+1}.B) \\ = & \quad \{ \text{ plat calculus } \} \\ & \sqcup(i : \; i \geq 0 : \; \theta^i.B) \\ \sqsubseteq & \quad \{ \; B \sqsubseteq F.\perp\!\!\perp, \text{(16.15) } \} \\ & \sqcup(i : \; i \geq 0 : \; F^i.\perp\!\!\perp) \\ \sqsubseteq & \quad \{ \text{ elementary induction } \} \\ & \mu F \end{array}$$

This completes the first half of the theorem.

Induction is needed to show that $\theta^i.B$ is a monotype. The basis is $B$ is a monotype: this is true because $B$ is the result of applying the relator $+\perp\!\!\perp$ to the monotype $A$. The induction step is also straightforward: $\theta$ preserves monotypes because it is the composition of two relators ($\perp\!\!\perp+$ and $H$) and hence is itself a relator.

The final step is to investigate the circumstances under which these monotypes are disjoint. Formally we prove that if $H$ is, in addition, $\sqcap$-junctive then, for all natural numbers $i$ and $j$, $\theta^{i+j+1}.B \;\sqcap\; \theta^i.B \;\;=\;\; \perp\!\!\perp$.

$$\begin{array}{ll}
 & \theta^{i+j+1}.B \sqcap \theta^i.B \\
= & \qquad \{ \;\; \theta \text{ inherits } \sqcap\text{-junctivity from } H. \\
 & \qquad\quad \text{Hence so does } \theta^i \;\; \} \\
 & \theta^i.(\theta^{j+1}.B \sqcap B) \\
= & \qquad \{ \;\; \text{definitions of } \theta \text{ and } B \;\; \} \\
 & \theta^i.((\perp\!\!\perp + H.\theta^j.B) \;\sqcap\; (A + \perp\!\!\perp)) \\
= & \qquad \{ \;\; + \text{ and } \sqcap \text{ abide: theorem 12.84(d)} \;\; \} \\
 & \theta^i.(\perp\!\!\perp + \perp\!\!\perp) \\
= & \qquad \{ \;\; \text{co-strictness of } +\text{: theorem 12.70,} \\
 & \qquad\quad \theta \text{ inherits strictness from } H \;\; \} \\
 & \perp\!\!\perp
\end{array}$$

□

The fact that $\mu F$ is itself a catamorphism (see the identity rule: theorem 13.20) leads one to speculate that — in the case of strict, denumerably $\sqcup$-junctive $H$ — all catamorphisms are finitely computable when applied to elements of $\mu F$ (provided their arguments are computable). This is indeed the case. We leave it to the reader to verify (using the computation rule: 13.4(a)) that for all specs $R$ and $S$,

$$(\!|R \triangledown S|\!) \circ B \;\;=\;\; R \circ A \circ \hookrightarrow_{\cup}$$

and, for all $i \geq 0$,

$$(\!|R \triangledown S|\!) \circ \theta^{i+1}.B \;\;=\;\; S \circ H.((\!|R \triangledown S|\!) \circ \theta^i.B) \circ \hookleftarrow_{\cup}$$

For anyone wishing to base a programming language on our calculus the details of this last remark are highly significant. The remark is also significant to programmers but the details less so: all the programmer need know is that if $H$ obeys the three conditions stipulated in theorem 16.12 each element of $\mu F$ has an easily identified "size" and the application of an $F$-catamorphism to such an element can be evaluated in time proportional to the product of the element's size and the complexity of the arguments of the catamorphism.

The extra structure introduced into grounded types makes little difference to the computation rule; where it is needed we shall simply instantiate theorem 16.6(a) with $G.X = A$. The fusion property for ground-relator-catamorphisms is worth stating, however, because we can exploit the extra structure to strengthen the general result.

**Theorem 16.18 (Ground-Relator Fusion)** For $\trianglelefteq$ in $\{\sqsubseteq, =, \sqsupseteq\}$,

$$\begin{array}{ll} & U \circ (\!|R \triangledown S|\!) \;\trianglelefteq\; (\!|P \triangledown Q|\!) \\ \Leftarrow & \quad U \circ R \circ A \;\trianglelefteq\; P \circ A \quad \wedge \quad U \circ S \circ H.I \;\trianglelefteq\; Q \circ H.U \end{array}$$

□

The added-value of this theorem relative to theorem 13.14 — apart from the antecedent having been split into two conjuncts — is the introduction of the domain restrictions $A$ and $H.I$ in the first and second conjuncts, respectively, of the antecedent. Note that

$$U \circ R \circ A \;\trianglelefteq\; P \circ A \quad \Leftarrow \quad U \circ R \;\trianglelefteq\; P$$

Thus the first conjunct in the antecedent has been weakened. (That it is a true weakening is easily seen by taking $A = \perp\!\!\perp$.) The second conjunct has been similarly weakened.

**Proof** Let $\trianglelefteq \in \{\sqsupseteq, =, \sqsubseteq\}$. Then

$$\begin{array}{ll} & U \circ (\!|R \triangledown S|\!) \;\trianglelefteq\; (\!|P \triangledown Q|\!) \\ \equiv & \quad \{ \text{ domain trading: theorem 13.19(a), since } \; A{+}H.I = F.I \\ & \quad\;\; \text{and junc-sum fusion: theorem 12.66(a) } \} \\ & U \circ (\!|(R \circ A) \triangledown (S \circ H.I)|\!) \;\trianglelefteq\; (\!|P \triangledown Q|\!) \\ \Leftarrow & \quad \{ \text{ catamorphism fusion: theorem 13.14 } \} \\ & U \circ (R \circ A) \triangledown (S \circ H.I) \;\trianglelefteq\; P \triangledown Q \circ F.U \\ \equiv & \quad \{ \text{ spec-junc fusion: theorem 12.68;} \\ & \quad\;\; \text{definition of } F\text{: (16.7),} \\ & \quad\;\; \text{and + abides with composition: theorem 12.66(b) } \} \end{array}$$

$$
\begin{array}{ll}
 & (U \circ R \circ A) \triangledown (U \circ S \circ H.I) \quad \unlhd \quad (P \circ A) \triangledown (Q \circ H.U) \\
\equiv & \qquad \{ \text{ junc cancellation: theorem 12.80(a) } \} \\
 & U \circ R \circ A \;\unlhd\; P \circ A \;\wedge\; U \circ S \circ H.I \;\unlhd\; Q \circ H.U
\end{array}
$$

□

### 16.3.2 Introducing Polymorphism via Map

We come now to the first of the primitive operators in the Bird-Meertens formalism, namely the map operator. Section 14.1 provides the appropriate mechanism for introducing such an operator: we must express $F$ in the form $I\otimes$ for some binary relator $\otimes$. This we can do by choosing $A = K.I$ for some relator $K$ and defining binary relator $\otimes$ by

(16.19) $R \otimes S \quad = \quad K.R + H.S$

Accordingly we have:

(16.20) $F.X \quad = \quad (I\otimes).X \quad = \quad K.I + H.X$

Note that $K.I$ is a monotype so that $F$ is indeed grounded. It is also polymorphic in the sense that we have defined a family of relators, namely the set of relators $(B\otimes)$ for $B$ ranging over all monotypes. More importantly we can instantiate the theorems of section 14 to obtain the sought-after map operator. Specifically, instantiating definition 14.3 and citing theorem 14.18, we have:

**Theorem 16.21 (Map)** The function $\varpi$ from specs to specs defined by

$$\varpi R \;=\; (\!|K.R + H.I|\!)$$

is a relator.

□

The function $\varpi$ defines a family of monotypes, namely the monotypes $\varpi B$ where $B$ ranges over monotypes. In particular, $\varpi I = \mu F$. For each spec $R$, the spec $\varpi R$ has left domain $\varpi(R<)$ and right domain $\varpi(R>)$. In addition, for monotypes $A$ and $B$ and imps $f \in A \longleftarrow B$, $\varpi f \in \varpi A \longleftarrow \varpi B$. An instance of such a relator is the List relator which is sometimes denoted by $*$. In functional programming texts $*f$ is commonly called "map $f$" (and sometimes written that way too) and denotes a function from lists to lists that "maps" the given function $f$ over the elements of the argument list (i.e. constructs a list of the same length as the argument list whereby the elements are obtained by applying $f$ to each of the elements of the argument list). This then is the origin of the name “map” for $\varpi$.

We will mostly use another but equivalent definition for map that exploits the particular structure of the relator $\otimes$. That definition is obtained by first instantiating the map fusion theorem (theorem 14.14) of section 14.

**Theorem 16.22 (Map Fusion)**

$$(\!|P \triangledown Q|\!) \circ \varpi R \;=\; (\!|(P \circ K.R) \triangledown Q|\!)$$

**Proof**

$$\begin{array}{cl}
 & (\!|P \triangledown Q|\!) \circ \varpi R \\
= & \quad \{ \text{ map fusion: theorem 14.14, definition of } \otimes\text{: (16.19) } \} \\
 & (\!|P \triangledown Q \circ K.R + H.I|\!) \\
= & \quad \{ \text{ junc-sum fusion: theorem 12.66(a) } \} \\
 & (\!|(P \circ K.R) \triangledown Q \circ K.I + H.I|\!) \\
= & \quad \{ \text{ domain trading: theorem 13.19(c), } K.I + H.I \;=\; F.I \text{ } \} \\
 & (\!|(P \circ K.R) \triangledown Q|\!)
\end{array}$$

□

**Theorem 16.23 (Map – Alternative Definition)**

$$\varpi R \;=\; (\!|(\tau \circ K.R) \triangledown \eta|\!)$$

**Proof**

$$\begin{array}{cl}
 & \varpi R \\
= & \quad \{ \; \varpi \text{ is a relator } \} \\
 & \varpi I \circ \varpi R \\
= & \quad \{ \; \varpi I \;=\; \mu F \;=\; (\!|\mu F|\!) \; \} \\
 & (\!|\mu F|\!) \circ \varpi R \\
= & \quad \{ \; \mu F \;=\; \tau \triangledown \eta, \text{ map fusion: theorem 16.22 } \} \\
 & (\!|(\tau \circ K.R) \triangledown \eta|\!)
\end{array}$$

□

The reason why we sometimes prefer this definition is that catamorphisms of the shape $(\!|R \triangledown \eta|\!)$ enjoy many properties.

Instantiating the computation rule (16.6) with the revised definition of $F$ — (16.7) — and the above definition of $\varpi$ we obtain the following computation rules:

$$\varpi R \circ \tau \;=\; \tau \circ K.R$$
$$\varpi R \circ \eta \;=\; \eta \circ H.\varpi R$$

These two equations can be recombined into one using theorem 12.80 viz:

(16.24) $\varpi R \circ \tau \triangledown \eta \;=\; (\tau \circ K.R) \triangledown (\eta \circ H.\varpi R)$

Recalling that

$$\varpi I \;=\; \mu F \;=\; \tau \triangledown \eta \;=\; (\tau \circ K.I) \triangledown (\eta \circ H.\varpi I)$$

(see theorems 14.12, 16.2 and equations (16.8), (16.9) and (16.20)) one can view $\varpi R$ as a spec which, when applied to an element of $\mu F$, applies $R$ to the ground elements but does not destroy the original structure.

## 16.4 Defining Reduce

The second primitive in the Bird-Meertens formalism is called "reduce" and is denoted by the symbol "/". In the context of our work, reduce is a function from specs to specs. We shall adopt the same symbol but use it as a prefix operator in order to be consistent with our convention of always writing function and argument in that order. Thus we write $/S$ and read "reduce with $S$" or just "reduce $S$".

(In choosing to write reduce as a prefix operator we are turning the clock back to Backus' Turing award lecture [11] rather than following the example of Bird and Meertens. In the context of Bird and Meertens' original work reduce was a binary infix operator with argument a pair consisting of a binary operator, say $\oplus$, and a list, say $x$, thus giving $\oplus/x$. In the course of time it was recognised that calculations and laws could be made more compact by working with the *function* $(x \mapsto \oplus/x)$ rather than the *object* $\oplus/x$. To achieve the compactness the notation $\oplus/$ (or sometimes $(\oplus/)$) was adopted for the function, the process of abstracting one of the arguments of a binary operator being commonly referred to as "sectioning". By this development, presumably, they came to the convention of using "/" as a postfix operator. Since our concern is to profit from what has been learnt rather than repeat the learning process we shall not adopt their notation in its entirety.)

The idea behind reduce is that it should have a complementary behaviour to map. Recall that map, applied to an element of $\mu F$, leaves the structure unchanged but applies its argument to the ground elements. Reduce should do the opposite: leave the ground elements unchanged but destroy the structure. Since a catamorphism does both (modifies the ground elements and the structure) we formulate the requirement on reduce as being that every catamorphism is factorisable into a reduce composed with a map. I.e. for all specs $R$ and $S$,

$$/S \circ \varpi R \;=\; (\!|R \triangledown S|\!)$$

Let us try to calculate a suitable definition for $/S$.

$$\begin{array}{ll}
 & /S \circ \varpi R \\
= & \quad \{ \text{ We try to express } /S \text{ as a catamorphism} \\
 & \quad\;\; \bullet \;\; /S \;=\; (\!|P \triangledown Q|\!) \;\; \} \\
 & (\!|P \triangledown Q|\!) \circ \varpi R \\
= & \quad \{ \text{ map fusion: theorem 16.22 } \} \\
 & (\!|(P \circ K.R) \triangledown Q|\!)
\end{array}$$

Now we cannot choose $P$ and $Q$ (for arbitrary relator $K$) such that

$$(\!|(P \circ K.R) \;\triangledown\; Q|\!) \;=\; (\!|R \triangledown S|\!)$$

But if we take $P \,=\, I$ and $Q \,=\, S$, i.e. we define the reduce operator by:

(16.25) $/S \;=\; (\!|K.I \;\triangledown\; S|\!)$

then we have established the following factorisation property:

**Lemma 16.26**

$$/S \circ \varpi R \;=\; (\!|K.R \;\triangledown\; S|\!)$$

□

Some simplification of (16.25) is possible using domain trading and junc-sum fusion (theorems 13.19(a) and 12.66(a)). Specifically, we claim that the term $K.I$ in (16.25) may be replaced by $I$ (the verification being left to the reader) which leads us to the following definition of reduce:

**Definition 16.27 (Reduce)**

$$/S \;=\; (\!|I \triangledown S|\!)$$

□

For $/S$ we have the following computation rules (obtained by instantiating theorem 16.6 with $G.X = K.I$ for all $X$):

$$\begin{array}{rcl} /S \circ \tau & = & K.I \\ /S \circ \eta & = & S \circ H./S \end{array}$$

So one can view $/S$ as a spec which, when applied to an element of $\mu F$, strips the ground elements of the constructor $\tau$ and replaces the constructor $\eta$ by $S$.

## 16.5 Monadic Relators

As mentioned before, with $F$ having the form given by (16.20), we cannot factorise every catamorphism into a reduce and a map for arbitrary relator $K$. For relator $K$ defined by $K.X = X$ —i.e. the identity relator—we can, since

$$\begin{array}{ll} & (\!|R \triangledown S|\!) \\ = & \quad \{ \bullet \quad K.R = R \ \} \\ & (\!|K.R \triangledown S|\!) \\ = & \quad \{ \text{ catamorphism factorisation: theorem 16.26 } \} \\ & /S \circ \varpi R \end{array}$$

So we further specialise the binary relator $\otimes$ and the unary relator $F$ by defining

(16.28) $K.X \quad = \quad X$

(16.29) $X \otimes Y \quad = \quad X + H.Y$

and

(16.30) $F.X \;=\; (I\otimes).X \;=\; I + H.X$

for all specs $X$ and $Y$. Then we have established the all-important:

**Theorem 16.31 (Factorisation)** With relator $F$ defined by (16.29) and (16.30) we have, for all specs $R$ and $S$,

$$(\!|R \triangledown S|\!) \quad = \quad /S \circ \varpi R$$

□

The importance of this theorem derives from the fact that it enhances further decomposition of calculations with catamorphisms. Instead of working with the entire catamorphism one works with the components $/S$ and $\varpi R$. Laws are also formulated concerning the individual behaviours of reduce and map as well as their interaction. The advantage is that the laws become extremely compact and thus more manageable, the disadvantage is that there are more of them. Let us illustrate this by considering the computation rules, the unique extension property and the fusion properties of reduce and map.

First, the definitions of the constructors $\tau$ and $\eta$ are specialised accordingly:

(16.32) $\tau \;=\; \mu F \circ \hookrightarrow \;=\; \hookrightarrow$

(16.33) $\eta \;=\; \mu F \circ \hookleftarrow \;=\; \hookleftarrow \circ H.\mu F$

Whereas before we had two computation rules, one for each of the constructors, we now have four rules:

**Theorem 16.34 (Computation Rule)**

(a) $\varpi R \circ \tau \;=\; \tau \circ R$
(b) $\varpi R \circ \eta \;=\; \eta \circ H.\varpi R$
(c) $/S \circ \tau \;=\; I$
(d) $/S \circ \eta \;=\; S \circ H./S$
□

(Of course these rules can be recombined into two using the factorisation theorem, and whether one chooses to do so is a matter of taste.)

In the case of the unique extension property there is little gain from the use of the factorisation theorem.

**Theorem 16.35 (Unique Extension Property)**

$$X \circ \mu F \;=\; /S \circ \varpi R$$
$$\equiv$$
$$X \circ \tau \;=\; R \;\wedge\; X \circ \eta \;=\; S \circ H.(X \circ \mu F)$$
□

On the other hand, the fusion law becomes more compact since it suffices to state the law only for a reduce. We call the resulting theorem a "leapfrog" rule because its symbol dynamics is that a reduce "leapfrogs" from one side to the other of a composition of two specs. (The more general fusion law can be recovered by combining the reduce leapfrog theorem with the monotonicity of the relator $\varpi$.)

**Theorem 16.36 (Reduce Leapfrog)** For $\trianglelefteq$ in $\{\sqsupseteq, =, \sqsubseteq\}$,

$$R \circ /S \quad \trianglelefteq \quad /T \circ \varpi R \quad \Leftarrow \quad R \circ S \circ H.I \quad \trianglelefteq \quad T \circ H.R$$

**Proof**

$$\begin{array}{ll} & R \circ /S \;\trianglelefteq\; /T \circ \varpi R \\ \equiv & \quad \{ \text{ definition 16.27, factorisation: theorem 16.31 } \} \\ & R \circ (\![I \triangledown S]\!) \;\trianglelefteq\; (\![R \triangledown T]\!) \\ \Leftarrow & \quad \{ \text{ ground relator fusion: theorem 16.18, } \; A = K.I = I \;\} \\ & R \circ I \circ I \;\trianglelefteq\; R \circ I \quad \wedge \quad R \circ S \circ H.I \;\trianglelefteq\; T \circ H.R \\ \equiv & \quad \{ \text{ calculus } \} \\ & R \circ S \circ H.I \;\trianglelefteq\; T \circ H.R \end{array}$$

□

Because $\mu F$ is expressible as a catamorphism, it too can be factorised:

**Theorem 16.37 (Identity Rule)**

$$/\eta \circ \varpi\tau \quad = \quad \varpi I$$

**Proof**

$$\begin{array}{ll} & /\eta \circ \varpi\tau \\ = & \quad \{ \text{ factorisation: theorem 16.31 } \} \\ & (\![\tau \triangledown \eta]\!) \\ = & \quad \{ \text{ constructors: theorem 16.2 } \} \\ & (\![\mu F]\!) \\ = & \quad \{ \text{ identity rules: theorems 13.20 and 14.12 } \} \\ & \varpi I \end{array}$$

□

Theorem 16.37 is one of those theorems that, because of their simplicity, are very often overlooked and yet prove to be vital.

A special reduce is $/\eta$ (for list-structures this is the "flattening" catamorphism; it maps a list of lists to a list). For this catamorphism there exist two special leapfrog properties:

**Theorem 16.38 ($/\eta$ Leapfrog)**

(a) $/S \circ /\eta \;=\; /S \circ \varpi/S$
(b) $\varpi R \circ /\eta \;=\; /\eta \circ \varpi\varpi R$

**Proof** Immediate from the reduce leapfrog rule — theorem 16.36 — and the two $\eta$-computation rules — theorem 16.34(b) and (d).
□

**Corollary 16.39** The triple ($\varpi$, $\tau$, $/\eta$) is a monad in the following sense:

(a) $\varpi$ is a relator.
(b) $\tau \;\in\; \varpi \leftarrow\!\!\!\!\rightarrow I\!I$
(c) $/\eta \;\in\; \varpi \leftarrow\!\!\!\!\rightarrow \varpi\varpi$
(d) $/\eta \circ \varpi\tau \;=\; \varpi I$
(e) $/\eta \circ \tau \;=\; I$
(f) $/\eta \circ /\eta \;=\; /\eta \circ \varpi/\eta$

**Proof** Part (a) has already been mentioned. Parts (b) and (e) follow from the computation rule of $\tau$ (theorem 16.34), (c) and (f) follow from theorem 16.38 and (d) is just the identity rule.
□

The concept of a monad is highly significant and is given due prominence in the mathematical literature. (See for instance [13, 58]. Note that monads are also called "triples".) In the computing science literature the importance of monads is as yet difficult to assess but appears to be steadily growing, the best known example being lists: a monad is formed by the triple $*$, [_] and *flatten*, where $*$ denotes the list map operation discussed earlier, [_] is the function that constructs a singleton list, and *flatten* is the function that "flattens" a list of lists into a single list. See for instance [92] for examples of particular relevance to the design and implementation of functional programming languages.

The existence of a monad structure is the reason why we call the relator of this subsection a "monadic" relator.

## 16.6 Pointed Relators and Filter

The third, and final, primitive operator in the Bird-Meertens formalism is called "filter" and denoted by $\triangleleft$. The function of $\triangleleft p$ ( read "filter with $p$", or just "filter $p$") is just to filter out the elements in a given data structure that do not satisfy the predicate $p$.

