Galois Connections and Fixed Point Calculus

Roland Backhouse[*]

October 15, 2001

Abstract

Fixed point calculus is about the solution of recursive equations defined by a monotonic endofunction on a partially ordered set. This tutorial presents the basic theory of fixed point calculus together with a number of applications of direct relevance to the construction of computer programs. The tutorial also presents the theory and application of Galois connections between partially ordered sets. In particular, the intimate relation between Galois connections and fixed point equations is amply demonstrated.

[*]School of Computer Science and Information Technology, University of Nottingham, Nottingham NG8 1BB, England

Contents

1 Introduction

1.1 Fixed Point Equations

Formulating and solving equations is a fundamental mathematical activity and whole areas of mathematics are devoted to the methods for solving particular classes of equations — think, for example, of the differential calculus developed in order to solve differential equations. This chapter is about a class of equations called *fixed point* equations that is particularly important in computing.

Fixed point equations have a very simple form, namely

$$x = f.x \quad ,$$

where f is a given function and x is the unknown. A solution to the equation is called a "fixed point" of the function f because applying f to a solution leaves the value "fixed", i.e. unchanged.

1.2 Languages

In the literature on computing science, fixed point equations are most often called *recursive* equations because the unknown x "recurs" on the right side of the equation. Recursion was first used extensively in computing science in the now-classic Algol 60 report [Nau63] which defined the programming language Algol 60. The Algol 60 report introduced so-called "Backus-Naur Form" to define the syntax of the language. Here is a small, simplified extract from the language definition.

$$\begin{array}{lll} \langle\mathit{Expression}\rangle & ::= & \langle\mathit{Expression}\rangle + \langle\mathit{Expression}\rangle \quad | \quad (\ \langle\mathit{Expression}\rangle\ ) \\ & & |\quad \langle\mathit{Variable}\rangle \end{array}$$

This defines (arithmetic) expressions by recursion. The symbol "$::=$" can be read simply as "is" or, indeed, as an equality symbol. The symbol "$|$" is read as "or". Terms enclosed within angle brackets (thus $\langle\mathit{Expression}\rangle$ and $\langle\mathit{Variable}\rangle$) are called "non-terminals" —these are the entities being defined— and the remaining symbols (here "+" and the two parentheses "(" and ")") are called the "terminal" symbols —these are the symbols that form words in the language being defined— . In this way the definition of $\langle\mathit{Expression}\rangle$ is read as:

> An expression is either an expression followed by the symbol "+" followed by an expression, or a parenthesised expression, or a variable.

Assuming that x and y are variables, the following are examples of expressions:

$$x{+}y \quad , \quad (x) \quad , \quad x \quad , \quad x{+}x{+}y \quad , \quad y+(x{+}y) \quad .$$

The definition of $\langle \mathrm{Expression} \rangle$ is an example of a fixed point equation in which the unknown x is $\langle \mathrm{Expression} \rangle$ and the function f is a function from languages (sets of words) to languages. Language theory provides many good examples of fixed point calculus; we use it frequently as a source of examples.

1.3 Functions

Recursion is used extensively in mathematics for the definition of functions. An elementary example is the factorial function, which is specified by the equation in fac,

$$(1) \qquad \begin{array}{lcl} fac.0 & = & 1 \\ fac.n & = & n * fac.(n-1), \text{ for } n>0. \end{array}$$

The definition of the factorial function also has the form of a fixed point equation although this is not so easy to see as in the previous example. To verify that this is indeed the case we need to rewrite the definition in the form

$$fac \;=\; \cdots \quad .$$

Using the notation $\langle x{:}\, x{\in}\mathrm{Type}{:}\, \mathrm{Exp} \rangle$ for a function that maps a value x of type Type to the value given by expression Exp, we have:

$$fac \;=\; \langle n{:}\, n{\in}\mathbb{N}{:}\, \textbf{if } n=0 \textbf{ then } 1 \textbf{ else } n{*}fac.(n-1) \rangle \quad .$$