There are two obvious requirements on the definition of a filter operation. The first is that $\triangleleft \mathit{true}$ should be the identity function on $\mu F$. The second is that $\triangleleft \mathit{false}$ should return an "empty" data-structure. In order to meet the latter requirement we introduce a so-called "unit element" into the definition of $H$, viz:

(16.40) $H.X \;\; = \;\; \mathbb{1} + J.X$

where $J$ is a relator. Consequently, $F$ is specialised to:

(16.41) $F.X \; = \; I \; + \; (\mathbb{1} \; + \; J.X)$

with the two constructors we already have

(16.42) $\tau \;\; = \;\; \mu F \circ \hookrightarrow \;\; = \;\; \hookrightarrow$

(16.43) $\eta \;\; = \;\; \mu F \circ \hookleftarrow \;\; = \;\; \hookleftarrow \;\circ\; \mathbb{1} + J.\mu F$

and two new ones

(16.44) $\Box \;\; = \;\; \mu F \circ \hookleftarrow \circ \hookrightarrow \;\; = \;\; \hookleftarrow \circ \hookrightarrow \circ \mathbb{1}$

(16.45) $+\!\!\!+ \;\; = \;\; \mu F \circ \hookleftarrow \circ \hookleftarrow \;\; = \;\; \hookleftarrow \circ \hookleftarrow \;\circ\; J.\mu F$

Note that

(16.46) $\eta \;\; = \;\; \Box \triangledown +\!\!\!+$

Because this relator has a disjoint unit in its ground as well, we call these relators "pointed relators". Again we want to point out that because this relator $F$ is just an instance of the previous one, the definition of map and reduce stay the same and all the theorems stated so far remain valid. For our immediate purposes we only need to update the computation rule:

**Theorem 16.47 (Computation Rule)** In addition to the computation rules given in theorem 16.34 we have:

(a) $\varpi R \circ \square \; = \; \square$
(b) $\varpi R \circ +\!\!+ \; = \; +\!\!+ \circ J.\varpi R$
(c) $/(S \triangledown T) \circ \square \; = \; S \circ 1\!\!1$
(d) $/(S \triangledown T) \circ +\!\!+ \; = \; T \circ J./(S \triangledown T)$

**Proof** There are two pairs of computation rules given in the theorem but by using junc cancellation (theorem 12.80(a)) we can derive the elements of each pair simultaneously. We illustrate the method on the second pair:

$$
\begin{array}{ll}
 & (/(S \triangledown T) \circ \square) \;\triangledown\; (/(S \triangledown T) \circ +\!\!+) \\
= & \quad \{ \text{ spec-junc fusion: theorem 12.68(a) } \} \\
 & /(S \triangledown T) \circ \square \triangledown +\!\!+ \\
= & \quad \{ \text{ (16.46) } \} \\
 & /(S \triangledown T) \circ \eta \\
= & \quad \{ \text{ computation rule: theorem 16.34(d) } \} \\
 & S \triangledown T \;\circ\; 1\!\!1 + J./(S \triangledown T) \\
= & \quad \{ \text{ junc-sum fusion: theorem 12.66(a) } \} \\
 & (S \circ 1\!\!1) \;\triangledown\; (T \circ J./(S \triangledown T))
\end{array}
$$

We have thus proved the equality of two juncs. Rules (c) and (d) now follow by the junc cancellation theorem. The first pair is derived similarly.
□

### 16.6.1 Definition of Filters

The definition of filter is borrowed directly from the work of Meertens [68] and Bird [19]:

**Definition 16.48 (Filter)** For right-condition $p$,

$$\lhd p \;=\; /\eta \circ \varpi(\tau \lhd p \rhd (\square \circ \top\!\top))$$
□

Note that from the fact that $\tau$ and $\square \circ \top\!\top$ are imps and the fact that conditionals, junc and catamorphism respect imps it follows that $\lhd p$ is an imp.

In this section we explore several algebraic properties of the filter operation. The properties that we seek are motivated by the relationship between the Bird-Meertens formalism and the so-called quantifier calculus, which relationship will be clarified in the next section.

By design $\triangleleft true$ is the identity function on specs of the correct type:

**Theorem 16.49**

$$\triangleleft true \quad = \quad \varpi I$$

**Proof**

$$
\begin{array}{ll}
 & \triangleleft true \\
= & \qquad \{ \text{ definition 16.48 } \} \\
 & /\eta \circ \varpi(\tau \triangleleft true \triangleright (\Box \circ \top\!\top)) \\
= & \qquad \{ \text{ conditionals: theorem 15.18(a) } \} \\
 & /\eta \circ \varpi\tau \\
= & \qquad \{ \text{ identity rule: theorem 16.37 } \} \\
 & \varpi I
\end{array}
$$

□

Now we consider whether two filters can be fused into one. Since $\triangleleft p$ is a catamorphism of the form $/\eta \circ \varpi\overline{p}$ where $\overline{p} = \tau \triangleleft p \triangleright (\Box \circ \top\!\top)$ it pays to begin by exploring whether a map can be fused with a filter. Indeed it can.

**Lemma 16.50**

(a) $\varpi R \circ \triangleleft p \;=\; /\eta \circ \varpi((\tau \circ R) \triangleleft p \triangleright (\Box \circ \top\!\top))$

(b) $/\eta \circ \varpi R \circ \triangleleft p \;=\; /\eta \circ \varpi(R \triangleleft p \triangleright (\Box \circ \top\!\top))$

**Proof** For brevity let $\overline{p}$ denote $\tau \triangleleft p \triangleright (\Box \circ \top\!\top)$. Then we prove part (a) as follows:

$$
\begin{array}{ll}
 & \varpi R \circ \triangleleft p \\
= & \qquad \{ \text{ definition 16.48 } \} \\
 & \varpi R \circ /\eta \circ \varpi\overline{p} \\
= & \qquad \{ \; /\eta \text{ leapfrog: theorem 16.38(b) } \} \\
 & /\eta \circ \varpi\varpi R \circ \varpi\overline{p}
\end{array}
$$

$$
\begin{array}{ll}
= & \{\ \varpi \text{ is a relator, definition of } \overline{p}\ \} \\
& /\eta \circ \varpi(\varpi R \circ \tau \lhd p \rhd (\Box \circ \top\!\!\top)) \\
= & \{\ \text{conditionals: theorem 15.18(n)}\ \} \\
& /\eta \circ \varpi((\varpi R \circ \tau) \lhd p \rhd (\varpi R \circ \Box \circ \top\!\!\top)) \\
= & \{\ \text{computation rule: theorem 16.47(a)}\ \} \\
& /\eta \circ \varpi((\tau \circ R) \lhd p \rhd (\Box \circ \top\!\!\top))
\end{array}
$$

Part (b) is derived from (a) using the leapfrog rule, theorem 16.38(a), followed by theorem 15.18(n) and the computation rule 16.47(c).

□

A direct consequence of lemma 16.50 is:

**Theorem 16.51 ($\lhd$ distribution)**

$$\lhd p \ \circ\ \lhd q \ = \ \lhd (p \wedge q)$$

**Proof**

$$
\begin{array}{ll}
& \lhd p \ \circ\ \lhd q \\
= & \{\ \text{definition 16.48, lemma 16.50(b)}\ \} \\
& /\eta \circ \varpi((\tau \lhd p \rhd (\Box \circ \top\!\!\top)) \lhd q \rhd (\Box \circ \top\!\!\top)) \\
= & \{\ \text{conditionals: theorem 15.18(f)}\ \} \\
& /\eta \circ \varpi(\tau \lhd (p \wedge q) \rhd (\Box \circ \top\!\!\top)) \\
= & \{\ \text{definition 16.48}\ \} \\
& \lhd (p \wedge q)
\end{array}
$$

□

Yet another fusion property for filters is

**Theorem 16.52 (Filter Translation)** For all imps $f$

$$\lhd p \circ \varpi f \ = \ \varpi f \ \circ\ \lhd (p \circ f) \ \circ\ \varpi f{>}$$

**Proof**

$$
\begin{array}{ll}
& \varpi f \ \circ\ \lhd (p \circ f) \ \circ\ \varpi f{>} \\
= & \{\ \text{lemma 16.50(a)}\ \} \\
& /\eta \circ \varpi((\tau \circ f) \lhd (p \circ f) \rhd (\Box \circ \top\!\!\top)) \ \circ\ \varpi f{>} \\
= & \{\ \text{relator.}\varpi\ \} \\
& /\eta \circ \varpi((\tau \circ f) \lhd (p \circ f) \rhd (\Box \circ \top\!\!\top) \ \circ\ f{>})
\end{array}
$$

$$
\begin{array}{ll}
= & \{\ \text{imp.}(f{>}),\ \text{conditionals: theorem 15.18(o)}\ \} \\
& /\eta \circ \varpi((\tau \circ f \circ f{>}) \lhd (p \circ f \circ f{>}) \rhd (\Box \circ \top\!\!\top \circ f{>})) \\
= & \{\ \text{domains: (10.25) and } \top\!\!\top \circ f{>} \;=\; \top\!\!\top \circ f \text{ : (10.20)}\ \} \\
& /\eta \circ \varpi((\tau \circ f) \lhd (p \circ f) \rhd (\Box \circ \top\!\!\top \circ f)) \\
= & \{\ \bullet\ \ \text{imp.}f,\ \text{conditionals: theorem 15.18(o)}\ \} \\
& /\eta \circ \varpi(\tau \lhd p \rhd (\Box \circ \top\!\!\top) \circ f) \\
= & \{\ \text{relator.}\varpi\ \} \\
& /\eta \circ \varpi(\tau \lhd p \rhd (\Box \circ \top\!\!\top)) \circ \varpi f \\
= & \{\ \text{definition 16.48}\ \} \\
& \lhd p \circ \varpi f
\end{array}
$$

□

Theorem 16.52 can also be strengthened in the same way that theorem 15.18(o) was strengthened to theorem 15.18(p).

The syntactic resemblance of theorems 15.8 and 16.52 should not go unnoticed. After some thought the resemblance is not surprising: $p{>}$ is a sort of filter but on elements of some base set, $\lhd p$ is the same filter but "lifted" to elements of $\varpi I$. (By the way, Meertens [68] included both laws but they are somewhat hidden in the text.)

## Acknowledgement

We would like to thank Peter de Bruin, Henk Doornbos, Netty van Gasteren, Rik van Geldrop, Grant Malcolm, Asia van Mortel-Fronczak, Frans Rietman and Martin Simons for their various contributions to this work.

Preparation of this paper was expedited by the use of the proof editor developed by Paul Chisholm [28].

# Chapter 17

# Solutions to Exercises

**Solution to exercise: 3.27**
**a**: We derive for any $y$

$$
\begin{array}{ll}
 & y \sqsubseteq \sqcap.(x:\ x \in S:\ f.x)\ \sqcap\ \sqcap.(x:\ x \in S:\ g.x) \\
\equiv & \qquad \{\ \text{definition of infimum and (3.22)}\ \} \\
 & \forall(x:\ x \in S:\ y \sqsubseteq f.x)\ \ \wedge\ \ \forall(x:\ x \in S:\ y \sqsubseteq g.x) \\
\equiv & \qquad \{\ \text{distribution of } \forall \text{ over } \wedge\ \} \\
 & \forall(x:\ x \in S:\ y \sqsubseteq f.x\ \wedge\ y \sqsubseteq g.x) \\
\equiv & \qquad \{\ \text{(3.22) and definition of infimum}\ \} \\
 & y \sqsubseteq \sqcap.(x:\ x \in S:\ f.x \sqcap g.x) \quad .
\end{array}
$$

**b**: For any $y$ we derive

$$
\begin{array}{ll}
 & y \sqsubseteq a\ \sqcap\ \sqcap.S \\
\equiv & \qquad \{\ \text{(3.22) and definition of infimum}\ \} \\
 & y \sqsubseteq a\ \wedge\ \forall(x:\ x \in S:\ y \sqsubseteq x) \\
\equiv & \qquad \{\ S \neq \emptyset \text{ hence distribution of } \wedge \text{ over } \forall \text{ allowed}\ \} \\
 & \forall(x:\ x \in S:\ y \sqsubseteq a\ \wedge\ y \sqsubseteq x) \\
\equiv & \qquad \{\ \text{(3.22) and definition of infimum}\ \} \\
 & y \sqsubseteq \sqcap.(x:\ x \in S:\ a \sqcap x) \quad .
\end{array}
$$

**c**: For any $y$ we observe

192z
$$
\begin{array}{ll}
 & y \sqsubseteq \sqcap.(x:\ x \in S:\ \top\!\top) \\
\equiv & \qquad \{\ \text{definition of infimum}\ \}
\end{array}
$$

$$\begin{array}{ll} & \forall(x:\ x \in S:\ y \sqsubseteq \top\!\top) \\ \equiv & \quad \{\ \text{term } \mathit{true}\ \} \\ & \mathit{true} \quad . \end{array}$$

□

**Solution to exercise: 3.28**

**a**: Let $R$ be reflexive and anti-symmetric, for any $x$ and $y$ we have

$$\begin{array}{ll} & x = y \\ \Rightarrow & \quad \{\ \text{Leibniz}\ \} \\ & \forall(z ::\ zRx \equiv zRy) \\ \Rightarrow & \quad \{\ \text{Instantiate } z := x \text{ and } z := y\ \} \\ & xRx \equiv xRy \quad \wedge \quad yRx \equiv yRy \\ \equiv & \quad \{\ R \text{ is reflexive}\ \} \\ & xRy \wedge yRx \\ \Rightarrow & \quad \{\ R \text{ is anti-symmetric}\ \} \\ & x = y \quad . \end{array}$$

**b**:

($\Rightarrow$) Assume that $R$ is reflexive and transitive. For any $x$ and $y$ we derive

$$\begin{array}{ll} & xRy \\ \Rightarrow & \quad \{\ R \text{ is transitive}\ \} \\ & \forall(z ::\ zRx \Rightarrow zRy) \\ \Rightarrow & \quad \{\ \text{instantiate } z := x\ \} \\ & xRx \Rightarrow xRy \\ \equiv & \quad \{\ R \text{ is reflexive}\ \} \\ & xRy \quad . \end{array}$$

($\Leftarrow$) Assume $xRy \equiv \forall(z ::\ zRx \Rightarrow zRy)$ holds for all $x$ and $y$. For reflexivity of $R$ we have

$$\begin{array}{ll} & xRx \\ \equiv & \quad \{\ \text{assumption}\ \} \\ & \forall(z ::\ zRx \Rightarrow zRx) \\ \equiv & \quad \{\ \text{term } \mathit{true}\ \} \\ & \mathit{true} \quad . \end{array}$$

For the transitivity of $R$ we derive

$$
\begin{array}{ll}
 & xRv \ \wedge \ vRy \\
\equiv & \quad \{ \text{ assumption } \} \\
 & \forall(z :: \ zRx \ \Rightarrow \ zRv) \ \wedge \ \forall(z :: \ zRv \ \Rightarrow \ zRy) \\
\equiv & \quad \{ \text{ distribution of } \forall \text{ over } \wedge \ \} \\
 & \forall(z :: \ zRx \ \Rightarrow \ zRv \ \wedge \ zRv \ \Rightarrow \ zRy) \\
\Rightarrow & \quad \{ \ \Rightarrow \text{ is transitive } \} \\
 & \forall(z :: \ zRx \ \Rightarrow \ zRy) \\
\Rightarrow & \quad \{ \text{ assumption } \} \\
 & xRy \quad .
\end{array}
$$

**c**: In view of part **a** and **b**, it only remains to prove that $\forall(z :: \ zRx \ \equiv \ zRy) \ \equiv \ x = y$ and $xRy \ \equiv \ \forall(z :: \ zRx \ \Rightarrow \ zRy)$, for all $x$ and $y$, implies that $R$ is anti-symmetric.

$$
\begin{array}{ll}
 & xRy \ \wedge \ yRx \\
\equiv & \quad \{ \text{ assumption } \} \\
 & \forall(z :: \ zRx \ \Rightarrow \ zRy) \ \wedge \ \forall(z :: \ zRy \ \Rightarrow \ zRx) \\
\equiv & \quad \{ \text{ distribution of } \forall \text{ over } \wedge \text{, calculus } \} \\
 & \forall(z :: \ zRx \ \equiv \ zRy) \\
\equiv & \quad \{ \text{ assumption } \} \\
 & x = y \quad .
\end{array}
$$

□

**Solution to exercise: 3.46**

**a**: First observe that $x \sqsubseteq x \sqcup y$, since for any $z$ we have

$$
\begin{array}{ll}
 & x \sqsubseteq z \ \ \Leftarrow \ \ x \sqcup y \sqsubseteq z \\
\equiv & \quad \{ \text{ definition supremum } \} \\
 & x \sqsubseteq z \ \ \Leftarrow \ \ x \sqsubseteq z \ \wedge \ y \sqsubseteq z \\
\equiv & \quad \{ \text{ elementary predicate calculus } \} \\
 & \mathit{true} \quad .
\end{array}
$$

Hence, by duality $x \sqcap y \sqsubseteq x$ .

$$
\begin{array}{ll}
 & x \sqcup (x \sqcap y) \ = \ x \\
\equiv & \quad \{ \ x \sqcup y \sqsupseteq x \text{ for arbitrary } y \ \} \\
 & x \sqcup (x \sqcap y) \sqsubseteq x \\
\equiv & \quad \{ \text{ definition supremum } \}
\end{array}
$$

$$
\begin{array}{ll}
 & x \sqsubseteq x \quad \wedge \quad x \sqcap y \sqsubseteq x \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \mathit{true} \quad .
\end{array}
$$

**b**: Note that $\sqcap.(y : \ y \sqsupseteq S : \ y) = \sqcap.\hat{S}$ where $\hat{S}$ is defined by $y \in \hat{S} \equiv y \sqsupseteq S$. We show that $\sqcup.S$ solves

$$
x :: \qquad z \sqsubseteq x \quad \equiv \quad z \sqsubseteq \hat{S}.
$$

$$
\begin{array}{ll}
 & z \sqsubseteq \sqcup.S \\
\equiv & \qquad \{ \text{ indirect equality } \} \\
 & \forall(y : \ \sqcup.S \sqsubseteq y : \ z \sqsubseteq y) \\
\equiv & \qquad \{ \text{ suprema } \} \\
 & \forall(y : \ S \sqsubseteq y : \ z \sqsubseteq y) \\
\equiv & \qquad \{ \text{ definition of } \hat{S} \ \} \\
 & \forall(y : \ y \in \hat{S} : \ z \sqsubseteq y) \\
\equiv & \qquad \{ \text{ definition of “below” } \} \\
 & z \sqsubseteq \hat{S} \quad .
\end{array}
$$

□

**Solution to exercise: 3.51**

Let $S$ be a non-empty, finite set. From property 3.50(**a**) we observe that it is sufficient to prove $\sqcap.S \in S$ . The proof is by induction on the cardinality of $S$. For $S$ a one-element set, the result follows from (3.15). If $|S| > 1$ we can choose two subsets of $S$, $X$ and $Y$, both non-empty and stricly contained in $S$ such that $S = X \cup Y$ . By induction we have $\sqcap.X \in X$ and $\sqcap.Y \in Y$ . Since $\sqsubseteq$ is total we assume, without loss of generality, $\sqcap.X \sqsubseteq \sqcap.Y$ . Hence

$$
\begin{array}{ll}
 & \sqcap.S \\
= & \qquad \{ \text{ (3.16) range disjunction } \} \\
 & (\sqcap.X) \sqcap (\sqcap.Y) \\
= & \qquad \{ \text{ (3.26) since } \sqcap.X \sqsubseteq \sqcap.Y \ \} \\
 & \sqcap.X \\
\in & \qquad \{ \ X \subseteq S \ \} \\
 & S \quad .
\end{array}
$$

□

**Solution to exercise: 3.52**
For $S$ and $T$ subsets of $\mathcal{A}$ we derive

$$\begin{array}{ll} & \sqcap.S \sqsubseteq \sqcap.T \\ \equiv & \quad \{ (3.26) \} \\ & \sqcap.S \;=\; (\sqcap.S) \sqcap (\sqcap.T) \\ \equiv & \quad \{ \text{range disjunction: } (3.16) \} \\ & \sqcap.S \;=\; \sqcap.(S \cup T) \\ \Leftarrow & \quad \{ \text{Leibniz} \} \\ & S \;=\; S \cup T \\ \equiv & \quad \{ \text{set calculus} \} \\ & S \supseteq T \quad . \end{array}$$

As a dual we have

$$\sqcup.S \sqsubseteq \sqcup.T \quad \Leftarrow \quad S \subseteq T \quad .$$

□

**Solution to exercise: 4.10**
The proof is by mutual implication.
$\Leftarrow$: Assume $\forall(S : \underline{\mathbf{min}}.S \text{ exists} : \; f.\underline{\mathbf{min}}.S = \underline{\mathbf{min}}.f.S)$ . Let $x \sqsubseteq y$, hence $x = \underline{\mathbf{min}}.\{x, y\}$ . From the assumption it follows $f.x = \underline{\mathbf{min}}.\{f.x, f.y\}$, from which we deduce $f.x \sqsubseteq f.y$ .
$\Rightarrow$: Assume $\underline{\mathbf{min}}.S$ exists. We prove $f.\underline{\mathbf{min}}.S$ satisfies the definition of $\underline{\mathbf{min}}.f.S$ . First we observe $f.\underline{\mathbf{min}}.S \in f.S$ , since $\underline{\mathbf{min}}.S \in S$. It remains to prove $\forall(s : \; s \in S : \; f.\underline{\mathbf{min}}.S \sqsubseteq f.s)$ . Since $f$ is monotonic, this is implied by $\underline{\mathbf{min}}.S \sqsubseteq s$ for all $s \in S$; which is trivially true.
□

**Solution to exercise: 4.15**
For any $X \subseteq \mathcal{B}$ we derive

$$\begin{array}{ll} & \sqcap.f.X \;\sqsubseteq\; \sqcap.g.X \\ \equiv & \quad \{ \text{ characterisation of infimum: } (3.10) \; \} \\ & \forall(x : \; x \in X : \; \sqcap.f.X \;\sqsubseteq\; g.x) \\ \Leftarrow & \quad \{ \; \sqcap.f.X \;\sqsubseteq\; f.x \;\text{, transitivity} \; \} \\ & \forall(x : \; x \in X : \; f.x \;\sqsubseteq\; g.x) \quad . \end{array}$$

The dual property is

$$\sqcup.f.X \;\dot{\sqsupseteq}\; \sqcup.g.X \quad \Leftarrow \quad f \;\dot{\sqsupseteq}\; g \quad .$$

□

**Solution to exercise: 4.19**

First the easy part. Assume $(f\bullet)$ is univerally $\dot{\sqcup}$-junctive. Take an arbitrary $X \subseteq \mathcal{A}$ and define the set of endofunctions $\widehat{X}$ by $\widehat{x} \in \widehat{X} \equiv x \in X$. Since $(f\bullet)$ is univerally $\dot{\sqcup}$-junctive we obtain $(f\bullet).\dot{\sqcup}.\widehat{X} = \dot{\sqcup}.((f\bullet).\widehat{X})$ . Applying both functions to an arbitrary $y \in \mathcal{A}$, we obtain $f.\sqcup.X = \sqcup(f.X)$ .