Now, abstracting from fac on the right side of this equation, define the function $\mathcal{F}$ by

$$\mathcal{F} \;=\; \langle f{:}\, f \in \mathbb{N}{\leftarrow}\mathbb{N}{:}\, \langle n{:}\, n{\in}\mathbb{N}{:}\, \textbf{if } n=0 \textbf{ then } 1 \textbf{ else } n{*}f.(n-1) \rangle \rangle \quad .$$

This defines $\mathcal{F}$ to be an endofunction[1] on functions of type $\mathbb{N}{\leftarrow}\mathbb{N}$, the type of the factorial function. That is, the function $\mathcal{F}$ maps a function to natural numbers from natural numbers to a function to natural numbers from natural numbers. For example, applying $\mathcal{F}$ to the successor function $\langle n{::}\, n+1 \rangle$ we get

$$\mathcal{F}.\langle n{::}\, n+1 \rangle \;=\; \langle n{::}\; \textbf{if } n=0 \textbf{ then } 1 \textbf{ else } n{*}((n-1)+1) \rangle \quad .$$

(For brevity we have omitted the type information on n.) Simplifying,

$$\mathcal{F}.\langle n{::}\, n+1 \rangle \;=\; \langle n{::}\; \textbf{if } n=0 \textbf{ then } 1 \textbf{ else } n^2 \rangle \quad .$$

The characteristic feature of the factorial function is that its definition demands that

$$fac = \mathcal{F}.fac \quad .$$

That is, fac is a *fixed point* of $\mathcal{F}$.

[1]An *endofunction* is a function whose target and source are the same.

1.4 Datatypes

Recursion is also used in the definition of datatypes in programming languages. A definition of, for example, lists in a functional programming language looks like the following:

$$\mathsf{List}\ \alpha \quad = \quad \mathsf{Nil} \quad | \quad \mathsf{Cons}\ \alpha\ (\mathsf{List}\ \alpha)$$

This states that a list of α's (where α stands for an arbitrary type) is either Nil or the operator Cons applied to a value of type α and a list of α's.

The definition of List is not strictly a fixed-point equation. Rather than expressing the equality of $\mathsf{List}\ \alpha$ and $\mathsf{Nil}\ \ |\ \ \mathsf{Cons}\ \alpha\ (\mathsf{List}\ \alpha)$ (misleadingly suggested by the equals sign) the declaration expresses an *isomorphism* between the type $\mathsf{List}\ \alpha$ and the type $\mathbb{1}{+}\alpha{\times}(\mathsf{List}\ \alpha)$, the disjoint sum ("$+$") of the unit type (a set containing exactly one element, here denoted by "$\mathbb{1}$") and the cartesian product ("$\times$") of the type α and the type $\mathsf{List}\ \alpha$.

1.5 Galois Connections

Fixed point calculus is only of value to computing science if many of the things we want to compute are specified by fixed point equations. Fortunately, this is very much the case. Indeed, many specifications are directly expressed in terms of the solution of a fixed point equation. Moreover, a large number of additional problems have specifications that involve a fixed point equation, albeit less directly. In such cases it is often the case that the specification can be transformed to one that expresses the solution to the specification directly in terms of a fixed point equation. The key to such transformations is the notion of a *Galois connection*.

Galois connections are of interest in their own right, even when not related to the theory of fixed point equations. Their typical use is to define one, relatively complex, function in terms of another, relatively simple, function. A number of the examples we present are of very elementary and well-known functions; even so, readers not familiar with Galois connections may find the calculations we present delightfully refreshing!

Galois connections are also basic to the fundamental notions of infimum and supremum. Section 4.1 discusses these notions; in particular, the notion of suprema and infima of a certain "shape" is discussed in detail.