For the other part, assume $f$ is universally $\sqcup$-junctive. Let $G \subseteq \mathcal{A} \longleftarrow \mathcal{A}$, then for arbitrary $x \in \mathcal{A}$ we have

$$\begin{array}{ll}
 & ((f\bullet).(\dot{\sqcup}.G)).x \\
\equiv & \qquad \{ \text{ definition of } (f\bullet) \text{ and } \dot{\sqcup} \ \} \\
 & f.\sqcup.(g:\ g \in G:\ g.x) \\
\equiv & \qquad \{ \ f \text{ is universally } \sqcup\text{-junctive } \} \\
 & \sqcup.(g:\ g \in G:\ f.g.x) \\
\equiv & \qquad \{ \text{ definition of } \dot{\sqcup} \ \} \\
 & (\dot{\sqcup}.(f \bullet G)).x \\
\equiv & \qquad \{ \text{ definition of } (f\bullet) \ \} \\
 & (\dot{\sqcup}.((f\bullet).G)).x \quad .
\end{array}$$

□

**Solution to exercise: 5.40**

We first prove the equivalence between **a** and **b**. The rest is proven by cyclic implication. Assume $(F, G)$ is a Galois connection.

$$\begin{array}{ll}
 & \forall(x:\ x \in \mathcal{B}:\ F.x = \underline{\mathbf{min}}.(y:\ x = G.y:\ y)) \\
\equiv & \qquad \{ \text{ definition } \underline{\mathbf{min}} \ \} \\
 & \forall(x:\ x \in \mathcal{B}:\ x = G.F.x \wedge \forall(y:\ x = G.y:\ F.x \sqsubseteq y)) \\
\equiv & \qquad \{ \ (F,G) \text{ is a Galois connection } \} \\
 & \forall(x:\ x \in \mathcal{B}:\ G.F.x = x) \quad .
\end{array}$$

We now prove **b**⇒ **c**⇒ **d**⇒**e**⇒**b**.

$$
\begin{array}{ll}
 & \forall(x:\ x \in \mathcal{B}:\ G.F.x = x) \\
\equiv & \qquad \{\ \text{theorem } 5.20(\mathbf{a})\ \} \\
 & \forall(x:\ x \in \mathcal{B}:\ x \in G.\mathcal{A}) \\
\equiv & \qquad \{\ \text{calculus}\ \} \\
 & \mathcal{B} = G.\mathcal{A} \\
\Rightarrow & \qquad \{\ \text{theorem } 5.16(\mathbf{a}),\ \text{restriction to } G.\mathcal{A} \text{ is vacuous}\ \} \\
 & F \text{ is a poset-monomorphism} \\
\Rightarrow & \qquad \{\ \text{calculus}\ \} \\
 & F \text{ is injective} \\
\Rightarrow & \qquad \{\ \text{definition of injective}\ \} \\
 & \forall(x:\ x \in \mathcal{B}:\ F.G.F.x = F.x \equiv G.F.x = x) \\
\equiv & \qquad \{\ \text{semi-inverse}\ \} \\
 & \forall(x:\ x \in \mathcal{B}:\ G.F.x = x) \quad .
\end{array}
$$

The fact that any of the clauses **a** through **e** implies $F.x = \sqcap.(y:\ x = G.y:\ y)$ for all $x \in \mathcal{B}$ is trivial.
The dual theorem is:
For $(F, G)$ a Galois connection the following are equivalent

**a** $\forall(y:\ y \in \mathcal{A}:\ G.y = \underline{\mathbf{max}}.(x:\ F.x = y:\ x))$ ,

**b** $\forall(y:\ y \in \mathcal{A}:\ F.G.y = y)$ ,

**c** $F$ is surjective,

**d** $G$ is a poset-monomorphism,

**e** $G$ is injective.

And any one of the above implies

- $\forall(y:\ y \in \mathcal{A}:\ G.y = \sqcup.(x:\ F.x = y:\ x))$ .

□

**Solution to exercise: 6.8**
The proof is by cyclic implication. **a**⇒**b**

$$
\begin{array}{ll}
 & \mathbf{b} \\
\equiv & \qquad \{ \text{ suprema and infima } \} \\
 & x \sqcap (y \sqcup z) \sqsubseteq (x \sqcap y) \sqcup (x \sqcap z) \\
\Leftarrow & \qquad \{ \ \mathbf{a} \text{ with } z := x \sqcap z \ \} \\
 & x \sqcap (y \sqcup z) \sqsubseteq x \sqcap (y \sqcup (x \sqcap z)) \\
\equiv & \qquad \{ \text{ infima, } x \sqcap (y \sqcup z) \sqsubseteq x \ \} \\
 & x \sqcap (y \sqcup z) \sqsubseteq y \sqcup (x \sqcap z) \\
\Leftarrow & \qquad \{ \ \mathbf{a} \text{ with } y,z \ := \ z,y \ \} \\
 & x \sqcap (y \sqcup z) \sqsubseteq x \sqcap (z \sqcup y) \\
\equiv & \qquad \{ \text{ supremum is commutative } \} \\
 & \mathit{true} \quad .
\end{array}
$$

**b⇒c**

$$
\begin{array}{ll}
 & (x \sqcup y) \sqcap (x \sqcup z) \\
= & \qquad \{ \ \mathbf{b} \text{ with } x,y \ := x \sqcup y, x \ \} \\
 & ((x \sqcup y) \sqcap x) \sqcup ((x \sqcup y) \sqcap z) \\
= & \qquad \{ \ ((x \sqcup y) \sqcap x) \ = \ x; \ \mathbf{b} \text{ with } x,z \ := z,x \ \} \\
 & x \sqcup (x \sqcap z) \sqcup (y \sqcap z) \\
= & \qquad \{ \ x \sqcup (y \sqcap z) \ = \ x \ \} \\
 & x \sqcup (y \sqcap z) \quad .
\end{array}
$$

**c⇒d**

$$
\begin{array}{ll}
 & (x \sqcap y) \sqcup (y \sqcap z) \sqcup (z \sqcap x) \\
= & \qquad \{ \ \mathbf{c} \text{ with } x \ := \ x \sqcap y \ \} \\
 & (((x \sqcap y) \sqcup y) \sqcap ((x \sqcap y) \sqcup z)) \sqcup (z \sqcap x) \\
= & \qquad \{ \ ((x \sqcap y) \sqcup y) \ = \ y \ \} \\
 & (y \sqcap ((x \sqcap y) \sqcup z)) \sqcup (z \sqcap x) \\
= & \qquad \{ \ \mathbf{c} \text{ with } x,z \ := \ z \sqcap x, (x \sqcap y) \sqcup z \ \} \\
 & (y \sqcup (z \sqcap x)) \sqcap ((x \sqcap y) \sqcup z \sqcup (z \sqcap x)) \\
= & \qquad \{ \ z \sqcup (z \sqcap x \ = \ z \ \} \\
 & (y \sqcup (z \sqcap x)) \sqcap ((x \sqcap y) \sqcup z) \\
= & \qquad \{ \ \mathbf{c} \text{ twice, calculus } \} \\
 & (x \sqcup y) \sqcap (y \sqcup z) \sqcap (z \sqcup x) \quad .
\end{array}
$$

**d⇒a**
Instantiate in **d** $x,\ y,\ z\ :=\ (x \sqcap y) \sqcup (x \sqcap z),\ y \sqcap z,\ x$ . Then the right-hand side of **d** becomes

$$
\begin{array}{ll}
 & ((x \sqcap y) \sqcup (x \sqcap z) \sqcup (y \sqcap z)) \sqcap \\
 & \ ((y \sqcap z) \sqcup x) \sqcap (x \sqcup (x \sqcap y) \sqcup (x \sqcap z)) \\
= & \qquad \{\ \mathbf{d};\ x \sqcup (x \sqcap w) \ = \ x, \text{twice}\ \} \\
 & ((x \sqcup y) \sqcap (x \sqcup z) \sqcap (y \sqcup z)) \sqcap ((y \sqcap z) \sqcup x) \sqcap x \\
= & \qquad \{\ x \sqcap (x \sqcup w) \ = \ x\ \} \\
 & (y \sqcup z) \sqcap x \quad ,
\end{array}
$$

and the left-hand side of **d** becomes

$$
\begin{array}{ll}
 & (((x \sqcap y) \sqcup (x \sqcap z)) \sqcap y \sqcap z) \sqcup \\
 & ((y \sqcap z) \sqcap x) \sqcup (((x \sqcap y) \sqcup (x \sqcap z)) \sqcap x) \\
= & \qquad \{\ \text{infima, suprema}\ \} \\
 & (x \sqcap y \sqcap z) \sqcup ((x \sqcap y) \sqcup (x \sqcap z)) \\
\sqsubseteq & \qquad \{\ \text{infima, suprema}\ \} \\
 & (x \sqcap y) \sqcup z \quad .
\end{array}
$$

□

**Solution to exercise: 6.9**
The proof is by mutual implication.
First assume distributivity, then

$$
\begin{array}{ll}
 & x \sqcap z \sqsubseteq y \ \wedge \ x \sqsubseteq z \sqcup y \\
\equiv & \qquad \{\ (3.40) \text{ and } (3.26)\ \} \\
 & y \ = \ (x \sqcap z) \sqcup y \ \wedge \ x \ = \ x \sqcap (z \sqcup y) \\
\Rightarrow & \qquad \{\ \text{exercise } 6.8(\mathbf{a})\ \} \\
 & x \sqsubseteq y \quad .
\end{array}
$$

For the other part, assume $x \sqcap z \sqsubseteq y \wedge x \sqsubseteq z \sqcup y \Rightarrow x \sqsubseteq y$ holds. From exercise 6.8(**a**) it is sufficient to prove $x \sqcap (y \sqcup z) \sqsubseteq (x \sqcap y) \sqcup z$

$$
\begin{array}{ll}
 & x \sqcap (y \sqcup z) \sqsubseteq (x \sqcap y) \sqcup z \\
\Leftarrow & \qquad \{\ \text{assumption}, \bullet \quad \text{some } w\ \} \\
 & x \sqcap (y \sqcup z) \sqcap w \sqsubseteq (x \sqcap y) \sqcup z \\
 & \ \wedge \ x \sqcap (y \sqcup z) \sqsubseteq w \sqcup (x \sqcap y) \sqcup z \\
\Leftarrow & \qquad \{\ \text{choose } w \ := \ y\ \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

**Solution to exercise: 6.10**
With the notation as in the hint:
Since $\sqcup.(x \sqcap \mathcal{S}) \sqsubseteq x \sqcap (\sqcup.\mathcal{S})$ , it suffices to prove for every ordinal $\beta$ that $x \sqcap \Sigma_\beta \sqsubseteq \sqcup.(x \sqcap \mathcal{S})$ . By transfinite induction: for the step we have

$$
\begin{array}{ll}
 & x \sqcap \Sigma_{\beta+1} \\
= & \quad \{ \text{ definition } \Sigma \ \} \\
 & x \sqcap (\Sigma_\beta \sqcup S_\beta) \\
= & \quad \{ \text{ distributiion } \} \\
 & (x \sqcap \Sigma_\beta) \sqcup (x \sqcap S_\beta) \\
\sqsubseteq & \quad \{ \text{ induction, calculus } \} \\
 & \sqcup.(x \sqcup \mathcal{S}) \quad .
\end{array}
$$

For the limit case we observe

$$
\begin{array}{ll}
 & x \sqcap \Sigma_\beta \\
= & \quad \{ \ \beta \text{ is a limit ordinal } \} \\
 & x \sqcap \sqcup.(\alpha :\ \alpha < \beta :\ S_\alpha) \\
= & \quad \{ \ S_\alpha \text{ forms a chain, chain distribution } \} \\
 & \sqcup.(\alpha :\ \alpha < \beta :\ x \sqcap S_\alpha) \\
\sqsubseteq & \quad \{ \text{ induction } \} \\
 & \sqcup.(\alpha :\ \alpha < \beta :\ x \sqcap \mathcal{S}) \\
= & \quad \{ \text{ calculus } \} \\
 & \sqcup.(x \sqcap \mathcal{S}) \quad .
\end{array}
$$

□

**Solution to exercise: 6.30**
Let $x'$ and $x''$ be complements of $x$. By symmetry the following suffices

$$
\begin{array}{ll}
 & x' \\
= & \quad \{ \ x \sqcup x'' = \top\!\top \ \} \\
 & x' \sqcap (x \sqcup x'') \\
\sqsubseteq & \quad \{ \text{ exercise 6.8(}\mathbf{a}\text{) } \} \\
 & (x' \sqcap x) \sqcup x'' \\
= & \quad \{ \ x' \sqcap x = \bot\!\bot \ \} \\
 & x'' \quad .
\end{array}
$$

□

**Solution to exercise: 6.31**
Just for fun we calculate the upper adjoint of $n\downarrow$, so it exists.

$$\begin{array}{ll}
 & n\downarrow k \ \le\ l \\
\equiv & \qquad \{\ \text{supremum}\ \} \\
 & n \ \le\ l \ \vee\ k \ \le\ l \\
\equiv & \qquad \{\ \text{calculus}\ \} \\
 & \mathtt{if}\ \ n \ \le\ l \rightarrow \mathit{true} \ [\!]\ n \ >\ l \rightarrow k \ \le\ l \ \ \mathtt{fi} \\
\equiv & \qquad \{\ \mathit{true} \ \equiv\ k \ \le\ \infty\ \} \\
 & k \ \le\ \mathtt{if}\ \ n \ \le\ l \rightarrow \infty \ [\!]\ n \ >\ l \rightarrow l \ \ \mathtt{fi} \quad ,
\end{array}$$

hence $(n\downarrow)^\sharp.l \ =\ \mathtt{if}\ \ n \ \le\ l \rightarrow \infty \ [\!]\ n \ >\ l \rightarrow l \ \ \mathtt{fi}$ .
Similarly $(n\uparrow)^\flat.k \ =\ \mathtt{if}\ \ k \ \le\ n \rightarrow 0 \ [\!]\ k \ >\ n \rightarrow k \ \ \mathtt{fi}$ . So

$$\begin{array}{ll}
 & (n\downarrow)^\sharp.0 \\
= & \qquad \{\ \text{definition } (n\downarrow)^\sharp\ \} \\
 & \mathtt{if}\ \ n \ =\ 0 \rightarrow \infty \ [\!]\ n \ >\ 0 \rightarrow 0 \ \ \mathtt{fi} \\
\neq & \qquad \{\ \text{for } n \neq 0 \text{ and } n \neq \infty\ \} \\
 & \mathtt{if}\ \ n \ <\ \infty \rightarrow \infty \ [\!]\ n \ =\ 0 \rightarrow 0 \ \ \mathtt{fi} \\
= & \qquad \{\ \text{definition } (n\uparrow)^\flat\ \} \\
 & (n\uparrow)^\flat.\infty \quad .
\end{array}$$

□

**Solution to exercise: 6.32**
For any endofunction $f$ and a set $S$:

$$\begin{array}{ll}
 & f.\sqcup.S \ =\ \sqcup.f.S \\
\equiv & \qquad \{\ f \ =\ f^{\diamond\diamond}, \text{ definition conjugate}\ \} \\
 & \neg(f^\diamond.\neg(\sqcup.S)) \ =\ \sqcup.\neg(f^\diamond.(\neg S)) \\
\equiv & \qquad \{\ \text{calculus, de Morgan}\ \} \\
 & f^\diamond.\sqcap.\neg S \ =\ \sqcap.f^\diamond.\neg S \quad .
\end{array}$$

□

**Solution to exercise: 6.33**
Part **a** is trivial, since $y \ \sqsubseteq\ x \ \sqcup\ y$ and $x \ \sqcap\ y \ \sqsubseteq\ y$ .
For part **b**

$$
\begin{array}{ll}
 & (x\sqcap)^\sharp.y \\
\sqsubseteq & \qquad \{ \ z \sqsubseteq x \sqcup (x\sqcup)^\flat.z \text{ with } z \ := \ (x\sqcap)^\sharp.y \ \} \\
 & (x \sqcup (x\sqcup)^\flat.(x\sqcap)^\sharp.y) \sqcap (x\sqcap)^\sharp.y \\
\sqsubseteq & \qquad \{ \ \text{distributution, calculus} \ \} \\
 & (x \sqcap (x\sqcap)^\sharp.y) \sqcup (x\sqcup)^\flat.(x\sqcap)^\sharp.y \\
\sqsubseteq & \qquad \{ \ x \sqcap (x\sqcap)^\sharp.y \sqsubseteq y \ \} \\
 & y \sqcup (x\sqcup)^\flat.(x\sqcap)^\sharp.y \\
\sqsubseteq & \qquad \{ \ \mathbf{a} \text{ twice} \ \} \\
 & (x\sqcap)^\sharp.y \quad .
\end{array}
$$

Note: the first step generalises the first step in the proof of (6.15), introducing $(x\sqcup)^\flat$ . The third step generalises the third step in the proof of (6.15).
For **c** we derive for any $w$

$$
\begin{array}{ll}
 & w \sqsubseteq (x\sqcap)^\sharp.y \sqcap (z\sqcap)^\sharp.y \\
\equiv & \qquad \{ \ \text{infimum} \ \} \\
 & w \sqsubseteq (x\sqcap)^\sharp.y \ \wedge \ w \sqsubseteq (z\sqcap)^\sharp.y \\
\equiv & \qquad \{ \ \text{definition } \sharp \ \} \\
 & x \sqcap w \sqsubseteq y \ \wedge \ z \sqcap w \sqsubseteq y \\
\equiv & \qquad \{ \ \text{supremum, distributivity} \ \} \\
 & (x \sqcup z) \sqcap w \sqsubseteq y \\
\equiv & \qquad \{ \ \text{definition } \sharp \ \} \\
 & w \sqsubseteq ((x \sqcup z)\sqcap)^\sharp.y \quad .
\end{array}
$$

□

**Solution to exercise: 6.45**
□

**Solution to exercise: 6.46**
We first prove that **a** implies **b**. Assume $p \in \mathbb{A}$, then

$$
\begin{array}{ll}
 & p \sqsubseteq x \sqcup y \\
\equiv & \qquad \{ \ \text{calculus, distributivity} \ \} \\
 & (p \sqcap x) \sqcup (p \sqcap y) \ = \ p \\
\equiv & \qquad \{ \ p \in \mathbb{A} \text{ so } p \sqcap x \ = \ p \ \vee \ p \sqcap x \ = \ \perp\!\!\perp \ \} \\
 & p \sqcap x \ = \ \ \vee \ \perp\!\!\perp \sqcup (p \sqcap y) \ = \ p \\
\equiv & \qquad \{ \ \text{calculus} \ \} \\
 & p \sqsubseteq y \ \vee \ p \sqsubseteq y \quad .
\end{array}
$$

The equivalence of **b** of **c** is proven by mutual implication.
**b⇒c**

$$
\begin{array}{ll}
& p = x \sqcup y \\
\equiv & \qquad \{ \text{ part } \mathbf{b} \ \} \\
& p = x \sqcup y \ \wedge \ (p \sqsubseteq x \ \vee \ p \sqsubseteq y) \\
\Rightarrow & \qquad \{ \ p = x \sqcup y \text{ implies } x \sqsubseteq p \text{ and } y \sqsubseteq p\text{, calculus } \} \\
& p = x \ \vee \ p = y \qquad .
\end{array}
$$

**c⇒b**

$$
\begin{array}{ll}
& p \sqsubseteq x \sqcup y \\
\equiv & \qquad \{ \text{ calculus, distribution } \} \\
& p = (p \sqcap x) \sqcup (p \sqcap y) \\
\Rightarrow & \qquad \{ \text{ part } \mathbf{c} \ \} \\
& p = p \sqcap x \ \vee \ p = p \sqcap y \\
\equiv & \qquad \{ \text{ calculus } \} \\
& p \sqsubseteq x \ \vee \ p \sqsubseteq y \qquad .
\end{array}
$$

If we assume complementation, we can prove **b⇒c**. Assume **b** and $p \sqsubseteq x$, then

$$
\begin{array}{ll}
& \mathit{true} \\
\equiv & \qquad \{ \ x \sqcup \neg x = \top\!\!\top \sqsupseteq p\text{, part } \mathbf{b} \ \} \\
& p \sqsubseteq x \ \vee \ p \sqsubseteq \neg x \\
\Rightarrow & \qquad \{ \ x \sqsubseteq p \ \} \\
& x = p \ \vee \ x \sqsubseteq \neg x \\
\equiv & \qquad \{ \text{ calculus } \} \\
& x = p \ \vee \ x = \bot\!\!\bot \qquad .
\end{array}
$$

□

**Solution to exercise: 6.47**
We first establish the equivalence between part **a** and **b**.

$$
\begin{array}{ll}
& \mathcal{A} \text{ is saturated} \\
\equiv & \qquad \{ \text{ theorem 6.47 } \} \\
& \forall(x :: \ x = \sqcup.\mathbb{A} \sqcap x) \\
\Rightarrow & \qquad \{ \text{ instantiate } x := \top\!\!\top \ \} \\
& \top\!\!\top = \sqcup.\mathbb{A} \\
\Rightarrow & \qquad \{ \text{ Leibniz with } (\sqcap x) \text{ for every } x \in \mathcal{A} \ \} \\
& \forall(x :: \ x = \sqcup.\mathbb{A} \sqcap x) \qquad .
\end{array}
$$

Assume **a**. We prove **c** holds.

$$
\begin{array}{ll}
 & x \sqsubseteq y \\
\equiv & \quad \{ \text{ definition 6.39 } \} \\
 & \sqcup.(a :\ a \sqsubseteq x :\ a) \sqsubseteq \sqcup.(a :\ a \sqsubseteq y :\ a) \\
\Leftarrow & \quad \{ \text{ exercise 3.52 } \} \\
 & \forall(a :\ a \sqsubseteq x :\ a \sqsubseteq y) \\
\Leftarrow & \quad \{ \text{ transitivity } \} \\
 & x \sqsubseteq y \quad .
\end{array}
$$

Assume **c**. We show the validity of **b**.

$$
\begin{array}{ll}
 & \top\!\!\top \sqsubseteq \sqcup.\mathbb{A} \\
\equiv & \quad \{ \ \mathbf{c} \ \} \\
 & \forall(a :\ a \sqsubseteq \top\!\!\top :\ a \sqsubseteq \sqcup.\mathbb{A}) \\
\equiv & \quad \{ \ a \text{ ranges over } \mathbb{A} \ \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

**Solution to exercise: 7.5**
Let $f$ be reflexive, idempotent and monotonic.

$$
\begin{array}{ll}
 & g \mathrel{\dot\sqsubseteq} f \bullet h \\
\Rightarrow & \quad \{ \ f \text{ is monotonic } \} \\
 & f \bullet g \mathrel{\dot\sqsubseteq} f \bullet f \bullet h \\
\equiv & \quad \{ \ f \text{ is idempotent } \} \\
 & f \bullet g \mathrel{\dot\sqsubseteq} f \bullet h \\
\Rightarrow & \quad \{ \ f \text{ is reflexive } \} \\
 & g \mathrel{\dot\sqsubseteq} f \bullet h \quad .
\end{array}
$$

□

**Solution to exercise: 7.6**

Function $f$ is a closure operator over the poset $(\mathcal{A}, \sqsubseteq)$, hence $f \in \mathcal{A} \longleftarrow \mathcal{A}$. Take $\overline{\mathcal{A}}$ to be the set of closed elements, i.e. $\overline{\mathcal{A}} = f.\mathcal{A}$ . We construct a Galois connection between $\overline{\mathcal{A}}$ and $\mathcal{A}$. Let $F$ be $f$ but typed as $F \in \overline{\mathcal{A}} \longleftarrow \mathcal{A}$ . Take $G \in \mathcal{A} \longleftarrow \overline{\mathcal{A}}$ to be the inclusion, i.e. $G.x = x$ for all $x \in \overline{\mathcal{A}}$. Now observe that $G \bullet F = f$ and $F \bullet G$ is the identity on $\overline{\mathcal{A}}$. It is easy to check that $F$ and $G$ satisfy the clauses of theorem 5.30.

□

**Solution to exercise: 7.16**

By definition 7.10 $f^\star$ is an $f$-closure. It remains to prove that it is the least $f$-closure. Let $\varphi$ be an $f$-closure.

$$
\begin{array}{ll}
 & f^\star \mathrel{\dot\sqsubseteq} \varphi \\
\Leftarrow & \qquad \{ \text{ corollary 7.12}(\mathbf{e}) \text{ with } g,h := I_{\mathcal{A}},\varphi \ \} \\
 & I_{\mathcal{A}} \mathrel{\dot\sqsubseteq} \varphi \ \wedge\ f \bullet \varphi \mathrel{\dot\sqsubseteq} \varphi \\
\equiv & \qquad \{ \ \varphi \text{ is an } f\text{-closure } \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

**Solution to exercise: 7.17**

We prove the statement by mutual containment.

$$
\begin{array}{ll}
 & (f\bullet)^{\dot\star} \mathrel{\ddot\sqsubseteq} (f^\star)\bullet \\
\Leftarrow & \qquad \{ \text{ corollary 7.12}(\mathbf{e}) \ \} \\
 & \dot{I_{\mathcal{A}}} \mathrel{\ddot\sqsubseteq} (f^\star)\bullet \ \wedge\ (f\bullet) \bullet ((f^\star)\bullet) \mathrel{\ddot\sqsubseteq} (f^\star)\bullet \\
\Leftarrow & \qquad \{ \text{ corollary 7.12}(\mathbf{b}), \ \bullet \text{ is associative } \} \\
 & \dot{I_{\mathcal{A}}} \mathrel{\ddot\sqsubseteq} I_{\mathcal{A}}\bullet \ \wedge\ (f \bullet f^\star)\bullet \mathrel{\ddot\sqsubseteq} (f^\star)\bullet \\
\Leftarrow & \qquad \{ \text{ calculus, monotonicity of } \bullet \text{ and theorem 7.10}(\mathbf{a}) \ \} \\
 & \mathit{true} \quad .
\end{array}
$$

For the other containment we observe

$$
\begin{array}{ll}
 & (f^\star)\bullet \mathrel{\ddot\sqsubseteq} (f\bullet)^{\dot\star} \\
\equiv & \qquad \{ \text{ definition of lifting } \} \\
 & \forall(g :: \ f^\star \bullet g \mathrel{\dot\sqsubseteq} (f\bullet)^{\dot\star}.g) \quad .
\end{array}
$$

Hence, for $g$ an arbitrary endofunction we derive

$$
\begin{array}{ll}
 & f^\star \bullet g \mathrel{\dot\sqsubseteq} (f\bullet)^{\dot\star}.g \\
\equiv & \qquad \{ \ (\bullet g) \text{ has an upper adjoint } (\bullet g)^\sharp \ \} \\
 & f^\star \mathrel{\dot\sqsubseteq} (\bullet g)^\sharp.(f\bullet)^{\dot\star}.g \\
\Leftarrow & \qquad \{ \text{ corollary 7.12}(\mathbf{e}) \ \} \\
 & I_{\mathcal{A}} \mathrel{\dot\sqsubseteq} (\bullet g)^\sharp.(f\bullet)^{\dot\star}.g \ \wedge\ f \bullet (\bullet g)^\sharp.(f\bullet)^{\dot\star}.g \mathrel{\dot\sqsubseteq} (\bullet g)^\sharp.(f\bullet)^{\dot\star}.g \\
\equiv & \qquad \{ \text{ adjoints } \}
\end{array}
$$

$$
\begin{array}{ll}
 & g \;\dot{\sqsubseteq}\; (f\bullet)^{\star}.g \;\wedge\; f\bullet(\bullet g)^{\sharp}.(f\bullet)^{\star}.g\bullet g \;\dot{\sqsubseteq}\; (f\bullet)^{\star}.g \\
\Leftarrow & \quad \{ \text{ corollary 7.12}(\mathbf{b})\text{; cancellation: } (\bullet g).(\bullet g)^{\sharp}.h \;\dot{\sqsubseteq}\; h \;\} \\
 & f\bullet(f\bullet)^{\star}.g \;\dot{\sqsubseteq}\; (f\bullet)^{\star}.g \\
\equiv & \quad \{ \text{ theorem 7.10}(\mathbf{a}) \;\} \\
 & \mathit{true} \quad .
\end{array}
$$

□

**Solution to exercise: 7.22**

**d** We are obliged to prove two inclusions. The inclusion $g^{\star} \;\dot{\sqsubseteq}\; f^{\star}$ does not require induction: by the monotonicity of $^{\star}$ it suffices to show that $g \;\dot{\sqsubseteq}\; f$. Now, for all $x$, we have:

$$
\begin{array}{ll}
 & g.x \\
= & \quad \{ \text{ definition } \} \\
 & x \oplus x \\
= & \quad \{ \text{ calculus } \} \\
 & (y \;\mapsto\; x \oplus y).x \\
\dot{\sqsubseteq} & \quad \{ \text{ 7.12}(\mathbf{f}) \;\} \\
 & (y \;\mapsto\; x \oplus y)^{\star}.x \\
= & \quad \{ \text{ definition } \} \\
 & f.x \quad .
\end{array}
$$

Hence, abstracting from $x$, $g \;\dot{\sqsubseteq}\; f$ as required.

The other inclusion is the one requiring induction:

$$
\begin{array}{ll}
 & f^{\star}.x \;\sqsubseteq\; g^{\star}.x \\
\Leftarrow & \quad \{ \text{ 7.12}(\mathbf{e})\text{, (b) } \} \\
 & f.g^{\star}.x \;\sqsubseteq\; g^{\star}.x \\
\equiv & \quad \{ \text{ definition of } f \;\} \\
 & (y \;\mapsto\; g^{\star}.x \;\oplus\; y)^{\star}.g^{\star}.x \;\sqsubseteq\; g^{\star}.x \\
\Leftarrow & \quad \{ \text{ 7.12}(\mathbf{e}) \;\} \\
 & g^{\star}.x \;\oplus\; g^{\star}.x \;\sqsubseteq\; g^{\star}.x \\
\equiv & \quad \{ \text{ 7.10}(\mathbf{a})\text{, definition of } g \;\} \\
 & \mathit{true} \quad .
\end{array}
$$

**e** Again we begin with the inclusion not requiring induction.

$$
\begin{array}{ll}
& \mu f \\
= & \quad \{ \text{ definition of } f \ \} \\
& \mu(x \ \mapsto \ (y \ \mapsto \ x \oplus y)^\star.x) \\
\sqsupseteq & \quad \{ \ \mu \text{ and closure operators are monotonic, } \perp\!\!\perp \ \sqsubseteq \ x \ \} \\
& \mu(x \ \mapsto \ (y \ \mapsto \ x \oplus y)^\star.\perp\!\!\perp) \\
= & \quad \{ \text{ definition of } \mu \ \} \\
& \mu(x \ \mapsto \ \mu(y \ \mapsto \ x \oplus y)) \\
= & \quad \{ \text{ definition of } h \ \} \\
& \mu h \quad .
\end{array}
$$

Now for the other inclusion we first simplify the proof requirement:

$$
\begin{array}{ll}
& \mu f \ \sqsubseteq \ \mu h \\
\Leftarrow & \quad \{ \ 7.21 \ \} \\
& f.\mu h \ \sqsubseteq \ \mu h \\
\equiv & \quad \{ \text{ definition of } f \ \} \\
& (y \ \mapsto \ \mu h \ \oplus \ y)^\star.\mu h \ \sqsubseteq \ \mu h \\
\Leftarrow & \quad \{ \ 7.12(\mathbf{e}) \ \} \\
& \mu h \ \oplus \ \mu h \ \sqsubseteq \ \mu h \quad .
\end{array}
$$

But,

$$
\begin{array}{ll}
& \mu h \\
= & \quad \{ \ 7.20 \ \} \\
& h.\mu h \\
= & \quad \{ \text{ definition of } h \ \} \\
& \mu(y \ \mapsto \ \mu h \ \oplus \ y) \\
= & \quad \{ \ 7.20 \ \} \\
& \mu h \ \oplus \ \mu(y \ \mapsto \ \mu h \ \oplus \ y) \quad .
\end{array}
$$

Hence $\mu h \ = \ \mu h \oplus \mu h$ and the proof is complete.

**f** This last part is just a combination of the previous two. From **d** we have $f^\star \ = \ g^\star$, hence $\mu f \ = \ \mu g$ (since $\mu f \ = \ f^\star.\perp\!\!\perp$). Together with part **e** we have $\mu g \ = \ \mu h$ .

**Solution to exercise: 7.23**

The proof is by mutual containment.

$$
\begin{array}{ll}
 & \mu(f\bullet g)\ \dot{\sqsubseteq}\ f.\mu(g\bullet f) \\
\Leftarrow & \quad \{\ \mu(f\bullet g) \text{ is the least fixed point of } f\bullet g\ \} \\
 & f.g.f.\mu(g\bullet f)\ =\ f.\mu(g\bullet f) \\
\equiv & \quad \{\ \mu(g\bullet f) \text{ is a fixed point of } g\bullet f\ \} \\
 & \mathit{true} \quad .
\end{array}
$$

For the other containment we observe

$$
\begin{array}{ll}
 & f.\mu(g\bullet f)\ \dot{\sqsubseteq}\ \mu(f\bullet g) \\
\equiv & \quad \{\ \mu(f\bullet g) \text{ is a fixed point of } f\bullet g\ \} \\
 & f.\mu(g\bullet f)\ \dot{\sqsubseteq}\ f.g.\mu(f\bullet g) \\
\Leftarrow & \quad \{\ f \text{ is monotonic}\ \} \\
 & \mu(g\bullet f)\ \dot{\sqsubseteq}\ g.\mu(f\bullet g) \\
\equiv & \quad \{\ \text{see above, with } f \text{ and } g \text{ interchanged}\ \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

**Solution to exercise: 7.24**
For part **a**

$$
\begin{array}{ll}
 & \dot{\mu}(\widehat{f}) \\
= & \quad \{\ \text{definition fixpoint}\ \} \\
 & \widehat{f}.\dot{\mu}(\widehat{f}) \\
= & \quad \{\ \widehat{f}.g\ =\ f \text{ , for all } g\ \} \\
 & f \quad .
\end{array}
$$

For part **b** we observe that we have the following characterisation for $\dot{\mu}(f\bullet)$:

$i \qquad f.\dot{\mu}(f\bullet).x\ =\ \dot{\mu}(f\bullet).x \quad \text{for all } x\in\mathcal{A}$ ,

$ii \qquad \dot{\mu}(f\bullet).x\ \sqsubseteq\ y \quad \Leftarrow \quad f.y\ \sqsubseteq\ y \quad \text{for all } x,y\in\mathcal{A}$ .

From this it is immediate that $\dot{\mu}(f\bullet).x$ is $\mu f$ for every $x\in\mathcal{A}$. Hence $\dot{\mu}(f\bullet)\ =\ \widehat{\mu f}$ .

Now for the interesting claim, part **c**. First we show that $f^\star\bullet g$ is a fixpoint of $\widehat{g}\,\ddot{\sqcup}\,(f\bullet)$ .

$$
\begin{array}{ll}
 & (\widehat{g}\,\ddot{\sqcup}\,(f\bullet)).(f^\star\bullet g) \\
= & \quad \{\ \text{application}\ \} \\
 & g\,\dot{\sqcup}\,f\bullet f^\star\bullet g
\end{array}
$$

$$
\begin{array}{ll}
= & \quad \{ \text{ distribution of } \bullet g \ \ \} \\
& (I_{\mathcal{A}} \mathbin{\dot\sqcup} f \bullet f^\star) \bullet g \\
= & \quad \{ \text{ corollary 7.12}(\mathbf{f}) \ \ \} \\
& f^\star \bullet g \quad .
\end{array}
$$

Now we prove $f^\star \bullet g$ is the least prefix point of $\widehat{g} \mathbin{\ddot\sqcup} (f\bullet)$.

$$
\begin{array}{ll}
& (\widehat{g} \mathbin{\ddot\sqcup} (f\bullet)).h \ \dot\sqsubseteq \ h \\
\equiv & \quad \{ \text{ application } \ \} \\
& g \mathbin{\dot\sqcup} f \bullet h \ \dot\sqsubseteq \ h \\
\Rightarrow & \quad \{ \text{ suprema, corollary 7.12}(\mathbf{e}) \ \ \} \\
& f^\star \bullet g \ \dot\sqsubseteq \ h \quad .
\end{array}
$$

□

**Solution to exercise: 7.25**

$$
\begin{array}{ll}
& f^\star \bullet (g \bullet f^\star)^\star \\
= & \quad \{ \text{ 7.24}(\mathbf{c}) \ \ \} \\
& f^\star \bullet \dot\mu(h \mapsto I \mathbin{\dot\sqcup} g \bullet f^\star \bullet h) \\
= & \quad \{ \text{ fixed point fusion: (7.23)} \\
& \quad \quad f \ := f^\star\bullet \ , \quad g \ := \ \widehat{I} \mathbin{\ddot\sqcup} (g\bullet) \ \ \} \\
& \dot\mu(h \mapsto f^\star \bullet (I \mathbin{\dot\sqcup} g \bullet h)) \\
= & \quad \{ \text{ 7.24}(\mathbf{c}) \ \ \} \\
& \dot\mu(h \mapsto \dot\mu(k \mapsto (I \mathbin{\dot\sqcup} g \bullet h) \mathbin{\dot\sqcup} f \bullet h)) \\
= & \quad \{ \text{ 7.22}(\mathbf{f}) \text{ and rearrangement of terms } \ \} \\
& \dot\mu(h \mapsto I \mathbin{\dot\sqcup} f \bullet h \mathbin{\dot\sqcup} g \bullet h) \\
= & \quad \{ \text{ definition of } \dot\sqcup \ \ \} \\
& \dot\mu(h \mapsto I \mathbin{\dot\sqcup} (f \mathbin{\dot\sqcup} g) \bullet h) \\
= & \quad \{ \text{ 7.24}(\mathbf{c}) \ \ \} \\
& (f \mathbin{\dot\sqcup} g)^\star \quad .
\end{array}
$$

□

**Solution to exercise: 7.26**

The fixed points of $f$ are the postfix points of $f$ (those $x$ for which $x \sqsubseteq f.x$ holds) in the lattice of prefix points. The prefix points of $f$ form a complete lattice by the prefix lemma 7.7. By the dual of the prefix lemma, the set of postfix points, hence the fixed points, forms a complete lattice.

Now for the second part, an expression for the suprema and infima. Let $\mathcal{F}$ denote the lattice of prefix points and $\mathcal{L}$ the lattice of fixed points. We calculate for $X \subseteq \mathcal{L}$ the supremum, i.e. $\sqcup_{\mathcal{L}}.X$ .

$$
\begin{array}{ll}
 & \sqcup_{\mathcal{L}}.X \\
= & \quad \{ \text{ dual of the prefix lemma with } \mathcal{A} := \mathcal{F} \ \} \\
 & \sqcup_{\mathcal{F}}.X \\
= & \quad \{ \ (3.42) \ \} \\
 & \sqcap_{\mathcal{F}}.\{y \in \mathcal{F} \mid X \sqsubseteq y\} \\
= & \quad \{ \text{ prefix lemma } \} \\
 & \sqcap_{\mathcal{A}}.\{y \in \mathcal{F} \mid X \sqsubseteq y\} \\
= & \quad \{ \text{ definition of } \mathcal{F} \ \} \\
 & \sqcap_{\mathcal{A}}.\{y \in \mathcal{A} \mid X \sqsubseteq y \ \wedge \ f.y \sqsubseteq y\} \\
= & \quad \{ \quad \begin{array}{ll} & X \sqsubseteq y \ \wedge \ f.y \sqsubseteq y \\ \equiv & \quad \{ \ \Rightarrow: X = f.X \text{ and monotonicity of } f, \\ & \quad\ \ \Leftarrow: \text{transitivity of } \sqsubseteq \ \} \\ & X \sqsubseteq f.y \sqsubseteq y \ \ \} \end{array} \\
 & \sqcap_{\mathcal{A}}.\{y \in \mathcal{A} \mid X \sqsubseteq f.y \sqsubseteq y\} \quad .
\end{array}
$$

□

**Solution to exercise: 8.20**

For the proof of (8.21); instantiating $x$ to $\perp\!\!\perp$ in (8.19) gives

$$\perp\!\!\perp^* \quad = \quad I^+ \quad = \quad I^* \quad ,$$

and

$$
\begin{array}{ll}
 & I^+ \ = \ I \\
\equiv & \quad \{ \ (8.9) \ \} \\
 & I^+ \ \sqsubseteq \ I \\
\Leftarrow & \quad \{ \ (8.10) \ \} \\
 & I \ \sqsubseteq \ I \ \ \wedge \ \ I \circ I \ \sqsubseteq \ I \\
\equiv & \quad \{ \ I \text{ is unit } \} \\
 & \mathit{true} \quad .
\end{array}
$$

Not that only the proof of $I^+ = I$ uses the fact that $I$ is the unit of $\circ$ .

For (8.22)

$$\begin{array}{ll}
& x^* \ = \ I \sqcup x^+ \\
\equiv & \quad \{ \ (8.11) \text{ and } (8.17) \ \} \\
& x^* \ \sqsubseteq \ I \sqcup x^+ \\
\Leftarrow & \quad \{ \ (8.13) \ \} \\
& x \ \sqsubseteq \ I \sqcup x^+ \ \ \wedge \ \ (I \sqcup x^+) \circ (I \sqcup x^+) \ \sqsubseteq \ I \sqcup x^+ \\
\equiv & \quad \{ \ (8.12), \ \circ \text{ is } \sqcup\text{-junctive} \ \} \\
& I \circ I \ \sqcup \ I \circ x^+ \ \sqcup \ x^+ \circ I \ \sqcup \ x^+ \circ x^+ \ \sqsubseteq \ I \sqcup x^+ \\
\equiv & \quad \{ \ I \text{ is unit of } \circ \ \} \\
& I \ \sqcup \ x^+ \ \sqcup \ x^+ \circ x^+ \ \sqsubseteq \ I \sqcup x^+ \\
\equiv & \quad \{ \ \text{calculus, } (8.11) \ \} \\
& \mathit{true} \quad .
\end{array}$$

Finally for (8.23), observe

$$\begin{array}{ll}
& x^* \ = \ x^* \circ x^* \\
\equiv & \quad \{ \ (8.11) \ \} \\
& x^* \ \sqsubseteq \ x^* \circ x^* \\
\Leftarrow & \quad \{ \ (8.11){:}\ I \sqsubseteq x^* \ \} \\
& x^* \ \sqsubseteq \ I \circ x^* \\
\equiv & \quad \{ \ I \text{ is identity of } \circ \ \} \\
& \mathit{true} \quad .
\end{array}$$

□

**Solution to exercise: 8.35**

The proof is by mutual containment

$$\begin{array}{ll}
& (x\circ)^\star.x \ \sqsubseteq \ x^+ \\
\Leftarrow & \quad \{ \ (8.26) \ \} \\
& x \ \sqsubseteq \ x^+ \ \ \wedge \ \ x \circ x^+ \ \sqsubseteq \ x^+ \\
\Leftarrow & \quad \{ \ (8.9) \text{ — applied twice } \} \\
& x^+ \circ x^+ \ \sqsubseteq \ x^+ \\
\equiv & \quad \{ \ (8.8) \ \} \\
& \mathit{true} \quad .
\end{array}$$

The other inclusion is proven by

$$\begin{array}{ll}
& x^+ \ \sqsubseteq \ (x\circ)^\star.x \\
\Leftarrow & \quad \{ \ (8.10) \ \}
\end{array}$$

$$
\begin{array}{ll}
 & x \sqsubseteq (x\circ)^\star.x \quad \wedge \quad (x\circ)^\star.x \circ (x\circ)^\star.x \sqsubseteq (x\circ)^\star.x \\
\equiv & \qquad \{ \text{ corollary 7.12}(\mathbf{b})\text{; (8.29) } \} \\
 & (x\circ)^\star.(x \circ (x\circ)^\star.x) \quad \sqsubseteq \quad (x\circ)^\star.x \\
\Leftarrow & \qquad \{ \text{ corollary 7.12}(\mathbf{e}) \text{ } \} \\
 & x \circ (x\circ)^\star.x \sqsubseteq (x\circ)^\star.x \quad \wedge \quad (x\circ)^\star.(x\circ)^\star.x \sqsubseteq (x\circ)^\star.x \\
\equiv & \qquad \{ \text{ theorem 7.10}(\mathbf{a}) \text{ and corollary 7.12}(\mathbf{c}) \text{ } \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

**Solution to exercise: 8.36**
It is the objective to use

$$(17.1) \quad ((a\circ)^\star.x \sqsubseteq y \quad \equiv \quad x \sqsubseteq y) \quad \Leftarrow \quad a \circ y \sqsubseteq y \quad .$$

$$
\begin{array}{ll}
 & (a\circ)^\star.x \quad \sqsubseteq \quad x \circ (b\circ)^\star.I \\
\equiv & \qquad \{ \text{ (8.28) } \} \\
 & (a\circ)^\star.I \circ x \quad \sqsubseteq \quad x \circ (b\circ)^\star.I \\
\equiv & \qquad \{ \text{ factors: (8.3) } \} \\
 & (a\circ)^\star.I \quad \sqsubseteq \quad (x \circ (b\circ)^\star.I)/x \\
\equiv & \qquad \{ \text{ (17.1), see below } \} \\
 & I \quad \sqsubseteq \quad (x \circ (b\circ)^\star.I)/x \\
\equiv & \qquad \{ \text{ factors: (8.3) } \} \\
 & x \quad \sqsubseteq \quad x \circ (b\circ)^\star.I \\
\Leftarrow & \qquad \{ \text{ corollary 7.12}(\mathbf{b})\text{, } I \text{ is unit of } \circ \text{ } \} \\
 & \mathit{true} \quad .
\end{array}
$$

In the middle step we appeal to (17.1). To verify the antecedent for the case in question we calculate as follows:

$$
\begin{array}{ll}
 & a \circ (x \circ (b\circ)^\star.I)/x \quad \sqsubseteq \quad (x \circ (b\circ)^\star.I)/x \\
\Leftarrow & \qquad \{ \text{ factors: (8.3), and cancellation: (8.5) } \} \\
 & a \circ x \circ (b\circ)^\star.I \quad \sqsubseteq \quad x \circ (b\circ)^\star.I \\
\Leftarrow & \qquad \{ \text{ theorem 7.10}(\mathbf{a})\text{, monotonicity } \} \\
 & a \circ x \circ (b\circ)^\star.I \quad \sqsubseteq \quad x \circ b \circ (b\circ)^\star.I \\
\Leftarrow & \qquad \{ \text{ monotonicity } \} \\
 & a \circ x \quad \sqsubseteq \quad x \circ b \\
\equiv & \qquad \{ \text{ assumption } \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

**Solution to exercise: 8.45**

A counterexample to (8.46) is the following.