1.6 Basic Assumptions

The simple form of a fixed-point equation means that it is a very general notion which captures a very broad range of computational problems. But a theory that is too general is usually also too weak to be of practical value. In developing a useful fixed-point calculus we need to impose some non-trivial mathematical properties on the fixed point equations we consider. Here we require two properties. The first property we demand is that the domain of solutions be a *complete lattice*. That is the domain of solutions is a partially ordered set such that suprema and infima of any shape always exist. This requirement clearly holds in the case of languages. When a language is being defined it is required that the "alphabet" of the language is clearly stated. The *alphabet* of a language is just a finite set of symbols, and *words* in the language are finite sequences of symbols in the alphabet. If Σ denotes an alphabet then Σ^* is the notation used to denote the set of all words in that alphabet (that is, the set of all finite sequences —including the empty sequence— of symbols in the alpahabet). A language is thus a subset of Σ^* and, as is well known, the set of all subsets of a given set is a complete lattice under the subset ordering.

Although there is no ordering relation explicitly mentioned in the definition of fac (other than the equality relation, which is pretty uninteresting as an ordering relation) the theory we develop still applies by the simple device of viewing functions as special cases of binary relations. A binary relation between two sets A and B is a subset of the cartesian product $A{\times}B$ and the set of all subsets of $A{\times}B$ forms a complete lattice. This will be our solution domain when solving a fixed-point equation that is intended to define a function to set A from set B.

The second requirement we impose is that the function f in the given fixed point equation be *monotonic* in the ordering on the solution domain. This requirement is also clearly met in the case of language definitions. The function $\langle X{::}\ \{b\}{\cup}\{a\}{\cdot}X{\cdot}\{c\}\rangle$ is typical of the sort of functions that are used in language definitions. (A raised infix dot denotes the concatenation operator extended to sets.) It is the composition of three functions, the function $\langle X{::}\ \{b\}{\cup}X\rangle$, which adds the word b to a language, the function $\langle X{::}\ \{a\}{\cdot}X\rangle$, which concatenates a at the beginning of every word in a given language, and $\langle X{::}\ X{\cdot}\{c\}\rangle$, which concatenates c at the end of every word in a given language. These functions are all clearly monotonic, and thus so is their composition.

1.7 Issues and Applications

An arbitrary fixed point equation may have no solutions, it may have exactly one solution, or it may have more than one solution. An important consequence of our two assumptions is that the fixed point equations we consider always have at least one solution. Indeed, the set of solutions is itself a complete lattice. In particular, a monotonic function on a complete lattice has a *least* and a *greatest* fixed point. Moreover, it has a unique fixed point if (and only if) the least and great fixed points coincide.

A major element of these lecture notes is how to manipulate fixed point equations. Finding out how to express the same quantity by different equations is important to understanding. Another important element is providing general, practical, sufficient conditions for a fixed point equation to have a unique solution. These foundational elements of fixed point calculus are discussed in depth in section 7.

An example of the sort of issue tackled in section 7 is the use of recursion to define a number of values simultaneously. The terminology "mutual recursion" is used. In the definition of Algol 60, for example, this is what the definition of $\langle \mathit{Expression} \rangle$ really looks like:

$$\begin{array}{lcl} \langle \mathit{Expression} \rangle & ::= & \langle \mathit{Expression} \rangle + \langle \mathit{Term} \rangle \quad | \quad \langle \mathit{Term} \rangle \\ \langle \mathit{Term} \rangle & ::= & \langle \mathit{Term} \rangle \times \langle \mathit{Factor} \rangle \quad | \quad \langle \mathit{Factor} \rangle \\ \langle \mathit{Factor} \rangle & ::= & (\ \langle \mathit{Expression} \rangle\ ) \quad | \quad \langle \mathit{Variable} \rangle \end{array}$$

This is still a fixed point equation, but it is an equation in the triple of languages

$$(\langle \mathit{Expression} \rangle\ ,\ \langle \mathit{Term} \rangle\ ,\ \langle \mathit{Factor} \rangle) \quad .$$

An important issue is whether a fixed point equation in a vector of values can be solved piecewise and, if so, how. This is the content of theorem 106.

Section 6.1 is introductory; section 6.3 is also introductory but, even so, considers an important application of fixed point calculus, namely Kleene algebra, which is the algebra of choice, composition and iteration, three indispensible ingredients of any programming language.

2 Galois Connections — Introductory Examples

This section begins our discussion of "Galois connections", a concept first introduced by Oystein Ore in 1944 [Ore44].