Consider the set $\mathcal{R}$ of binary relations over the set $\{a,b\}$. Let $R$ range over $\mathcal{R}$. Take $g$ to be the identity function on $\mathcal{R}$ and $f = sq = (R \mapsto R \circ R)$. We demonstrate that $f \bullet f^\star \neq f^\star \bullet f$.

| | |
|---|---|
| Take | $R = \{(a,b),(b,a)\}$ . |
| Then | $sq.R = \{(a,a),(b,b)\}$ (which is a transitive relation), |
| and | $sq^\star.R = \{(a,b),(b,a),(a,a),(b,b)\}$ . |
| So, | $(sq^\star \bullet sq).R \;=\; \{(a,a),(b,b)\}$ , |
| and | $(sq \bullet sq^\star).R \;=\; \{(a,b),(b,a),(a,a),(b,b)\}$ . |

Note however that one inclusion is valid, namely:

(17.2) $(f \bullet g)^\star \bullet f \quad \dot{\sqsubseteq} \quad f \bullet (g \bullet f)^\star$ .

That this inclusion is valid but not the opposite inclusion is attributable to the fact that the function $(\bullet f)$ is universally $\sqcup$-junctive in the lattice of lifted functions whereas $(f\bullet)$ is not. Since the proof of 17.2 closely resembles the proof of the matching inclusion in the leapfrog rule we do not supply it here.
□

**Solution to exercise: 8.49**

**Step 1.** Reflexivity of \ is equivalent to $I \sqsubseteq X\backslash X$ for all $X$ which, in turn, is equivalent to $I$ being a right unit of composition. Dually, reflexivity of / is equivalent to $I$ being a left unit of composition. Transitivity of \ is the property that, for all $X$ and $Z$,

$$\sqcup.(Y :: X\backslash Y \circ Y\backslash Z) \quad \sqsubseteq \quad X\backslash Z \ .$$

This, by the definition of supremum, is equivalent to, for all $X$, $Y$ and $Z$,

$$X\backslash Y \circ Y\backslash Z \quad \sqsubseteq \quad X\backslash Z \ .$$

We leave this calculation (one use of the Galois connection between $(X\backslash)$ and $(X\circ)$ plus two uses of cancellation) to the reader.
**Step 2.** The Galois connection is

$$R\triangleleft \sqsupseteq S \quad \equiv \quad R \sqsubseteq S\triangleright$$

from which (8.55) and (8.56) follow immediately.

We prove (8.57) by the rule of indirect equality.

$$
\begin{array}{ll}
 & X \;\sqsubseteq\; E{\triangleleft}{\triangleright} \\
\equiv & \qquad \{\ (8.54),\ \text{factors}\ \} \\
 & E{\triangleleft} \circ X \;\sqsubseteq\; E \\
\equiv & \qquad \{\ (8.53)\ \} \\
 & E/E \circ X \;\sqsubseteq\; E \\
\equiv & \qquad \{\ (\Rightarrow)\ \ I \sqsubseteq E/E \\
 & \qquad\quad (\Leftarrow)\ \ \text{cancellation and monotonicity}\ \} \\
 & X \;\sqsubseteq\; E \quad .
\end{array}
$$

By a dual proof $E = E{\triangleright}{\triangleleft}$ .

**Step 3.** Suppose $F = X\backslash E/Y$ . Then we have:

$$
\begin{array}{ll}
 & X\backslash E/Y \\
= & \qquad \{\ (8.54)\ \} \\
 & (X{\triangleright})/Y \\
= & \qquad \{\ (8.56)\ \} \\
 & (X{\triangleright}{\triangleleft}{\triangleright})/Y \\
= & \qquad \{\ (8.61)\ \} \\
 & (X{\triangleright}{\triangleleft})\backslash(Y{\triangleleft}) \quad .
\end{array}
$$

Thus we take $L_0 = X{\triangleright}{\triangleleft}$ and $L_1 = Y{\triangleleft}$. To construct $L_2$ and $L_3$ we calculate:

$$
\begin{array}{ll}
 & L \\
= & \qquad \{\ L = X{\triangleleft}\ \text{for some}\ X,\ (8.55)\ \} \\
 & L{\triangleright}{\triangleleft} \\
= & \qquad \{\ (8.53)\ \} \\
 & E/(L{\triangleright}) \\
= & \qquad \{\ (8.57)\ \} \\
 & (E{\triangleleft}{\triangleright})/(L{\triangleright}) \\
= & \qquad \{\ (8.61)\ \} \\
 & (E{\triangleleft})\backslash(L{\triangleright}{\triangleleft}) \\
= & \qquad \{\ L = X{\triangleleft}\ \text{for some}\ X,\ (8.55)\ \} \\
 & (E{\triangleleft})\backslash L \quad .
\end{array}
$$

So we take $L_2 = E{\triangleleft}$ and $L_3 = L$. Since $L_2$ is independent of $L$ and $L_3$ is trivially uniquely defined by $L$, the $(E{\triangleleft})$th row of the factor matrix comprises exactly one occurrence of each left factor of $E$.

For the right factors we have the following calculation:

$$
\begin{array}{ll}
 & R \\
= & \qquad \{ \;\; R \; = \; X{\rhd} \text{ for some } X, \; (8.56) \;\; \} \\
 & R{\lhd\rhd} \\
= & \qquad \{ \;\; (8.54) \;\; \} \\
 & (R{\lhd})\backslash E \quad .
\end{array}
$$

Since $E$ is a left factor of itself (see (8.57)) we take $L_4 = R{\lhd}$ and $L_5 = E$. Again it is easy to see from the above calculation that the $E$th column of the factor matrix comprises exactly one occurrence of each right factor of $E$. Finally, property (8.62) is a trivial consequence of the identity $L = (E{\lhd})\backslash L$, for all left factors $L$, proved above and the fact that $E$ is a left factor of itself.
□

**Solution to exercise: 8.65**
Property (8.66) is verified as follows:

$$
\begin{array}{ll}
 & X \circ Y \;\; \sqsubseteq \;\; E \\
\equiv & \qquad \{ \;\; (8.54) \;\; \} \\
 & Y \;\; \sqsubseteq \;\; X{\rhd} \\
\equiv & \qquad \{ \;\; (8.56) \;\; \} \\
 & Y \;\; \sqsubseteq \;\; X{\rhd}{\lhd\rhd} \\
\equiv & \qquad \{ \;\; {\lhd\rhd} \text{ is a closure operator } \;\; \} \\
 & Y{\lhd\rhd} \;\; \sqsubseteq \;\; X{\rhd}{\lhd\rhd} \\
\equiv & \qquad \{ \;\; (8.54) \;\; \} \\
 & X{\rhd\lhd} \circ Y{\lhd\rhd} \;\; \sqsubseteq \;\; E \quad .
\end{array}
$$

Since ${\rhd\lhd}$ and ${\lhd\rhd}$ are both closure operators (and thus $X \sqsubseteq X{\rhd\lhd}$ and $Y \sqsubseteq Y{\lhd\rhd}$) it also follows that all four of the following inclusions are equivalent

$$
\begin{array}{l}
X \circ Y \;\; \sqsubseteq \;\; E \;\; , \\
X{\rhd\lhd} \circ Y \;\; \sqsubseteq \;\; E \;\; , \\
X \circ Y{\lhd\rhd} \;\; \sqsubseteq \;\; E \;\; , \\
X{\rhd\lhd} \circ Y{\lhd\rhd} \;\; \sqsubseteq \;\; E \;\; .
\end{array}
$$

Property (8.67) can now be verified using indirect equality:

$$
\begin{array}{ll}
 & Z \;\; \sqsubseteq \;\; (X \circ Y){\rhd} \\
\equiv & \qquad \{ \;\; (8.54) \;\; \}
\end{array}
$$

$$
\begin{array}{ll}
& X \circ Y \circ Z \quad \sqsubseteq \quad E \\
\equiv & \qquad \{ \text{ above } \} \\
& X{\rhd\lhd} \circ Y \circ Z \quad \sqsubseteq \quad E \\
\equiv & \qquad \{ \text{ (8.54) } \} \\
& Z \quad \sqsubseteq \quad (X{\rhd\lhd} \circ Y){\rhd} \qquad .
\end{array}
$$

□

# Appendix

## A Preliminary Remarks and Some Abbreviations

In this appendix we consider two aspects of the structured calculus of relations presented in section 9. These are:

(a) the completeness and independence of the constituent parts of the axiomatisation

(b) proofs of basic but non-evident results needed elsewhere in the paper.

Since the purpose is to support the use of the calculus in the remainder of the paper the discussion is at times terse and limited.

We may briefly summarise the discussion of the axiomatisation as follows. First, the system of axioms is a sound but not complete axiomatisation of the binary relations over some universe: we demonstrate the incompleteness by exhibiting a model that fulfills all the axioms but is obviously not isomorphic with some class of binary relations. Second, each of the different layers is independent of the others but for the reverse structure. The dependence of the reverse structure on the remainder of the axiomatisation permits an alternative formulation of the axioms in which reverse is a defined notion. The independence of the individual layers is discussed at the same time as we discuss the completeness and soundness of the axiom system (since both apects involve exhibiting models), namely in section C. How reverse might have been introduced as a defined notion is discussed in section B.

For reference purposes we name the constituent parts of the axiomatisation as follows:

| | |
|---|---|
| $\mathcal{P}$ | : the plat structure |
| $\mathcal{C}$ | : the composition structure |
| $\mathcal{R}$ | : the reverse structure |
| $\mathcal{PC}$ interface | : the interface between the $\mathcal{P}$ and $\mathcal{C}$ structures i.e. " $\circ$ " is universally cupjunctive |

| | |
|---|---|
| $\mathcal{PR}$ interface | : the interface between the $\mathcal{P}$ and $\mathcal{R}$ structures<br>i.e. "$\cup$" is a plat automorphism<br>specifically, $P \sqsupseteq Q \;\equiv\; P\cup \sqsupseteq Q\cup$ |
| $\mathcal{CR}$ interface | : the interface between the $\mathcal{C}$ and $\mathcal{R}$ structures<br>i.e. "$\cup$" is a contravariant monoid isomorphism<br>specifically, $(P \circ Q)\cup \;=\; Q\cup \circ P\cup$<br>and $I\cup \;=\; I$ |
| $\mathcal{PC}$ | : the combination of $\mathcal{P}$, $\mathcal{C}$ and their interfaces |
| $\mathcal{PCR}$ | : the combination of $\mathcal{P}$, $\mathcal{C}$, $\mathcal{R}$ and their interfaces |
| $\mathcal{M}$ | : the middle exchange rule |
| $c$ | : the cone rule |

The reader may wish to remind themself of our conventions on operator precedence, detailed in section 9.1.4, before reading further.

# B Dependence

In this section we discuss the reverse structure, $\mathcal{R}$, and its interfaces with the plat structure, $\mathcal{P}$, and the monoid structure, $\mathcal{C}$. It is shown that, with a suitable definition of "$\cup$", all of $\mathcal{R}$, the $\mathcal{PR}$ interface and the $\mathcal{CR}$ interface follow from $\mathcal{PCM}$. This opens up the possibility for an alternative presentation of the axiomatic framework in which the reverse operator does not appear within a separate layer but is a defined notion within the algebraic structure $\mathcal{PC}$. A second alternative is furnished by a mixture of the original presentation and the first alternative. Both alternatives are investigated in some detail.

Yet another presentation of the axiomatic framework is furnished by a rule dubbed "Dedekind's rule" by Schmidt and Ströhlein [84]. This alternative we also discuss in some detail. (We are grateful to Schmidt and Ströhlein for their insistence on the importance of this rule in lectures they gave in Utrecht in 1991.)

Which presentation of the axiomatisation one chooses is an important question. We have to admit that the presentation chosen here reflects our relative unfamiliarity with the relational calculus when we began this research rather than a well-considered choice. In future revisions of this report it is likely that we will build up the calculus in a quite different way.

## B.1 The Axiom $\mathcal{F}$

We begin by remarking that, within the algebraic framework $\mathcal{PC}$, definition 9.1 makes sense. We recall that the left factor, $S/R$, is defined by the property:

(B0) $\quad S/R \sqsupseteq X \quad\equiv\quad S \sqsupseteq X \circ R$

That it is well-defined is established by verifying that

$$S/R \quad=\quad \sqcup(X\colon\ S \sqsupseteq X \circ R\colon\ X)$$

We now remark that, for all $Q$, "$Q\cup$" can be reexpressed entirely within the language of $\mathcal{PC}$. Specifically,

**Theorem B1** Within the algebra $\mathcal{PCRM}$

$$Q\cup \quad=\quad \neg I/\neg Q$$

**Proof** We have, for arbitrary $X$,

$$\begin{array}{ll}
 & \neg I/\neg Q \sqsupseteq X \\
\equiv & \qquad \{\ \text{(B0); } I \text{ is the unit of composition}\ \} \\
 & \neg I \sqsupseteq X \circ \neg Q \circ I \\
\equiv & \qquad \{\ \text{middle exchange rule; } I \text{ is the unit of composition}\ \} \\
 & Q \sqsupseteq X\cup \circ I\cup \\
\equiv & \qquad \{\ \mathcal{CR} \text{ and } \mathcal{PR} \text{ interfaces}\ \} \\
 & Q\cup \sqsupseteq X
\end{array}$$

□

**Theorem B2** The system of axioms $\mathcal{PCMF}$ consisting of $\mathcal{PCM}$ supplemented by the definition

(B3) $\quad Q\cup \quad=\quad \neg I/\neg Q$

is equivalent to $\mathcal{PCRM}$.

**Proof**
On account of theorem B1 it suffices for us to show that the addition of (B3) to $\mathcal{PCM}$ implies all the remaining axioms of $\mathcal{PCRM}$. There are thus three elements to the proof. We have to establish that "$\cup$" is its own inverse, the $\mathcal{CR}$ interface and the $\mathcal{PR}$ interface. As a preliminary we show that $I\cup = I$.

From (B3) and (B0) it follows that

(B4) $\quad I\cup \quad\sqsupseteq\quad I$

whilst

$$
\begin{array}{ll}
 & I \;\sqsupseteq\; I^{\cup} \\
\Leftarrow & \quad \{\ (B4)\ ;\ \mathcal{PC}\ \} \\
 & I \;\sqsupseteq\; I^{\cup} \circ I \circ I^{\cup} \\
\equiv & \quad \{\ \mathcal{M}\ \} \\
 & \neg I \;\sqsupseteq\; I \circ \neg I \circ I \\
\equiv & \quad \{\ \mathcal{C}\ \} \\
 & \mathbf{true}
\end{array}
$$

Hence

(B5) $\quad I^{\cup} \;=\; I$

We make frequent use of the middle exchange rule in the remainder of the proof. On occasion, in order to use the rule, we use the axiom that $I$ is the unit of composition to insert "$I$" and/or — by (B5) — "$I^{\cup}$" into a sequence of compositions. Such insertions will go unannounced, the hint given being simply "$\mathcal{M}$". Similarly, deletions of "$I$" or "$I^{\cup}$" in a sequence of compositions will also occur without mention.

We can now prove that "$\cup$" is its own inverse since we have, for arbitrary $X$,

$$
\begin{array}{ll}
 & Q^{\cup\cup} \sqsupseteq X \\
\equiv & \quad \{\ (B3),\ (B0)\ \} \\
 & \neg I \sqsupseteq X \circ \neg(Q^{\cup}) \\
\equiv & \quad \{\ \mathcal{M}\ \} \\
 & Q^{\cup} \sqsupseteq X^{\cup} \\
\equiv & \quad \{\ (B3),\ (B0)\ \} \\
 & \neg I \sqsupseteq X^{\cup} \circ \neg Q \\
\equiv & \quad \{\ \mathcal{M}\ \} \\
 & \neg\neg Q \sqsupseteq X \circ \neg\neg I \\
\equiv & \quad \{\ \mathcal{P};\ \mathcal{C}\ \} \\
 & Q \sqsupseteq X
\end{array}
$$

Hence

(B6) $\quad Q^{\cup\cup} \;=\; Q$

The interface between $\mathcal{P}$ and $\mathcal{R}$ follows from:

$$
\begin{array}{ll}
 & P^{\cup} \sqsupseteq Q^{\cup} \\
\equiv & \quad \{\ (B3),\ (B0)\ \}
\end{array}
$$

$$
\begin{array}{ll}
 & \neg I \sqsupseteq Q^{\cup} \circ \neg P \\
\equiv & \quad \{ \mathcal{M} \text{ ; (B6) } \} \\
 & P \sqsupseteq Q
\end{array}
$$

Only the $\mathcal{CR}$ interface is left to prove (and we have already (B5)). The contravariance of $\cup$ follows from:

$$
\begin{array}{ll}
 & X \sqsupseteq (P \circ Q)^{\cup} \\
\equiv & \quad \{ \mathcal{PR} \text{ interface, (B6) } \} \\
 & X^{\cup} \sqsupseteq P \circ Q \\
\equiv & \quad \{ \text{(B3), (B0)} \} \\
 & \neg I \sqsupseteq P \circ Q \circ \neg X \\
\equiv & \quad \{ \mathcal{M} \} \\
 & \neg(Q \circ \neg X) \sqsupseteq P^{\cup} \\
\equiv & \quad \{ \mathcal{P} \} \\
 & \neg(P^{\cup}) \sqsupseteq Q \circ \neg X \\
\equiv & \quad \{ \mathcal{M} \} \\
 & X \sqsupseteq Q^{\cup} \circ P^{\cup}
\end{array}
$$

□

We may conclude from the above that the reverse structure is not independent of the remainder of the axiomatisation.

The first alternative formulation of the axiomatisation can now be explained. Let $\mathcal{PCRF}$ denote $\mathcal{PCR}$ supplemented by the property (B3). Then we have:

**Theorem B7** $\quad \mathcal{PCRM}$ and $\mathcal{PCRF}$ are equivalent.

**Proof**

On account of the above it suffices to show that the middle exchange rule can be derived within $\mathcal{PCRF}$. Our proof involves three stages. First, we derive a rule called the "divergence rule", next we derive the rotation rule first mentioned in section 9.1 and then we derive the complete middle exchange rule (these proofs being conducted, of course, under the assumption of $\mathcal{PCRF}$).

The statement of the divergence rule is as follows:

(B8) $\quad \neg I \sqsupseteq P \circ Q \quad \equiv \quad \neg I \sqsupseteq Q \circ P$

Note that (B8) is an expression within the language of $\mathcal{PC}$. Its name comes from the fact that $\neg I$ is sometimes given the name "divergence". (The interpretation of $\neg I$ is a relation between pairs of unequal, i.e. "divergent", elements.) To establish the rule we make the following calculation:

$$
\begin{array}{ll}
 & \neg I \sqsupseteq P \circ Q \\
\equiv & \qquad \{ \text{ factors: (B0) } \} \\
 & \neg I \;/\; Q \sqsupseteq P \\
\equiv & \qquad \{ \text{ (B3), } \mathcal{P} \ \} \\
 & (\neg Q)^{\cup} \sqsupseteq P \\
\equiv & \qquad \{ \ \mathcal{PR} \ \} \\
 & (\neg P)^{\cup} \sqsupseteq Q \\
\equiv & \qquad \{ \text{ (B3), } \mathcal{P} \ \} \\
 & \neg I \;/\; P \sqsupseteq Q \\
\equiv & \qquad \{ \text{ factors: (B0) } \} \\
 & \neg I \sqsupseteq Q \circ P
\end{array}
$$

Note that the last two steps are the mirror image of the first two.