The importance of Galois connections lies in their ubiquity and their simplicity. Mathematics and, in particular, computing science abounds with instances of Galois connections. Some simple examples are presented in subsection 2.1 whilst subsection 2.2 continues with a more detailed analysis of the floor and ceiling functions. Later sections discuss more abstract properties of Galois connections.

2.1 Simple Examples

A Galois connection involves two preordered sets[2] $(\mathcal{A}\,,\,\leq)$ and $(\mathcal{B}\,,\preceq)$ and two functions, $F\in\mathcal{A}\leftarrow\mathcal{B}$ and $G\in\mathcal{B}\leftarrow\mathcal{A}$. These four components together form a *Galois connection* iff for all $x\in\mathcal{B}$ and $y\in\mathcal{A}$ the following holds

(2) $\quad F.x \leq y \equiv x \preceq G.y \quad .$

[2]The pair $(\mathcal{A}\,,\,\leq)$ is a *preordered set* if $\leq$ is a binary relation on $\mathcal{A}$ that is reflexive (i.e. $x\leq x$ for all x) and transitive (i.e. $x\leq y \wedge y\leq z \Rightarrow x\leq z$ for all x, y and z).

This compact definition of a Galois connection was first introduced in [Sch53]. We refer to F as the *lower adjoint* and to G as the *upper adjoint*. (Many texts use the names "left" and "right" where we use "lower" and "upper". Our terminology is in line with [GHK⁺80] which is a standard reference on the theory presented here.)

Lots of examples of Galois connections can be given. In the first instance, examples can be given by observing that two inverse functions are Galois connected. Suppose $\mathcal{A}$ and $\mathcal{B}$ are two sets and $F \in \mathcal{A} \leftarrow \mathcal{B}$ and $G \in \mathcal{B} \leftarrow \mathcal{A}$ are inverse functions. Then their being inverse can be expressed by the equivalence, for all $x \in \mathcal{B}$ and $y \in \mathcal{A}$,

$$F.x = y \ \equiv\ x = G.y \quad .$$

This is a Galois connection in which we view $\mathcal{A}$ and $\mathcal{B}$ as ordered sets where the ordering relation is identical to the equality relation. (Two elements are ordered if and only if they are equal.)

That inverse functions are Galois connected is a useful observation — *not* because a study of Galois connections will tell us something we didn't already know about inverse functions, but because we can draw inspiration from our existing knowledge of properties of inverse functions to guide us in the study of Galois-connected functions.

Further examples of Galois connections are not hard to find although sometimes they are not immediately evident. One is the connection between conjunction and implication in the predicate calculus:

$$p \wedge q \Rightarrow r \ \equiv\ q \Rightarrow (p \Rightarrow r) \quad .$$

Here p, q and r denote predicates and the connection is between the functions $(p\wedge)$ and $(p\Rightarrow)$. To be more precise, both sets $\mathcal{A}$ and $\mathcal{B}$ in the definition of a Galois connection are taken to be the set of predicates, and the ordering relation is implication $(\Rightarrow)$. The above formula describes a *family* of Galois connections, one for each instance of the variable p.

An interesting example is provided by negation (**not**) in the predicate calculus. We have:

$$\neg p \Rightarrow q \ \equiv\ p \Leftarrow \neg q \quad .$$

The example is interesting because it involves two different orderings on the same set. Specifically, we can order predicates by implication $(\Rightarrow)$ or by the converse ordering follows-from $(\Leftarrow)$. *Predicates ordered by implication and predicates ordered by follows-from are quite different partially ordered sets.* The point is that there are four elements to the definition of a Galois connection: two ordered sets and two functions between the ordered sets. All four elements form an integral part of the definition and mistakes in the exploitation of the properties of Galois connections may occur if one is not clear about all four.