Now for the rotation rule:

$$
\begin{array}{ll}
 & \neg P^{\cup} \ \sqsupseteq \ Q \circ R \\
\equiv & \qquad \{ \text{ (B3), (B0) } \} \\
 & \neg I \ \sqsupseteq \ Q \circ R \circ P \\
\equiv & \qquad \{ \text{ divergence rule: (B8) } \} \\
 & \neg I \sqsupseteq R \circ P \circ Q \\
\equiv & \qquad \{ \text{ (B3), (B0) } \} \\
 & \neg Q^{\cup} \ \sqsupseteq \ R \circ P
\end{array}
$$

So we have established the rotation rule:

(B9) $\quad \neg P^{\cup} \sqsupseteq Q \circ R \;\equiv\; \neg Q^{\cup} \sqsupseteq R \circ P$

Finally we may proceed to the middle exchange rule.

$$
\begin{array}{ll}
 & \neg Y \ \sqsupseteq \ P \circ \neg X \circ Q \\
\equiv & \qquad \{ \text{ rotation rule: (B9) } \} \\
 & \neg P^{\cup} \ \sqsupseteq \ \neg X \circ Q \circ Y^{\cup} \\
\equiv & \qquad \{ \text{ rotation rule: (B9) } \} \\
 & X^{\cup} \ \sqsupseteq \ Q \circ Y^{\cup} \circ P \\
\equiv & \qquad \{ \ \mathcal{CR} \text{ interface } \} \\
 & X \ \sqsupseteq \ P^{\cup} \circ Y \circ Q^{\cup}
\end{array}
$$

□

## B.2 Dedekind's Rule

One of the more difficult but frequently occurring tasks in the relational calculus is to simplify an expression involving both composition and the cap operator. This is the primary motivation for the rule that Schmidt and Ströhlein [84] dub "Dedekind's rule".

Dedekind's rule is, on first encounter, yet more forbidding than the middle exchange rule: its syntactic shape is less attractive, the rule is plucked out of the hat, is relatively complicated to use and its proof involves an ugly case analysis. Against this must be weighed the fact that the rule is extraordinarily powerful — once proven it simplifies enormously the proofs of several other basic properties. Moreover the rule does not involve complementation and yet, in combination with $\mathcal{PCR}$ is completely equivalent to $\mathcal{PCRM}$.

**Lemma B10 (Dedekind's Rule: 1st Version)** In the axiom system $\mathcal{PCRM}$ the following inclusions are valid:

(a) $T \sqcap U \circ V \;\sqsubseteq\; (T \;\sqcap\; U, \circ V^{\cup}) \;\circ\; V$
(b) $T \sqcap U \circ V \;\sqsubseteq\; U \;\circ\; (U^{\cup} \circ T \;\sqcap\; V)$
(c) $T \sqcap U \circ V \;\sqsubseteq\; (V \;\sqcap\; T \circ U^{\cup}) \;\circ\; (U^{\cup} \circ T \;\sqcap\; V)$

**Proof** We begin with (a).

$$
\begin{array}{lll}
 & T \sqcap U \circ V & \\
= & \quad \{ \text{ excluded middle } \} & \\
 & T \;\sqcap\; (U \;\sqcap\; (T \circ V^{\cup} \sqcup \neg(T \circ V^{\cup}))) \circ V & \\
= & \quad \{ \text{ distributivity } \} & \\
 & (T \;\sqcap\; (U \sqcap T \circ V^{\cup}) \circ V) \;\sqcup\; (T \;\sqcap\; (U \sqcap \neg(T \circ V^{\cup})) \circ V) & \\
= & \quad \{ \quad T \;\sqcap\; (U \sqcap \neg(T \circ V^{\cup})) \circ V \;\sqsubseteq\; \perp\!\!\perp & \\
 & \qquad \equiv \qquad \{ \text{ shunting rule } \} & \\
 & \qquad\qquad (U \sqcap \neg(T \circ V^{\cup})) \circ V \;\sqsubseteq\; \neg T & \\
 & \qquad \Leftarrow \qquad \{ \text{ monotonicity } \} & \\
 & \qquad\qquad \neg(T \circ V^{\cup}) \circ V \;\sqsubseteq\; \neg T & \\
 & \qquad \equiv \qquad \{ \text{ left exchange rule } \} & \\
 & \qquad\qquad T \circ V^{\cup} \sqsubseteq T \circ V^{\cup} & \\
 & \qquad \equiv & \\
 & \qquad\qquad \textbf{true} & \\
 & \quad \} & \\
 & T \;\sqcap\; (U \sqcap T \circ V^{\cup}) \circ V &
\end{array}
$$

Property (a) now follows by simple plat calculus.

By a similar proof, or by applying reverse to the equality above, one obtains:

$$T \sqcap U \circ V \quad = \quad T \ \sqcap \ U \circ (U^{\cup} \circ T \sqcap V)$$

Property (b) is an immediate consequence. Property (c) combines (a) and (b):

$$\begin{array}{ll}
 & T \sqcap U \circ V \\
= & \quad \{ \ \text{(a)} \ \} \\
 & T \ \sqcap \ (U \ \sqcap \ T \circ V^{\cup}) \ \circ \ V \\
= & \quad \{ \ \text{(b)}, U \ := \ U \ \sqcap \ T \circ V^{\cup} \ \} \\
 & T \ \sqcap \ (U \ \sqcap \ T \circ V^{\cup}) \ \circ \ ((U \ \sqcap \ T \circ V^{\cup})^{\cup} \circ T \ \sqcap \ V) \\
= & \quad \{ \ \text{properties of reverse} \ \} \\
 & T \ \sqcap \ (U \ \sqcap \ T \circ V^{\cup}) \circ ((U^{\cup} \ \sqcap \ V \circ T^{\cup}) \ \circ \ T \ \sqcap \ V) \\
= & \quad \{ \ \text{(a)}, T, U, V \ := \ V, U^{\cup}, T \ \} \\
 & T \ \sqcap \ (U \ \sqcap \ T \circ V^{\cup}) \circ (U^{\cup} \circ T \ \sqcap \ V) \\
\sqsubseteq & \quad \{ \ \text{calculus} \ \} \\
 & (U \ \sqcap \ T \circ V^{\cup}) \ \circ \ (U^{\cup} \circ T \ \sqcap \ V)
\end{array}$$

□

A slight reformulation of Dedekind's rule, even though it involves two extra dummies, pays handsome dividends in terms of increased useability:

**Corollary B11 (Dedekind's Rule)** In the axiom system $\mathcal{PCRM}$ we have:

(a) $R \circ S \ \sqsupseteq \ U \sqcap T \circ S \ \Leftarrow \ R \ \sqsupseteq \ T \ \sqcap \ U \circ S^{\cup}$

(b) $R \circ S \ \sqsupseteq \ U \sqcap R \circ T \ \Leftarrow \ S \ \sqsupseteq \ T \ \sqcap \ R^{\cup} \circ U$

(c) $R \circ S \ \sqsupseteq \ T \ \sqcap \ U \circ V$

$\qquad \Leftarrow \ R \ \sqsupseteq \ U \ \sqcap \ T \circ V^{\cup} \ \wedge \ S \ \sqsupseteq \ U^{\cup} \circ T \sqcap V$

□

Schmidt and Ströhlein [84] attribute lemma B10 to Dedekind and J. Riguet and subsequently refer to it as Dedekind's rule ("Dedekind-Formel"). We will, however, never use the rule in that form preferring always to use corollary B11. It is therefore this corollary that we refer to when we cite "Dedekind's rule".

We asserted at the beginning of this subsection that Dedekind's rule is extraordinarily powerful. Formally, it is equivalent to the middle exchange rule in the context of $\mathcal{PCR}$. This we shall now prove. (This was pointed out to us by Henk Doornbos. Schmidt and Ströhlein [84] seem to make a similar claim, but the exercise they give to support the claim asserts only an implication.) Let $\mathcal{PCRD}$ denote the system of axioms $\mathcal{PCR}$ supplemented by the property B11(b).

**Theorem B12** $\mathcal{PCRD}$ is equivalent to $\mathcal{PCRF}$ and $\mathcal{PCRM}$.

**Proof** We have already shown that $\mathcal{PCRM}$ implies $\mathcal{PCRD}$ and that $\mathcal{PCRF}$ and $\mathcal{PCRM}$ are equivalent. It remains to show that $\mathcal{PCRD}$ implies $\mathcal{PCRF}$, i.e. (B3) can be derived within the context of $\mathcal{PCRD}$. An intermediate stage is to show that, in the context $\mathcal{PCRD}$,

(B13) $\neg I \sqsupseteq R \circ S \quad \equiv \quad \perp\!\!\perp \sqsupseteq S \sqcap R^{\cup}$

following which we complete the derivation of property (B3).

$$
\begin{array}{ll}
 & \neg I \sqsupseteq R \circ S \\
\equiv & \quad \{ \text{ shunting rule } \} \\
 & \perp\!\!\perp \sqsupseteq R \circ S \sqcap I \\
\equiv & \quad \{ \mathcal{PC} \} \\
 & R \circ \perp\!\!\perp \sqsupseteq R \circ S \sqcap I \\
\Leftarrow & \quad \{ \text{ Dedekind's rule: B11(b) } \} \\
 & \perp\!\!\perp \sqsupseteq S \sqcap R^{\cup} \circ I \\
\equiv & \quad \{ \mathcal{C} \} \\
 & \perp\!\!\perp \sqsupseteq S \sqcap R^{\cup} \\
\equiv & \quad \{ \mathcal{C}, \mathcal{P} \} \\
 & R^{\cup} \circ \perp\!\!\perp \sqsupseteq R^{\cup} \circ I \sqcap S \\
\Leftarrow & \quad \{ \text{ Dedekind's rule: B11(b), reverse } \} \\
 & \perp\!\!\perp \sqsupseteq I \sqcap R \circ S \\
\equiv & \quad \{ \text{ shunting rule } \} \\
 & \neg I \sqsupseteq R \circ S
\end{array}
$$

Now for (B3):

$$
\begin{array}{ll}
 & \neg I \,/\, \neg R \sqsupseteq X \\
\equiv & \quad \{ \text{ factors: (B0) } \} \\
 & \neg I \sqsupseteq X \circ \neg R \\
\equiv & \quad \{ \text{ (B13) } \} \\
 & \perp\!\!\perp \sqsupseteq X^{\cup} \sqcap \neg R
\end{array}
$$

$$
\begin{array}{ll}
\equiv & \{\ \text{plat calculus, shunting}\ \} \\
& R \sqsupseteq X^{\cup} \\
\equiv & \{\ \text{reverse}\ \} \\
& R^{\cup} \sqsupseteq X
\end{array}
$$

Hence

$$\neg I \;/\; \neg R \;=\; R^{\cup}$$

□

In conclusion, we have exhibited four equivalent axiomatisations: $\mathcal{PCRM}$, $\mathcal{PCRF}$, $\mathcal{PCMF}$ and $\mathcal{PCRD}$.

# C Independence and Completeness

In this section evidence is presented, via a variety of models, for the independence of several parts of the axiomatisation. It turns out that $\mathcal{PCRM}c$ does not characterize the binary relations completely. Most of the proofs are elementary so they are omitted.

## C.1 Power Sets

The starting point for all models is the powerset, for

**C0** *The powerset $\mathbb{P}(V)$, for any set $V$, with $\sqcup$ and $\sqcap$ interpreted as set union and set intersection, respectively, satisfies $\mathcal{P}$.*

In this power set model, $\top\!\!\top$ and $\bot\!\!\bot$ are, of course, $V$ and the empty set, respectively.

On $\mathbb{P}(V)$ we may define a composition and a reverse as the intersection and the identity, respectively. With these definitions, $\mathbb{P}(V)$ satisfies $\mathcal{PCR}$ and also $\mathcal{M}$, for $\mathcal{M}$ is just the shunting rule. However, $c$ is not satisfied for nontrivial $V$ since:

$$V \cap R \cap V \;=\; V \quad\equiv\quad R \;=\; V$$

Referring to the elements of $\mathbb{P}(V)$ as predicates we may summarise the foregoing by:

**C1** *The predicates satisfy $\mathcal{PCRM}$ but, in general, not $c$.*

## C.2 Binary Relations

The binary relations on some set $\mathbb{U}$ are obtained by choosing $V = \mathbb{U} \times \mathbb{U}$ and defining composition and reverse on $\mathbb{P}(V)$ by:

$$\begin{array}{rcl} (s,t) \in P \circ Q & \equiv & \exists(u\colon\ u \in \mathbb{U}\colon\ (s,u) \in P \wedge (u,t) \in Q) \\ (s,t) \in P^{\cup} & \equiv & (t,s) \in P \\ (s,t) \in I & \equiv & s\ =\ t \end{array}$$

As is usual we shall write $x \langle R \rangle y$ instead of $(x,y) \in R$.

With these definitions, $\mathbb{P}(\mathbb{U} \times \mathbb{U})$ satisfies $\mathcal{PCR}c$. For satisfaction of $\mathcal{M}$, we calculate:

$$\begin{array}{ll} & \neg X\ \sqsupseteq\ P\ \circ\ \neg Y\ \circ\ Q \\ \equiv & \qquad \{\ \text{definitions of } \circ \text{ and } \sqsupseteq\ \} \\ & \forall(s,t\colon\ \exists(u,v ::\ s \langle P \rangle u \wedge \neg(u \langle Y \rangle v) \wedge v \langle Q \rangle t)\colon\ \neg(s \langle X \rangle t)) \\ \equiv & \qquad \{\ \text{predicate calculus}\ \} \\ & \forall(s,t,u,v\colon\ s \langle P \rangle u \wedge \neg(u \langle Y \rangle v) \wedge v \langle Q \rangle t\colon\ \neg(s \langle X \rangle t)) \\ \equiv & \qquad \{\ \text{predicate calculus}\ \} \\ & \forall(u,v\colon\ \exists(s,t ::\ s \langle P \rangle u \wedge s \langle X \rangle t \wedge v \langle Q \rangle t)\colon\ u \langle Y \rangle v) \\ \equiv & \qquad \{\ \text{definition of } {\cup}\ \} \\ & \forall(u,v\colon\ \exists(s,t ::\ u \langle P^{\cup} \rangle s \wedge s \langle X \rangle t \wedge t \langle Q^{\cup} \rangle v)\colon\ u \langle Y \rangle v) \\ \equiv & \qquad \{\ \text{definitions of } \circ \text{ and } \sqsupseteq\ \} \\ & Y\ \ \sqsupseteq\ \ P^{\cup}\ \circ\ X\ \circ\ Q^{\cup} \end{array}$$

In conclusion:

**C2** *The binary relations over some set $\mathbb{U}$ satisfy $\mathcal{PCRM}c$.*

## C.3 Wp and wlp Pairs

The generalised statements, or, equivalently, the wp and wlp-pairs [36] without the law of the excluded miracle, are obtained by choosing

$$V \quad = \quad S \times (S \cup \{\perp\})$$

where $S$ is the statespace and "$\perp$" represents non-termination; and

$$(s,t)\ \in\ P \circ Q \quad \equiv \quad (t = \perp\ \ \wedge\ \ s \langle P \rangle t)\ \vee\ \exists(u ::\ s \langle P \rangle u \wedge u \langle Q \rangle t)$$

Then $\mathbb{P}(V)$ satisfies $\mathcal{P}$ and $\mathcal{C}$ but neither the $\mathcal{PC}$ interface nor $c$, since:

$$\begin{aligned} (S \times \{\perp\}) \circ \emptyset &= S \times \{\perp\} \\ V \circ (S \times \{\perp\}) \circ V &= S \times \{\perp\} \end{aligned}$$

That is:

**C3** *The predicate transformer pairs satisfy $\mathcal{P}$ and $\mathcal{C}$, but not $\mathcal{PC}$.*

## C.4 Monoids and Groups

A guaranteed way to obtain a model that satisfies $\mathcal{PC}$ is to choose, for a monoid $(M, \oplus, 1_\oplus)$,

$$\begin{aligned} V &= M \\ P \circ Q &= \{p, q :\ p \in P \wedge q \in Q :\ p \oplus q\} \\ I &= \{1_\oplus\} \end{aligned}$$

Satisfaction of $\mathcal{C}$ is straightforward and of the $\mathcal{PC}$ interface follows from the pointwise definition of composition. In other words:

**C4** *The powerset of a monoid satisfies $\mathcal{PC}$.*

If the monoid contains elements $x$ and $y$ such that $x \oplus y = 1_\oplus$ and $y \oplus x \neq 1_\oplus$ then $\mathbb{P}(M)$ fails to satisfy the divergence rule, (B8) which we remarked was an instance of the middle exchange rule. (Choose $P = \{y\}$ and $Q = \{x\}$.) The conclusion we reach from the consideration of C4 is thus:

**C5** *$\mathcal{PCRM}$ is a non-conservative extension of $\mathcal{PCR}$.*

For a group $(G, \oplus, 1_\oplus, ^{-1})$, with the construction above, $\mathbb{P}(G)$ satisfies $\mathcal{PC}c$, since

$$\text{(C6)} \quad \{g\} \circ G = G$$

An obvious way to define reverse is

$$P^\cup = \{p^{-1} | p \in P\}$$

With this definition, $\mathbb{P}(G)$ satisfies $\mathcal{PCRM}c$. For $\mathcal{M}$:

$$
\begin{array}{ll}
 & \neg X \;\sqsupseteq\; P \circ \neg Y \circ Q \\
\equiv & \qquad \{ \text{ definitions of } \sqsupseteq \text{ and } \circ \text{ } \} \\
 & \forall(p,y,q,x:\; p \in P \wedge y \notin Y \wedge q \in Q \wedge x \in X:\; p \oplus y \oplus q \neq x) \\
\equiv & \qquad \{ \text{ group calculus } \} \\
 & \forall(p,y,q,x:\; p \in P \wedge x \in X \wedge q \in Q \wedge y \notin Y:\; y \neq p^{-1} \oplus x \oplus q^{-1}) \\
\equiv & \qquad \{ \text{ definition of } {}^{\cup}\text{, first step backwards } \} \\
 & Y \;\sqsupseteq\; P^{\cup} \circ X \circ Q^{\cup}
\end{array}
$$

Since (C6) does not hold for binary relations, this shows that:

**C7** *$\mathcal{PCRM}c$ is not a complete characterisation of binary relations.*

Finally, we construct a model that satisfies $\mathcal{PCR}c$ but does not satisfy $\mathcal{M}$. (Thanks are due here to C.S.Scholten for simplifying our original construction.) As in the former example, we take a group in order to guarantee satisfaction of $\mathcal{PC}c$. But we differ from the former in the definition of reverse which we simply define as the identity function. Clearly, therefore, $\mathcal{PCR}c$ is satisfied. Now suppose we choose a group with two elements $a$ and $b$ such that $a \oplus b \;=\; 1_{\oplus}$ and $b \oplus b \;\neq\; 1_{\oplus}$. (A concrete example would be the natural numbers under addition modulo 3 with $a = 1$ and $b = 2$.) We claim that $\mathbb{P}(G)$ does not satisfy the middle exchange rule: Assign $P,\ Q,\ X,\ Y \;:=\; \{a\},\ \{1_{\oplus}\},\ \{b\},\ \{1_{\oplus}\}$. Then

$$Y \;=\; P^{\cup} \circ X \circ Q^{\cup}$$

but $b \;\notin\; \neg X$ whereas $b \oplus b \;\in\; \neg Y$ and, hence,

$$b \;=\; a \oplus b \oplus b \;\in\; P \circ \neg Y \circ Q$$

That is,

$$\neg X \quad \not\sqsupseteq \quad P \circ \neg Y \circ Q$$

We conclude:

**C8** *The middle exchange and cone rules are independent in the context of $\mathcal{PCR}$.*

# D Basic Properties

We now turn to the proofs of the basic properties used throughout the paper. The plat calculus is used extensively but, since it is (or should be!) well-known, we shall use it without further ado, often giving as hint the bland statement "plat calculus". Occasionally we provide a little more assistance by way of the following hints. (Note, however, that where such a hint is given the rule named is usually not the only element of the plat calculus that is required to verify the step.)

| | |
|---|---|
| shunting rule | $R \;\sqsupseteq\; S \sqcap T \quad\equiv\quad R \sqcup \neg S \;\sqsupseteq\; T$ |
| excluded middle | $\top\!\!\top \;=\; R \sqcup \neg R$ |
| contradiction | $\perp\!\!\perp \;=\; R \sqcap \neg R$ |

"Distributivity" and "monotonicity" are also hints that we occasionally supply: the former can refer to the distributivity properties of any of the three operators $\sqcup$, $\sqcap$ or $\neg$ with respect to each other. Similarly, the latter can refer to the monotonicity property of any of these three operators (in the case of $\neg$ anti-monotonicity, of course). Which is intended should be clear from the context.

We shall not assume the same level of familiarity with the $\mathcal{C}$, $\mathcal{R}$ and $\mathcal{M}$ calculi and their interfaces; accordingly the proof steps we take will be smaller. Sometimes, "$\mathcal{PC}$ interface" or "$\mathcal{PR}$ interface" is given as a hint; at other times we use the following terminology.

| | |
|---|---|
| bottom strictness | $\perp\!\!\perp \circ R \;=\; R \circ \perp\!\!\perp \;=\; \perp\!\!\perp$ |
| monotonicity | $R \sqsupseteq S \;\Rightarrow\; R \circ T \sqsupseteq S \circ T$ |
| distributivity | e.g. $R \circ S \;=\; ((R \sqcap T) \circ S) \sqcup ((R \sqcap \neg T) \circ S)$ |

(Again we would remark that to which operator the hint "monotonicity" or "distributivity" refers should be evident from the context.)

## D.1 Properties of Monotypes

In this section we consider the properties of monotypes quoted in section 10. First the non-trivial elements of properties (10.1) and (10.2) are proven.

**Lemma D0** Let $I \sqsupseteq A$ and $I \sqsupseteq B$. Then

(a) $\quad A \circ B \;=\; A \sqcap B$,
(b) $\quad A^{\cup} \;=\; A$

**Proof** As a preliminary we note that

$$
\begin{array}{ll}
 & I \sqsupseteq A^{\cup} \\
\equiv & \quad \{ \; I = I^{\cup} \; \} \\
 & I^{\cup} \sqsupseteq A^{\cup} \\
\Leftarrow & \quad \{ \; \text{monotonicity} \; \} \\
 & I \sqsupseteq A
\end{array}
$$

Assume $I \sqsupseteq A$ and $I \sqsupseteq B$. Then,

$$
\begin{array}{ll}
 & A \circ B = A \sqcap B \\
\equiv & \quad \{ \; \text{assumption, monotonicity} \; \} \\
 & A \circ B \sqsupseteq A \sqcap B \\
\equiv & \quad \{ \; I \text{ is the unit of composition} \; \} \\
 & A \circ B \sqsupseteq A \sqcap I \circ B \\
\Leftarrow & \quad \{ \; \text{Dedekind's rule: B11(a)} \; \} \\
 & A \sqsupseteq A \circ B^{\cup} \sqcap I \\
\Leftarrow & \quad \{ \; \text{above, monotonicity} \; \} \\
 & I \sqsupseteq A \wedge I \sqsupseteq B
\end{array}
$$

and

$$
\begin{array}{ll}
 & A^{\cup} = A \\
\equiv & \quad \{ \; \text{equality; } {}^{\cup} \text{ is its own inverse} \; \} \\
 & A \sqsupseteq A^{\cup} \quad \wedge \quad A^{\cup} \sqsupseteq A^{\cup\cup} \\
\equiv & \quad \{ \; \text{monotonicity of } {}^{\cup} \; \} \\
 & A \sqsupseteq A^{\cup} \\
\Leftarrow & \quad \{ \; I \sqsupseteq A^{\cup} \; \} \\
 & A^{\cup} \circ A \sqsupseteq A^{\cup} \\
\equiv & \quad \{ \; I \sqsupseteq A^{\cup} \; \} \\
 & A^{\cup} \circ A \sqsupseteq A^{\cup} \circ I \sqcap I \\
\Leftarrow & \quad \{ \; \text{Dedekind's rule: B11(b)} \; \} \\
 & A \sqsupseteq I \sqcap A^{\cup\cup} \circ I \\
\equiv & \quad \{ \; \mathcal{R} \; \} \\
 & A \sqsupseteq I \sqcap A \\
\equiv & \quad \{ \; \mathcal{P} \; \} \\
 & \mathbf{true}
\end{array}
$$

□

**Corollary D1** Every monotype is an imp and a co-imp.

**Proof** Assume $I \sqsupseteq A$. Then

$$
\begin{array}{ll}
 & I \ \sqsupseteq \ A \circ A^{\cup} \ \sqcup \ A^{\cup} \circ A \\
\equiv & \quad \{ \text{ lemma D0 and plat calculus } \} \\
 & I \ \sqsupseteq \ A \circ A \\
\equiv & \quad \{ \text{ assumption, monotonicity } \} \\
 & \textbf{true}
\end{array}
$$

Thus, $I \sqsupseteq A \circ A^{\cup}$ and $I \sqsupseteq A^{\cup} \circ A$. In words, $A$ is an imp and a co-imp.
□

**Theorem D2**

$$I \sqsupseteq A \wedge I \sqsupseteq B \quad \Rightarrow \quad (A \sqcap B) \circ R = (A \circ R) \sqcap (B \circ R)$$

**Proof** Assume $I \sqsupseteq A \wedge I \sqsupseteq B$. Then

$$
\begin{array}{ll}
 & (A \sqcap B) \circ R = A \circ R \sqcap B \circ R \\
\equiv & \quad \{ \text{ monotonicity } \} \\
 & (A \sqcap B) \circ R \sqsupseteq A \circ R \sqcap B \circ R \\
\equiv & \quad \{ \text{ assumption, D0(a) } \} \\
 & A \circ B \circ R \sqsupseteq A \circ R \sqcap B \circ R \\
\Leftarrow & \quad \{ \text{ Dedekind's rule: B11(b) } \} \\
 & B \circ R \sqsupseteq A^{\cup} \circ B \circ R \sqcap R \\
\equiv & \quad \{ \text{ assumption, lemma D0(b) and calculus } \} \\
 & \textbf{true}
\end{array}
$$

□

In general it is difficult to give an expression for $\neg(S \circ R)$ in terms of $\neg S$ and $\neg R$. By (D2), this is possible in the case that $S$ (or, dually, $R$) is a monotype:

**Lemma D3** Let $I \sqsupseteq A$. Then

$$\neg(A \circ R) \quad = \quad (I \sqcap \neg A) \circ R \sqcup \neg R$$

**Proof** The proof is based on the identity

$$\neg X = Y \quad \equiv \quad X \sqcap Y = \perp\!\!\perp \quad \wedge \quad X \sqcup Y = \top\!\!\top$$

The two conjuncts are, first,

$A \circ R \;\sqcap\; ((I \sqcap \neg A) \circ R \;\sqcup\; \neg R)$

$=$ { distributivity }

$(A \circ R \;\sqcap\; (I \sqcap \neg A) \circ R) \;\sqcup\; (A \circ R \;\sqcap\; \neg R)$

$=$ { corollary D2 }

$(A \sqcap I \sqcap \neg A) \circ R \;\sqcup\; (A \circ R \;\sqcap\; \neg R)$

$=$ { contradiction; bottom strictness }

$A \circ R \;\sqcap\; \neg R$

$=$ { $R \sqsupseteq A \circ R$, contradiction }

$\perp\!\!\perp$

and, second,

$A \circ R \;\sqcup\; (I \sqcap \neg A) \circ R \;\sqcup\; \neg R$

$=$ { distributivity of composition }

$(A \sqcup (I \sqcap \neg A)) \circ R \;\sqcup\; \neg R$

$=$ { plat calculus; assumption }

$I \circ R \;\sqcup\; \neg R$

$=$ { $I$ is unit of composition; excluded middle }

$\top\!\!\top$

□

## D.2 Left and Right Domains

****To be revised*****

The subject matter of this section is the verification of those properties of left and right domains stated in section 10.1 of the paper. In fact, we only consider the properties of left domains since it is obvious that all our proofs can be dualised (by reversing the order of all compositions) to encompass right domains. First we establish the equivalence of the two modes of definition of $R<$ of section 10.1, and simultaneously introduce a third definition. By way of preparation we have the following lemma.

**Lemma D4** Let $I \sqsupseteq A$. Then

$$A \circ R = R \quad \equiv \quad A \;\sqsupseteq\; I \sqcap R \circ \top\!\!\top$$

**Proof**

$$
\begin{array}{ll}
 & A \;\sqsupseteq\; I \sqcap R \circ \top\!\!\top \\
\equiv & \quad \{ \text{ shunting rule } \} \\
 & \neg I \sqcup A \;\sqsupseteq\; R \circ \top\!\!\top \\
\equiv & \quad \{ \text{ rotation rule } \} \\
 & \neg \top\!\!\top^{\cup} \;\sqsupseteq\; \neg(\neg I \sqcup A)^{\cup} \circ R \\
\equiv & \quad \{ \ \mathcal{PR} \text{ and } \mathcal{CR} \text{ interfaces, distributivity and (D0b) } \} \\
 & \bot\!\!\bot \;\sqsupseteq\; (I \sqcap \neg A) \circ R \\
\equiv & \quad \{ \text{ lemma D3, plat calculus } \} \\
 & \neg(A \circ R) \;=\; \neg R \\
\equiv & \quad \{ \text{ plat calculus } \} \\
 & A \circ R \;=\; R
\end{array}
$$

□

**Theorem D5** The following three statements are equivalent:

$$
\begin{array}{l}
X \;=\; I \sqcap R \circ R^{\cup} \\
X \;=\; I \sqcap R \circ \top\!\!\top \\
\forall(A\colon I \sqsupseteq A\colon\ A \circ R = R \;\equiv\; A \sqsupseteq X)
\end{array}
$$

**Proof** We begin by establishing the equivalence of the first two statements.

$$
\begin{array}{ll}
 & I \sqcap R \circ R^{\cup} \;=\; I \sqcap R \circ \top\!\!\top \\
\equiv & \quad \{ \text{ excluded middle and } \mathcal{PC} \text{ interface } \} \\
 & I \sqcap R \circ R^{\cup} \;=\; I \sqcap (R \circ R^{\cup} \sqcup R \circ \neg R^{\cup}) \\
\Leftarrow & \quad \{ \text{ plat calculus } \} \\
 & \bot\!\!\bot \;\sqsupseteq\; I \sqcap R \circ \neg R^{\cup} \\
\equiv & \quad \{ \text{ shunting rule } \} \\
 & \neg I \;\sqsupseteq\; R \circ \neg R^{\cup} \\
\equiv & \quad \{ \text{ rotation rule } \} \\
 & R^{\cup} \;\sqsupseteq\; R^{\cup} \\
\equiv & \quad \{ \text{ plat calculus } \} \\
 & \textbf{true}
\end{array}
$$

Now we prove the equivalence of the second and third statements. To do so, let us consider the equation

$$
X :: \qquad \forall(A\colon I \sqsupseteq A\colon\ A \circ R = R \;\equiv\; A \sqsupseteq X)
$$

By lemma D4, a solution of this equation exists: $I \sqcap (R \circ \top\!\top)$. Let $X$ be an arbitrary solution. Then $I \sqsupseteq X$. Moreover, for any $Y$ such that $I \sqsupseteq Y$ we have

$$\begin{array}{ll} & Y \sqsupseteq X \\ \equiv & \quad \{ X \text{ is a solution} \} \\ & Y \circ R \;=\; R \\ \equiv & \quad \{ \text{lemma D4} \} \\ & Y \;\sqsupseteq\; I \sqcap (R \circ \top\!\top) \end{array}$$

Hence, $\quad X \;=\; I \sqcap (R \circ \top\!\top) \;.$

□

By theorem D5 we are free to define $R<$ by

(D6) $\quad R< \;=\; I \;\sqcap\; R \circ R\cup$

(D7) $\quad R< \;=\; I \;\sqcap\; R \circ \top\!\top$

or by

(D8) $\quad \forall(A\colon\ A \sqsubseteq I\colon\ A \circ R = R \;\equiv\; A \sqsupseteq R<)$

The right domain operator is similarly defined.

A few easy properties of $<$ are summarised in the next theorem.

**Theorem D9**

(a) $\quad (\sqcup(R\colon\ R \in V\colon\ R))< \;=\; \sqcup(R\colon\ R \in V\colon\ R<)$
for arbitrary set of specs $V$. In particular: $<$ is monotonic.

(b) $\quad I \sqsupseteq A \;\Rightarrow\; A = A<$

(c) $\quad (R\cup)< \;=\; R>$

(d) $\quad R = \perp\!\!\perp \;\equiv\; R< = \perp\!\!\perp \;.$

**Proof** Straightforward calculation. (For (a) use definition (D7), for (b) use definition (D6) and for (c) either of these two together with the corresponding dual definition of $R>$ . For (d) use (D8).)

□

**Theorem D10**

$$(R \sqcap S)< \;=\; I \;\sqcap\; R \circ S\cup$$

**Proof** The proof establishes mutual inclusion. First,

$$
\begin{array}{ll}
 & (R \sqcap S)^{<} \\
= & \quad \{ \text{ definition } \} \\
 & I \sqcap (R \sqcap S) \circ (R \sqcap S)^{\cup} \\
\sqsubseteq & \quad \{ \text{ monotonicity } \} \\
 & I \sqcap R \circ S^{\cup}
\end{array}
$$

Second,

$$
\begin{array}{ll}
 & (R \sqcap S)^{<} \sqsupseteq I \sqcap R \circ S^{\cup} \\
\equiv & \quad \{ \text{ definition } \} \\
 & I \sqcap (R \sqcap S) \circ \top\!\top \sqsupseteq I \sqcap R \circ S^{\cup} \\
\equiv & \quad \{ \text{ calculus } \} \\
 & (R \sqcap S) \circ \top\!\top \sqsupseteq I \sqcap R \circ S^{\cup} \\
\Leftarrow & \quad \{ \text{ Dedekind's rule: B11(c) } \} \\
 & R \sqcap S \sqsupseteq R \sqcap I \circ S^{\cup\cup} \;\wedge\; \top\!\top \sqsupseteq R^{\cup} \circ I \sqcap S^{\cup} \\
\equiv & \quad \{ \text{ calculus } \} \\
 & \textbf{true}
\end{array}
$$

□

The following lemma is the one that we drew particular attention to in section 10.1.

**Lemma D11**

(a) $R \circ \top\!\top \;=\; R^{<} \circ \top\!\top$

(b) $(R \circ \top\!\top) \sqcap S \;=\; R^{<} \circ S$

(c) The following three statements are equivalent:

$$
\begin{array}{l}
R^{<} \sqsupseteq S^{<} \\
R \circ \top\!\top \sqsupseteq S \\
R \circ \top\!\top \sqsupseteq S \circ \top\!\top
\end{array}
$$

**Proof**

(a)
$$
\begin{array}{ll}
 & R \circ \top\!\top \\
= & \quad \{ \text{(D8)} \} \\
 & R^{<} \circ R \circ \top\!\top
\end{array}
$$

$\sqsubseteq \quad \{\ R \circ \top\!\!\top \ \sqsubseteq\ \top\!\!\top\ ,\quad \text{monotonicity}\ \}$

$\quad R_< \circ \top\!\!\top$

$\sqsubseteq \quad \{\ (\text{D7})\ \}$

$\quad R \circ \top\!\!\top \circ \top\!\!\top$

$= \quad \{\ \top\!\!\top \circ \top\!\!\top \ =\ \top\!\!\top\ \}$

$\quad R \circ \top\!\!\top$

Part (b) is an instance of theorem D15. (Make the substitution $R, S, T := I, R, S$.) Finally, we prove part (c):

$\quad R_< \ \sqsupseteq\ S_<$

$\Rightarrow \quad \{\ \top\!\!\top \ \sqsupseteq\ S\ ,\ \text{monotonicity}\ \}$

$\quad R_< \circ \top\!\!\top \ \sqsupseteq\ S_< \circ S$

$\equiv \quad \{\ (\text{a});\ (\text{D8})\ \}$

$\quad R \circ \top\!\!\top \ \sqsupseteq\ S$

$\Rightarrow \quad \{\ \top\!\!\top \circ \top\!\!\top \ =\ \top\!\!\top\ ,\ \text{monotonicity}\ \}$

$\quad R \circ \top\!\!\top \ \sqsupseteq\ S \circ \top\!\!\top$

$\Rightarrow \quad \{\ (\text{D7})\ \text{and plat calculus}\ \}$

$\quad R_< \ \sqsupseteq\ S_<$

$\Box$

Note that part (b) above appears as (10.15) in the main body of the paper, and that it subsumes part (a). (Instantiate $S$ to $\top\!\!\top$ and then simplify the resulting equation.)

We are now ready to prove (10.17) and (10.16) in section 10.1:

**Theorem D12**

(a) $\quad S_< \ \sqsupseteq\ (S \circ T)_<$

(b) $\quad (R \circ S)_< \ =\ (R \circ S_<)_<$

**Proof**

(a) $\quad S_< \ \sqsupseteq\ (S \circ T)_<$

$\equiv \quad \{\ \text{lemma D11(b)}\ \}$

$\quad S \circ \top\!\!\top \ \sqsupseteq\ S \circ T \circ \top\!\!\top$

$\equiv \quad \{\ \top\!\!\top \ \sqsupseteq\ T \circ \top\!\!\top\ ,\ \text{monotonicity}\ \}$

$\quad \textbf{true}$

(b)

$$
\begin{array}{ll}
 & (R \circ S)_< \\
= & \qquad \{ \text{ domains, (D11b) } \} \\
 & I \ \sqcap \ R \circ S \circ \top\!\top \\
= & \qquad \{ \text{ domains, (D11b) } \} \\
 & I \ \sqcap \ R \circ S_< \circ \top\!\top \\
= & \qquad \{ \text{ domains, (D11b) } \} \\
 & (R \ \circ \ S_<)_<
\end{array}
$$

□

Property (10.17) follows from (10.16) by straightforward use of the fact that $S_<$ is a monotype and monotonicity of the left domain operator.

Finally we prove theorem 10.34(a) from section 10.4: Let $F$ be a relator (see definition 10.33).

**Theorem D13** $F.(R_<) \ = \ (F.R)_<$

**Proof** The strategy is to use (D11c) to prove the mutual containment. That is, we prove:

$$
\begin{array}{ll}
 & F.R \ \circ \ \top\!\top \ \sqsubseteq \ F.(R_<) \ \circ \ \top\!\top \\
\text{and} & F.(R_<) \ \circ \ \top\!\top \ \sqsubseteq \ F.R \ \circ \ \top\!\top
\end{array}
$$

For the first of these we have:

$$
\begin{array}{ll}
 & F.R \ \circ \ \top\!\top \\
\sqsubseteq & \qquad \{ \text{ by (D11a), } R \ \sqsubseteq \ R_< \ \circ \ \top\!\top\text{, monotonicity } \} \\
 & F.(R_< \circ \top\!\top) \ \circ \ \top\!\top \\
= & \qquad \{ \text{ relators distribute through composition } \} \\
 & F.(R_<) \ \circ \ F.\top\!\top \ \circ \ \top\!\top \\
\sqsubseteq & \qquad \{ \ \mathcal{PC} \ \} \\
 & F.(R_<) \ \circ \ \top\!\top
\end{array}
$$

and for the second:

$$
\begin{array}{ll}
 & F.(R_<) \ \circ \ \top\!\top \\
\sqsubseteq & \qquad \{ \text{ (D5), properties of relators } \} \\
 & F.R \ \circ \ F.R^{\cup} \ \circ \ \top\!\top \\
\sqsubseteq & \qquad \{ \ \mathcal{PC} \ \} \\
 & F.R \ \circ \ \top\!\top
\end{array}
$$

It follows by two applications of (D11b) that

$$(F.R)_{<} \quad = \quad (F.(R_{<}))_{<}$$

But

$$\begin{array}{ll} & I \;\sqsupseteq\; F.(R_{<}) \\ \Leftarrow & \qquad \{\; I \sqsupseteq F.I,\ F \text{ is monotonic } \} \\ & I \;\sqsupseteq\; R_{<} \\ \equiv & \qquad \{\; \text{definition: (D6)} \;\} \\ & \textbf{true} \end{array}$$

Hence, by (D9b),

$$(F.(R_{<}))_{<} \quad = \quad F.(R_{<})$$

and so transitivity of equality completes the proof.
□

## D.3 Distribution of Composition over Cap

The interface between the plat calculus and the monoid structure of composition guarantees that composition is universally cup-junctive. What about cap-junctivity? In general, this is not the case but in the presence of certain classes of specs more can be said. This section documents some of those classes.

**Theorem D14**

$$\begin{array}{l} (R \sqcap S) \circ T \;=\; R \circ T \;\sqcap\; S \circ T \\ \quad \Leftarrow \quad R \sqsupseteq R \circ T \circ T_{\cup} \;\;\vee\;\; S \sqsupseteq S \circ T \circ T_{\cup} \end{array}$$

**Proof** By monotonicity,

$$(R \sqcap S) \circ T \;\sqsubseteq\; R \circ T \sqcap S \circ T$$

The task is thus to prove the other inclusion, which we do as follows:

$$
\begin{array}{ll}
 & (R \sqcap S) \circ T \ \sqsupseteq \ R \circ T \ \sqcap \ S \circ T \\
\Leftarrow & \qquad \{ \text{ Dedekind's rule: B11(a) } \} \\
 & R \sqcap S \ \sqsupseteq \ S \ \sqcap \ R \circ T \circ T^{\cup} \\
\Leftarrow & \qquad \{ \text{ calculus } \} \\
 & R \ \sqsupseteq \ R \circ T \circ T^{\cup}
\end{array}
$$

□

Theorem D14 predicts a distributivity property when one of the given specs is a “left-condition” (i.e. of the form $U \circ \top\!\top$) but a more useful property in such a case is the following:

**Theorem D15**

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

**Proof** Two applications of lemma D11(a) and monotonicity.
□

The following theorem was pointed to us by Oege de Moor [74]. We make no use of it but include it for the sake of completeness. Define

$$R \div S \quad = \quad (R \sqcap \neg S) \sqcup (\neg R \sqcap S)$$

Then we have:

**Theorem D16** If $R \div S$ is a co-imp, then

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

Note: There must surely be a better proof than this one!
**Proof** First we rewrite the right-hand side of the claimed equality as follows:

$$
\begin{array}{ll}
 & (R \circ T) \ \sqcap \ (S \circ T) \\
= & \qquad \{ \text{ distributivity and excluded middle } \} \\
 & (((R \sqcap S) \ \circ \ T) \sqcup ((R \sqcap \neg S) \ \circ \ T)) \\
 & \sqcap (((S \sqcap R) \ \circ \ T) \sqcup ((S \sqcap \neg R) \ \circ \ T)) \\
= & \qquad \{ \text{ plat calculus } \} \\
 & ((R \sqcap S) \ \circ \ T) \sqcup (((R \sqcap \neg S) \ \circ \ T) \sqcap ((S \sqcap \neg R) \ \circ \ T))
\end{array}
$$

Proceeding with this new right-hand side, we have

$$
\begin{array}{ll}
 & (R \sqcap S) \circ T \\
 & = \;\; ((R \sqcap S) \circ T) \sqcup (((R \sqcap \neg S) \circ T) \sqcap ((S \sqcap \neg R) \circ T)) \\
\Leftarrow & \qquad \{ \text{ plat calculus } \} \\
 & \perp\!\!\perp \;\; \sqsupseteq \;\; ((R \sqcap \neg S) \circ T) \sqcap ((S \sqcap \neg R) \circ T) \\
\Leftarrow & \qquad \{ \text{ plat calculus } \} \\
 & \neg((\neg R \sqcap S) \circ T) \;\; \sqsupseteq \;\; (R \sqcap \neg S) \circ T \\
\equiv & \qquad \{ \text{ middle exchange rule } \} \\
 & \neg I \;\; \sqsupseteq \;\; (R \sqcap \neg S)^{\cup} \circ (\neg R \sqcap S) \circ T \circ T^{\cup} \\
\Leftarrow & \qquad \{ \text{ bottom strictness } \} \\
 & \perp\!\!\perp \;\; \sqsupseteq \;\; (R \sqcap \neg S)^{\cup} \circ (\neg R \sqcap S) \\
\equiv & \qquad \{ \qquad I \\
 & \qquad\qquad \sqsupseteq \qquad \{ \text{ assumption } \} \\
 & \qquad\qquad\quad (R \div S)^{\cup} \circ (R \div S) \\
 & \qquad\qquad \sqsupseteq \qquad \{ \text{ definition of } \div, \mathcal{PC} \text{ and } \mathcal{PR} \text{ interfaces } \} \\
 & \qquad\qquad\quad (R \sqcap \neg S)^{\cup} \circ (\neg R \sqcap S) \\
 & \qquad \} \\
 & \neg I \;\; \sqsupseteq \;\; (R \sqcap \neg S)^{\cup} \circ (\neg R \sqcap S) \\
\equiv & \qquad \{ \text{ rotation rule, de Morgan } \} \\
 & \neg R \sqcup S \;\; \sqsupseteq \;\; \neg R \sqcap S \\
\equiv & \qquad \{ \text{ plat calculus } \} \\
 & \mathbf{true}
\end{array}
$$

□

Clearly, theorem D16 can be dualized to:

(D17) $T \circ (S \sqcap R) \;\; = \;\; (T \circ S) \sqcap (T \circ R)$

if $R \div S$ is an imp.