One elementary example of a Galois connection that is not immediately evident is afforded by the binary maximum operator, max, on real numbers. Denoting it by the infix operator $\uparrow$, we have:

$$x{\uparrow}y \leq z \quad\equiv\quad x \leq z \wedge y \leq z \quad .$$

At first sight this doesn't look like a Galois connection principally on account of the conjunction on the righthand side of the equation. We can however identify it as such as follows. First note that max is a binary function, i.e. a function from the cartesian product $\mathbb{R}{\times}\mathbb{R}$ (the set of pairs of real numbers) to the set $\mathbb{R}$. Now $\mathbb{R}$ is ordered by the at-most relation ($\leq$), and this relation can be extended pointwise to an ordering relation on $\mathbb{R}{\times}\mathbb{R}$. Specifically, denoting the relation by $\leq^2$, we define

$$(u,v) \leq^2 (x,y) \quad\equiv\quad u \leq x \wedge v \leq y \quad .$$

Finally, we define the *doubling function*, denoted by $\triangle$, by

$$\triangle z = (z,z) \quad .$$

Having done so we can rewrite the definition of max as follows:

$$\mathsf{max}(x,y) \leq z \quad\equiv\quad (x,y) \leq^2 \triangle z \quad .$$

Thus max is a function mapping $\mathbb{R}{\times}\mathbb{R}$, ordered pointwise by the relation $\leq^2$, to $\mathbb{R}$, ordered by the at-most relation ($\leq$), and is defined by a Galois-connection connecting it to the doubling function.

Many predicates are themselves adjoints in a Galois connection, but the fact is rarely recognised. An example is the predicate even on integers (which is true when its argument is divisible by 2). Specifically, the integers are ordered by the "is-divisible-by" relation, which we denote here by "$/$" (so $m \,/\, n$ should be read as "m is divisible by n" or, more precisely, as "there is an integer k such that $n \times k = m$") and, for all integers m and booleans b,

$$\mathsf{even}.m \Leftarrow b \quad\equiv\quad m \,/\, (\textbf{if } b \textbf{ then } 2 \textbf{ else } 1) \quad .$$

This is an instance of the general result that a predicate p on the elements of a poset $(\mathcal{A}, \leq)$ with greatest element $\top\!\!\top$ is a lower adjoint in a Galois connection (between $(\mathcal{A}, \leq)$ and $(\mathsf{Bool}, \Leftarrow)$) if there is some constant a such that $p.x \equiv x \leq a$. Specifically, the Galois connection is, for all $x{\in}\mathcal{A}$ and $b{\in}\mathsf{Bool}$,

$$p.x \Leftarrow b \quad\equiv\quad x \leq \textbf{if } b \textbf{ then } a \textbf{ else } \top\!\!\top \quad .$$

It is quite surprising how many predicates fulfill this condition. The "is-empty" test on a set is an example (a set S is empty if $S \subseteq \phi$). Another example is the membership test on a set. Fix some value a in a universe of values and consider the predicate p that given a subset S of the universe returns the value of $a{\in}S$. Then, since $a{\in}S \equiv \{a\}{\subseteq}S$, the general result says that p is a lower adjoint in a Galois connection. Indeed,

$$a{\in}S \Leftarrow b \quad\equiv\quad S \supseteq \textbf{if } b \textbf{ then } \{a\} \textbf{ else } \phi \quad .$$

2.2 The Floor Function

This section discusses a simple but illuminating example of a Galois connection. The *floor* function from reals to integers is described in words as follows: for all real x we take $\lfloor x \rfloor$ (read "floor x") to be the greatest integer that is at most x. Formally, this is captured by a simple equivalence.

Definition 3 (Floor Function) For all real x, $\lfloor x \rfloor$ is an integer such that, for all integers n,

$$n{\leq}\lfloor x \rfloor \equiv n{\leq}x \quad .$$

□

In the definition of the floor function we use the mathematical convention of *not* denoting the conversion from integers to reals. It is implicit in the inequality $n{\leq}x$ which seems to compare an integer with a real. In fact, what is meant is the comparison of the real value corresponding to n with the real value x. On the right side of the equivalence the at-most relation ("$\leq$") is between reals whereas on the left side it is between integers.