**Theorem D18**

(a) $f$ is a co-imp $\Rightarrow$ $(f\circ)$ is positively cap-junctive

(b) $f$ is an imp $\Rightarrow$ $(\circ f)$ is positively cap-junctive.

**Proof** We prove (a); part (b) follows by duality. Within the proof we make extensive use of the terminology of definition 10.36 in section 14.2 which, if it is not already familiar, the reader may wish to consult.

First, we remark that

$$\text{(D19)} \quad \sqcap_{\mathcal{I}}(G \bullet \mathcal{R}) \;\sqsupseteq\; G.(\sqcap_{\mathcal{I}}\mathcal{R})$$

for all monotonic functions $G$, sets $\mathcal{I}$ and $\mathcal{I}$-bags $\mathcal{R}$. Now suppose $\mathcal{I}$ is a non-empty set, $\mathcal{R}$ is an $\mathcal{I}$-bag and $f$ is a co-imp. Since $(f\circ)$ is monotonic, property (D19) reduces the problem to showing that

$$f \circ \sqcap_{\mathcal{I}}\mathcal{R} \;\sqsupseteq\; \sqcap_{\mathcal{I}}((f\circ) \bullet \mathcal{R})$$

But,

$$\begin{array}{ll}
& f \circ \sqcap_{\mathcal{I}}\mathcal{R} \\
\sqsupseteq & \quad \{\ f \text{ is a co-imp, monotonicity } \} \\
& f \circ \sqcap_{\mathcal{I}}(((f{\cup} \circ f)\circ) \bullet \mathcal{R}) \\
\sqsupseteq & \quad \{\ \text{(D19) with } G := (f{\cup}\circ),\ \mathcal{R} := (f\circ) \bullet \mathcal{R}\ \} \\
& f \circ f{\cup} \circ \sqcap_{\mathcal{I}}((f\circ) \bullet \mathcal{R}) \\
\sqsupseteq & \quad \{\ \text{definition of } f{<}\ \} \\
& f{<} \circ \sqcap_{\mathcal{I}}((f\circ) \bullet \mathcal{R}) \\
= & \quad \{\ f{<} \sqsupseteq \{\text{see below}\}\ (\sqcap_{\mathcal{I}}((f\circ) \bullet \mathcal{R})){<}\ ;\ (D8)\ \} \\
& \sqcap_{\mathcal{I}}((f\circ) \circ \mathcal{R})
\end{array}$$

In the last step we claimed that

$$f{<} \;\sqsupseteq\; (\sqcap_{\mathcal{I}}((f\circ) \bullet \mathcal{R})){<}$$

The proof is

$$\begin{array}{ll}
& (\sqcap_{\mathcal{I}}((f\circ) \bullet \mathcal{R})){<} \\
\sqsubseteq & \quad \{\ \text{(D19)},\ G := {<},\ \mathcal{R} := (f\circ) \bullet \mathcal{R}\ \} \\
& \sqcap_{\mathcal{I}}({<} \bullet (f\circ) \bullet \mathcal{R}) \\
= & \quad \{\ \text{definition}\ \} \\
& \sqcap(i : i \in \mathcal{I} : (f \circ \mathcal{R}.i){<}) \\
\sqsubseteq & \quad \{\ \text{(D12a), monotonicity}\ \} \\
& \sqcap(i : i \in \mathcal{I} : f{<}) \\
= & \quad \{\ \mathcal{I} \text{ is non-empty, plat calculus}\ \} \\
& f{<}
\end{array}$$

□

Here is another condition under which composition distributes over cap.

**Theorem D20**

$$\begin{array}{ll} & (R \sqcap S) \circ T \;=\; R \circ T \;\sqcap\; S \circ T \\ \Leftarrow & (R \sqcap S)^{<} \;\sqsupseteq\; (R \circ T \sqcap S \circ T)^{<} \\ & \wedge \;\; T \;\sqsupseteq\; R^{\cup} \circ R \circ T \;\;\wedge\;\; T \;\sqsupseteq\; S^{\cup} \circ S \circ T \end{array}$$

**Proof**

$$\begin{array}{ll} & (R \sqcap S) \circ T \;=\; R \circ T \sqcap S \circ T \\ \equiv & \qquad \{ \text{ monotonicity } \} \\ & (R \sqcap S) \circ T \;\sqsupseteq\; R \circ T \sqcap S \circ T \\ \equiv & \qquad \{ \text{ assumption: } (R \sqcap S)^{<} \;\sqsupseteq\; (R \circ T \sqcap S \circ T)^{<}, \text{ (D8) } \} \\ & (R \sqcap S) \circ T \;\sqsupseteq\; (R \sqcap S)^{<} \circ (R \circ T \sqcap S \circ T) \\ \Leftarrow & \qquad \{ \text{ domains: (D6) } \} \\ & (R \sqcap S) \circ T \;\sqsupseteq\; (R \sqcap S) \circ (R \sqcap S)^{\cup} \circ (R \circ T \sqcap S \circ T) \\ \Leftarrow & \qquad \{ \text{ monotonicity } \} \\ & T \;\sqsupseteq\; (R \sqcap S)^{\cup} \circ (R \circ T \sqcap S \circ T) \\ \Leftarrow & \qquad \{ \text{ monotonicity } \} \\ & T \;\sqsupseteq\; (R \sqcap S)^{\cup} \circ R \circ T \;\;\wedge\;\; T \;\sqsupseteq\; (R \sqcap S)^{\cup} \circ S \circ T \\ \Leftarrow & \qquad \{ \text{ monotonicity } \} \\ & T \;\sqsupseteq\; R^{\cup} \circ R \circ T \;\;\wedge\;\; T \;\sqsupseteq\; S^{\cup} \circ S \circ T \end{array}$$

□

## D.4 Two Theorems Concerning Reverse

Verifying that one spec is the reverse of another is also a task that frequently occurs. In this section we establish two lemmas that assist in this task. The first is well-known and included for completeness sake. The second may not be so familiar.

**Theorem D21** Suppose $f \in A \longleftarrow B$ and $g \in B \longleftarrow A$. Then,

$$A \sqsupseteq f \circ g \;\;\wedge\;\; B \sqsupseteq g \circ f \quad \equiv \quad f^{\cup} \;=\; g$$

**Proof** The proof of "follows-from" is a straightforward application of definition 10.32. We content ourselves, therefore, with just the proof of the implication. To that end assume the given premises and the left-hand side of the claimed equivalence. The statement of the theorem is symmetrical in $f$ and $g$; it suffices therefore to show that $f^{\cup} \sqsupseteq g$. This is accomplished as follows.

$$
\begin{array}{ll}
 & f^{\cup} \\
= & \quad \{ \ f \in A \longleftarrow B \ \} \\
 & f^{\cup} \circ A \\
\sqsupseteq & \quad \{ \ A \sqsupseteq f \circ g \ \} \\
 & f^{\cup} \circ f \circ g \\
\sqsupseteq & \quad \{ \ f \in A \longleftarrow B \ \} \\
 & B \circ g \\
= & \quad \{ \ g \in B \longleftarrow A \ \} \\
 & g
\end{array}
$$

□

**Lemma D22** Suppose $A$ and $B$ are symmetric specs (i.e. $A^{\cup} = A$ and $B^{\cup} = B$). Suppose also that $R$ and $S$ are arbitrary specs related by the properties:

(a) $R \circ A = R$
(b) $S \circ B = S$
(c) $R \circ S = B$
(d) $S \circ R = A$

Then $R = S^{\cup}$.

**Proof**

$$
\begin{array}{ll}
 & R \\
= & \quad \{ \ \text{symmetric}.A \text{ , (a) } \} \\
 & R \circ A^{\cup} \\
= & \quad \{ \ \text{(d)} \ \} \\
 & R \circ R^{\cup} \circ S^{\cup} \\
\sqsupseteq & \quad \{ \ \text{domains: } (\textbf{??}) \ \} \\
 & R{<} \circ S^{\cup} \\
\sqsupseteq & \quad \{ \ \text{(c), domains: (10.17)} \ \} \\
 & B{<} \circ S^{\cup} \\
= & \quad \{ \ \text{domains: (10.19)} \ \} \\
 & (S \circ B{<})^{\cup} \\
= & \quad \{ \ \text{(b)} \ \} \\
 & (S \circ B \circ B{<})^{\cup} \\
= & \quad \{ \ \text{domains: (10.16), (b)} \ \} \\
 & S^{\cup}
\end{array}
$$

Swapping $A$ with $B$ and $R$ with $S$ the opposite inclusion is obtained and thus $R$ and $S$ are equal.
□

# E Solutions

## Natural Isomorphisms

The following is a list of the natural simulations discussed in section 12.5.

$\perp\!\!\!\perp \;\in\; \lambda(R ::\quad \perp\!\!\!\perp) \cong \lambda(R ::\quad R \times \perp\!\!\!\perp)$
$\hookrightarrow \;\in\; \lambda(R ::\quad R{+}\perp\!\!\!\perp) \cong \lambda(R ::\quad R)$
$\hookleftarrow \,\triangledown\, \hookrightarrow \;\in\; \lambda(R, S ::\quad R{+}S) \cong \lambda(R, S ::\quad S{+}R)$
$(I + \hookrightarrow) \;\triangledown\; (\hookrightarrow \circ \hookrightarrow)$
$\qquad \in\; \lambda(R, S, T ::\quad R{+}(S{+}T)) \cong \lambda(R, S, T ::\quad (R{+}S){+}T)$
$\gg \vartriangle \ll \;\in\; \lambda(R, S ::\quad R \times S) \cong \lambda(R, S ::\quad S \times R)$
$(\ll \circ \ll) \;\vartriangle\; (\gg \times I)$
$\qquad \in\; \lambda(R, S, T ::\quad R \times (S \times T)) \cong \lambda(R, S, T ::\quad (R \times S) \times T)$
$\ll \circ\, I \times \mathbb{1} \;\in\; \lambda(R ::\quad R) \cong \lambda(R ::\quad R \times \mathbb{1})$
$(I \times \hookrightarrow) \;\triangledown\; (I \times \hookleftarrow)$
$\qquad \in\; \lambda(R, S, T ::\quad R \times (S{+}T)) \cong \lambda(R, S, T ::\quad (R \times S){+}(R \times T))$

# Bibliography


[1] R.C. Backhouse. Naturality of homomorphisms. Lecture notes, International Summer School on Constructive Algorithmics, vol. 3, 1989.

[2] R.C. Backhouse. *Closure algorithms and the star-height problem of regular languages.* PhD thesis, University of London, 1975.

[3] R.C. Backhouse. *Program Construction and Verification.* Prentice-Hall International, 1986.

[4] R.C. Backhouse. An exploration of the Bird-Meertens formalism. Technical Report CS8810, Department of Mathematics and Computing Science, University of Groningen, 1988.

[5] R.C. Backhouse. Making formality work for us. *EATCS Bulletin*, 38:219–249, June 1989.

[6] 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.

[7] R.C. Backhouse and B.A. Carré. Regular algebra applied to path-finding problems. *Journal of the Institute of Mathematics and its Applications*, 15:161–186, 1975.

[8] R.C. Backhouse, P. Chisholm, G. Malcolm, and E. Saaman. Do-it-yourself type theory. *Formal Aspects of Computing*, 1:19–84, 1989.

[9] R.C. Backhouse and R.K. Lutz. Factor graphs, failure functions and bi–trees. In A. Salomaa and M. Steinby, editors, *Fourth Colloquium on Automata, Languages and Programming*, pages 61–75. Springer-Verlag, LNCS 52, July 1977.

[10] Roland Backhouse and A.J.M. van Gasteren. Calculating a path algorithm. To appear: 2nd International Conference on the Mathematics of Program Construction, 1992.

[11] J. Backus. Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *Communications of the ACM*, 21(8):613–641, August 1978.

[12] J.W. de Bakker and W.P. de Roever. A calculus for recursive program schemes. In M. Nivat, editor, *Proc. IRIA Symp. on Automata, Formal Languages and Programming*. North-Holland, Amsterdam, 1972.

[13] M. Barr and C. Wells. *Toposes, Triples and Theories*. Springer-Verlag, 1985.

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

[15] R.S. Bird. The promotion and accumulation strategies in transformational programming. *ACM. Transactions on Programming Languages and Systems*, 6(4):487–504, 1984.

[16] R.S. Bird. Transformational programming and the paragraph problem. *Science of Computing Programming*, 6:159–189, 1986.

[17] R.S. Bird. An introduction to the theory of lists. In M. Broy, editor, *Logic of Programming and Calculi of Discrete Design*. Springer-Verlag, 1987. NATO ASI Series, vol. F36.

[18] R.S. Bird. A calculus of functions for program derivation. Technical report, Programming Research Group, Oxford University, 11, Keble Road, Oxford, OX1 3QD, U.K., 1988.

[19] R.S. Bird. Lectures on constructive functional programming. In M. Broy, editor, *Constructive Methods in Computing Science*, pages 151–216. Springer-Verlag, 1989. NATO ASI Series, vol. F55.

[20] R.S. Bird, J. Gibbons, and G. Jones. Formal derivation of a pattern matching algorithm. Technical report, Programming Research Group, Oxford University, 11, Keble Road, Oxford, OX1 3QD, U.K., 1988.

[21] R.S. Bird and L. Meertens. Two exercises found in a book on algorithmics. In L.G.L.T. Meertens, editor, *Program Specification and Transformations*, pages 451–457. Elsevier Science Publishers B.V., North Holland, 1987.

[22] R.S. Bird and P. Wadler. *Introduction to Functional Programming.* Prentice-Hall, 1988.

[23] Garrett Birkhoff. *Lattice Theory.* American Mathematical Society, Providence, Rhode Island, revised edition, 1948.

[24] Garrett Birkhoff. *Lattice Theory*, volume 25 of *American Mathematical Society Colloquium Publications.* American Mathematical Society, Providence, Rhode Island, 3rd edition, 1967.

[25] N.G. de Bruijn. A survey of the project AUTOMATH. In J.P. Seldin and J.R. Hindley, editors, *To H.B. Curry: Essays in Combinatory Logic, Lambda Calculus, and Formalism*, pages 589–606. Academic Press, 1980.

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

[27] B.A. Carré. *Graphs and Networks.* Oxford University Press, 1979.

[28] P. Chisholm. Calculation by computer. In *Third International Workshop Software Engineering and its Applications*, pages 713–728, Toulouse, France, December 3-7 1990. EC2.

[29] R.L. Constable, et al. *Implementing Mathematics in the Nuprl Proof Development System.* Prentice-Hall, 1986.

[30] J.H. Conway. *Regular algebra and finite machines.* Chapman and Hall, London, 1971.

[31] B. A. Davey and H. A. Priestly. *Introduction to Lattices and Order.* Cambridge Mathematical Textbooks. Cambridge University Press, first edition, 1990.

[32] Augustus De Morgan. *On the Syllogism and Other Logical Writings.* Yale University Press, New Haven, 1966. Edited, with an Introduction, by Peter Heath.

[33] J. Desharnais. *Abstract Relational Semantics.* PhD thesis, School of Computer Science, McGill University, July 1989.

[34] E.W. Dijkstra. *A Discipline of Programming.* Prentice-Hall, 1976.

[35] 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.

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

[37] R.P. Dilworth. Non-commutative residuated lattices. *Transactions of the American Mathematical Society*, 46:426–444, 1939.

[38] P. Dybjer. An inversion principle for Martin-Löf's type theory. Dept. Comp. Sci., Univ. Göteborg, 1989.

[39] C.A.R. Hoare *et al.* Laws of programming. *Communications of the ACM*, 30(8):672–686, 1987. Corrigenda in *30*, 9, p. 770.

[40] C.J. Everett. Closure operators and galois theory in lattices. *Trans. Amer. Math. Soc.*, 55:514–525, 1944.

[41] Maureen H. Fenrick. *Introduction to the Galois Correspondence.* Birkhaüser Boston, 1991.

[42] G. Gierz, K. H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D. S. Scott. *A Compendium of Continuous Lattices.* Springer-Verlag, 1980.

[43] J.A. Goguen, J.W. Thatcher, and E.G. Wagner. An initial algebra approach to the specification, correctness and implementation of abstract data types. In R.T. Yeh, editor, *Current Trends in Programming Methodology, Volume 4: Data Structuring*, pages 80–149. Prentice-Hall, 1978.

[44] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics : a Foundation for Computer Science*. Addison-Wesley Publishing Company, 1989.

[45] T. Hagino. A typed lambda calculus with categorical type constructors. In D.H. Pitt, A. Poigne, and D.E. Rydeheard, editors, *Category Theory and Computer Science*, pages 140–57. Springer-Verlag Lecture Notes in Computer Science 283, 1988.

[46] Paul R. Halmos. *Lectures on Boolean Algebras*. Number 1 in van Nostrand Mathematical Studies. D. van Nostrand Company, Inc., 1963.

[47] J. Hartmanis and R.E. Stearns. Pair algebras and their application to automata theory. *Information and Control*, 7(4):485–507, 1964.

[48] J. Hartmanis and R.E. Stearns. *Algebraic Structure Theory of Sequential Machines*. Prentice-Hall, 1966.

[49] Horst Herrlich and Miroslav Hušek. Galois connections. In Austin Melton, editor, *Mathematical Foundations of Programming Semantics*, LNCS 239, pages 122–134. Springer-Verlag, 1985.

[50] C.A.R. Hoare. Notes on data structuring. In O.-J. Dahl, E.W. Dijkstra, and C.A.R. Hoare, editors, *Structured Programming*. Academic Press, 1972.

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

[52] C.A.R. Hoare and Jifeng He. The weakest prespecification. *Fundamenta Informaticae*, 9:51–84, 217–252, 1986.

[53] P. Hoogendijk. The Boom hierarchy. Proceedings of the EURICS workshop on Calculational Theories of Program Structure, Ameland, The Netherlands, September 1991.

[54] G. Jones. Constructing the fast Fourier transform. Lecture notes, International Summer School on Constructive Algorithmics, vol. 3, 1989.

[55] S.C. Kleene. Representation of events in nerve nets and finite automata. In Shannon and McCarthy, editors, *Automata Studies*, pages 3–41. Princeton Univ. Press, 1956.

[56] D.E. Knuth, J.H. Morris, and V.R. Pratt. Fast pattern matching in strings. *SIAM Journal of Computing*, 6:325–350, 1977.

[57] J. Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65:154–170, 1958.

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

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

[60] J.-L. Lassez, V.L. Nguyen, and E.A. Sonenburg. Fixed point theorems and semantics: a folk tale. *Information Processing Letters*, 14(3):112–116, 1982.

[61] Roger D. Maddux. The origin of relation algebras in the development and axiomatization of the calculus of relations. Department of Mathematics, Iowa State University, Ames, IOWA 50011, USA, May 1990.

[62] G. Malcolm. Homomorphisms and promotability. In J.L.A. van de Snepscheut, editor, *Conference on the Mathematics of Program Construction*, pages 335–347. Springer-Verlag LNCS 375, 1989.

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

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

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

[66] P. Martin Löf. Constructive mathematics and computer programming. In L.J. Cohen, J. Los, H. Pfeiffer, and K.-P. Podewski, editors, *Logic, Methodology and Philosophy of Science, IV*, pages 153–175. North-Holland, 1982.

[67] J. M. McDill, A. C. Melton, and G. E. Strecker. A category of Galois connections. In David H. Pitt, Axel Poigné, and David E. Rydeheard, editors, *Category Theory and Computer Science*, LNCS 283, pages 290–300. Springer-Verlag, 1987.

[68] L. Meertens. Algorithmics – towards programming as a mathematical activity. In *Proceedings of the CWI Symposium on Mathematics and Computer Science*, pages 289–334. North-Holland, 1986.

[69] 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.

[70] L. Meertens. Paramorphisms. To appear in Formal Aspects of Computing, 1991.

[71] A. C. Melton, D. A. Schmidt, and G. E. Strecker. Galois connections and computer science applications. In David H. Pitt, Axel Poigné, and David E. Rydeheard, editors, *Category Theory and Computer Science*, LNCS 283, pages 299–312. Springer-Verlag, 1987.

[72] Austin Melton, Bernd S.W. Schröder, and George E. Strecker. Connections. In S.Brookes, M.Main, A.Melton, M.Mislove, and D.Schmidt, editors, *Mathematical Foundations of Programming Semantics, 7th International Conference, Pittsburgh, March 1991*, volume LNCS598, pages 25–28. Springer-Verlag, 1992.

[73] J. Meseguer and J.A. Goguen. Initiality, induction and computability. In M. Nivat and J.C. Reynolds, editors, *Algebraic Methods in Semantics*, pages 459–542. Cambridge University Press, 1985.

[74] O. de Moor. Indeterminacy in optimization problems. Lecture Notes, International Summer School on Constructive Algorithmics, vol. 2, 1989.

[75] O. de Moor. Inverses in program synthesis. Lecture Notes, International Summer School on Constructive Algorithmics, vol. 2, 1989.

[76] O. de Moor. *Categories, Relations and Dynamic Programming.* PhD thesis, Oxford University Laboratory, Programming Research Group, April 1992.

[77] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory: An Introduction.* Oxford University Press, 1990.

[78] Oystein Ore. Galois connexions. *Transactions of the American Mathematical Society*, 55:493–513, 1944.

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

[80] F. Rietman. An aggregated segment sum theorem in the relational system. Afstudeer Verslag, University of Groningen, 1991.

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

[82] David E. Rydeheard and Rod M. Burstall. *Computational Category Theory.* Prentice Hall International Series in Computer Science. Prentice Hall, first edition, 1988.

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

[84] G. Schmidt and T. Ströhlein. *Relationen und Grafen.* Springer-Verlag, 1988.

[85] J. Schmidt. Beitrage für filtertheorie. II. *Math. Nachr.*, 10:197–232, 1953.

[86] M. Sheeran. Describing hardware algorithms in Ruby. In David et al., editor, *Proc. IFIP TC10/WG10.1 Workshop on Concepts and Characteristics of Declarative Systems.* North-Holland, 1989.

[87] Martin Simons. Galois connections and pair algebras. Unpublished draft, December 1991.

[88] A. Tarski. On the calculus of relations. *Journal of Symbolic Logic*, 6(3):73–89, 1941.

[89] A. Tarski. A fixed point theorem and its applications. *Pacific J. Math.*, pages 285–309, 1955.

[90] Herbert Westren Turnbull. The great mathematicians. In James R. Newman, editor, *The World of Mathematics*, volume 1, pages 73–160. Tempus Books of Microsoft Press, 1988.

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

[92] P. Wadler. Comprehending monads. In *ACM Conference on Lisp and Functional Programming*, June 1990.