Making explicit both conversions, temporarily adopting a Java-like casting notation, reveals the two adjoint functions in a Galois connection. We have, for all real x, $(\texttt{floor})x$ is an integer such that for all integers n,

$$\texttt{n} \leq (\texttt{floor})\texttt{x} \quad\equiv\quad (\texttt{real})\texttt{n} \leq x \quad .$$

So the floor of x is defined by connecting it to the conversion from integers to reals in a simple equivalence.

The floor function is an instance of a common phenomenon, namely that many functions are defined to be adjoint to an embedding of one partially ordered set in another. The functions so defined are *closure operators*[3]. Examples include the reflexive closure, symmetric closure and transitive closure of a relation. These are adjoints of the embeddings of, respectively, all reflexive relations, all symmetric relations and all transitive relations in the ordered set of relations (of an appropriate type).

[3]Formally, it is not the adjoint that is a closure operator but the composition of the adjoint with the embedding function. This said, it is useful to adopt the mathematical convention of omitting explicit mention of the embedding function and this is what we do from now on.

2.2.1 Properties of Floor

The first time that one encounters a definition like definition 3 it can be difficult to see how it is used. This section illustrates the basic techniques for exploiting Galois connections.

The first thing we can do is to try to identify some special cases that simplify the definition. Two possibilities present themselves immediately; both exploit the fact that the at-most relation is reflexive. The equation

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

is true for all integers n and reals x. Also, $\lfloor x \rfloor$ is by definition an integer. So we can instantiate n to $\lfloor x \rfloor$. We get

$$\lfloor x \rfloor \leq \lfloor x \rfloor \equiv \lfloor x \rfloor \leq x \quad .$$

The lefthand side that is obtained — $\lfloor x \rfloor \leq \lfloor x \rfloor$ — is true, and so the righthand side is also true. That is,

$$\lfloor x \rfloor \leq x \quad .$$

This tells us that the floor function rounds down. It returns an integer that is at most the given real value.

The second possibility is to instantiate x to n. This is allowed because every integer is a real. Strictly, however, we are instantiating x to the real value obtained by converting n. We get

$$n \leq \lfloor n \rfloor \equiv n \leq n \quad .$$

In this case it is the right side of the equivalence that is true. So we can simplify to

$$n \leq \lfloor n \rfloor \quad .$$

Earlier we determined that $\lfloor x \rfloor \leq x$ for all real values x. Instantiating x to n, we get

$$\lfloor n \rfloor \leq n \quad .$$

So, as the at-most relation is anti-symmetric, we have derived that for all integers n

$$\lfloor n \rfloor = n \quad .$$

A good understanding of the equivalence operator suggests something else we can do with the defining equation: use the rule of *contraposition*. The contrapositive of the definition of the floor function is, for all integers n and real x,

$$\neg(n \leq \lfloor x \rfloor) \equiv \neg(n \leq x) \quad .$$

But $\neg(n \le m) \equiv m < n$. So

$$\lfloor x \rfloor < n \equiv x < n \quad .$$

Equally, using that for integers m and n , $m < n \equiv m+1 \le n$,

$$\lfloor x \rfloor + 1 \le n \equiv x < n \quad .$$

Now we can exploit reflexivity of the at-most relation again. Instantiating n with $\lfloor x \rfloor + 1$ and simplifying we deduce:

$$x < \lfloor x \rfloor + 1 \quad .$$

Recalling that $\lfloor x \rfloor \le x$, we have established

$$\lfloor x \rfloor \le x < \lfloor x \rfloor + 1 \quad .$$

In words, $\lfloor x \rfloor$ is the (unique) integer such that $\lfloor x \rfloor$ is at most x and x is less than $\lfloor x \rfloor + 1$.

We now ask whether the floor function is monotonic. That is, we want to show that

$$\lfloor x \rfloor \le \lfloor y \rfloor \Leftarrow x \le y \quad .$$

Here we calculate:

$$\begin{array}{ll}
 & \lfloor x \rfloor \le \lfloor y \rfloor \\
= & \qquad \{ \qquad \text{definition 3, } x,n := \lfloor y \rfloor , \lfloor x \rfloor \quad \} \\
 & \lfloor x \rfloor \le y \\
\Leftarrow & \qquad \{ \qquad \text{transitivity of } \le \quad \} \\
 & \lfloor x \rfloor \le x \le y \\
= & \qquad \{ \qquad \lfloor x \rfloor \le x \quad \} \\
 & x \le y \quad .
\end{array}$$

Thus the floor function is indeed monotonic.

2.2.2 Rounding Off

Let us now demonstrate how to derive more complicated properties of the floor function. In the process we introduce an important technique for reasoning with Galois connections called the rule of *indirect equality*.

The definition of the language Java prescribes that integer division rounds *towards* zero. This causes difficulties in circumstances when it is required to round *away* from zero. Had the definition been that integer division rounds *down* then it is easy to implement rounding *up*, *towards* zero or *away* from zero. Rounding up, for example, corresponds to negating, then rounding down, and then negating again. (See exercise 7(a).)

The problem is thus how to implement rounding up integer divisions supposing that our programming language always rounds down.

In order to express the problem we need the *ceiling* function. The definition is a dual of the definition of the floor function.

Definition 4 For all real x, $\lceil x \rceil$ is an integer such that, for all integers n,

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

□

We leave it as an exercise to the reader to derive properties of the ceiling function dual to the properties of the floor function derived in section 2.2.1.

Rounding down an integer division of positive numbers m and n is expressed by

$$\left\lfloor \frac{m}{n} \right\rfloor$$

where $\frac{m}{n}$ is the real division of m and n. Dually, rounding up is expressed by

$$\left\lceil \frac{m}{n} \right\rceil \quad .$$

Implementing rounding up given an implementation of rounding down amounts to finding suitable values p and q so that

$$\left\lfloor \frac{p}{q} \right\rfloor = \left\lceil \frac{m}{n} \right\rceil \quad .$$

The values p and q should be expressed as arithmetic functions of m and n (that is, functions involving addition and multiplication, but not involving the floor or ceiling functions).

The rule of *indirect equality* is

$$m = n \;\equiv\; \forall\langle k{::}\; k \leq m \equiv k \leq n\rangle$$

where k, m and n all range over the same poset. Using this rule, we can *derive* suitable expressions for p and q. Specifically, for arbitrary integer k, we aim to eliminate the ceiling function from the inequality

$$k \leq \left\lceil \frac{m}{n} \right\rceil$$

obtaining an inequality of the form

$$k \leq e$$

where e is an arithmetic expression in m and n. We may then conclude that

$$\lfloor e \rfloor = \left\lceil \frac{m}{n} \right\rceil \ .$$

The first step in the calculation is perhaps the most difficult. This is because the definition of the ceiling function, definition 4, provides a rule for dealing with inequalities where a ceiling value is on the lower side of an at-most relation but not when it is on the higher side (which is the case we are interested in). However, recalling our discussion of the floor function, the solution is to consider the contrapositive of the defining equation. Specifically we have, by negating both sides of 4,

(5) $\quad n < \lceil x \rceil \equiv n < x \ .$

We can now proceed with the derivation:

$$\begin{array}{ll}
& k \leq \left\lceil \frac{m}{n} \right\rceil \\
\equiv & \{ \quad \text{integer arithmetic} \quad \} \\
& k-1 < \left\lceil \frac{m}{n} \right\rceil \\
\equiv & \{ \quad \text{contrapositive of definition of ceiling (rule (5))} \quad \} \\
& k-1 < \frac{m}{n} \\
\equiv & \{ \quad \text{arithmetic, } n > 0 \quad \} \\
& n(k-1) < m \\
\equiv & \{ \quad \text{integer inequalities} \quad \} \\
& n(k-1)+1 \leq m \\
\equiv & \{ \quad \text{arithmetic, } n > 0 \quad \} \\
& k \leq \frac{m+n-1}{n} \\
\equiv & \{ \quad \text{definition of floor function: (3)} \quad \} \\
& k \leq \left\lfloor \frac{m+n-1}{n} \right\rfloor \ .
\end{array}$$

Here k is arbitrary. So, by indirect equality, we get

(6) $\quad \left\lceil \frac{m}{n} \right\rceil = \left\lfloor \frac{m+n-1}{n} \right\rfloor \ .$

In Java, for example, if it is required to round up the result of dividing `m` by `n` one should compute `(m+n-1)/n` .

Exercise 7 *Prove the following properties.* (Hint: use indirect equality.)

(a) $-\lfloor x \rfloor = \lceil -x \rceil$,

(b) $\lfloor x+m \rfloor = \lfloor x \rfloor + m$,

(c) $\lfloor x/m \rfloor = \lfloor \lfloor x \rfloor / m \rfloor$ (assuming m is a positive integer),

(d) $\left\lfloor \sqrt{\lfloor x \rfloor} \right\rfloor = \left\lfloor \sqrt{x} \right\rfloor$.

□

3 Abstract Properties of Galois Connections

We now begin a study of the abstract properties of Galois connections. First, we record a number of basic properties following which we give an alternative, equivalent definition of a Galois connection (in fact, the original definition). An exercise presents a third definition. Knowing that a concept can be defined in several different but equivalent ways is an indicator of its importance as well as helping one to recognise it in other applications. This subsection is the one to read if you want more insight into what it means for one function to be the adjoint of another. Subsequently we look back at the examples in the section preceeding this one in order to see how the abstract properties might have lead us to anticipate those examples. The basic idea is that we seek a generalisation of the rule that inverse functions have inverse algebraic properties.

There are many more properties of Galois connections not discussed here. Some of these have been included in the exercises. Proofs of the lemmas and theorems are omitted when they present no real difficulty.

3.1 Combining Galois Connections

In what follows we take $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ to be *partially* ordered sets [4] and we assume that $F \in \mathcal{A} \leftarrow \mathcal{B}$ and $G \in \mathcal{B} \leftarrow \mathcal{A}$. For such an F and G we recall the following definition.

Definition 8 (Galois Connection) (F, G) is a Galois connection between the posets $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ means that, for all $x \in \mathcal{B}$ and $y \in \mathcal{A}$,

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

[4] The difference between a partially ordered set and a preordered set is that the ordering on a partially ordered set is anti-symmetric. That is, $x = y \equiv x \sqsubseteq y \wedge y \sqsubseteq x$ for all x and y . The theory of Galois connections can be developed in its entirety for preordered sets but it's occasionally simpler to assume a partial ordering. Where we do so will be apparent from the fact that we appeal to anti-symmetry.

□

In order to make the formulae more readable, we will often drop the subscripts from the orderings. This can be confusing, even though it can usually be deduced which ordering is meant from type considerations. Thus, occasionally we will reintroduce the subscripts.

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.

The names "lower" and "upper" are somewhat arbitrary because orderings can always be turned upside down to make big small and small big. The mathematical jargon for this is that statements about orderings can always be "dualised". In the case of Galois connections we have:

Theorem 9 (F, G) is a Galois connection between the posets $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ iff (G, F) is a Galois connection between the posets $(\mathcal{B}, \sqsupseteq_{\mathcal{B}})$ and $(\mathcal{A}, \sqsupseteq_{\mathcal{A}})$.

Proof Immediate from

$$(F.x \sqsubseteq_{\mathcal{A}} y \equiv x \sqsubseteq_{\mathcal{B}} G.y) \equiv (G.y \sqsupseteq_{\mathcal{B}} x \equiv y \sqsupseteq_{\mathcal{A}} F.x) \quad .$$

□

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.

A second, very basic, result is that Galois-connected functions can be composed to form new Galois connections.

Theorem 10 If (F, G) is a Galois connection between the posets $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ and (H, K) is a Galois connection between the posets $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ and $(\mathcal{C}, \sqsubseteq_{\mathcal{C}})$ then $(F \bullet H , K \bullet G)$ is a Galois connection between the posets $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{C}, \sqsubseteq_{\mathcal{C}})$. (Here and elsewhere we use the symbol "$\bullet$" for function composition.)

□

A final, very basic, theorem should not be neglected.

Theorem 11 (I, I), where I denotes the identity function on $\mathcal{A}$, is a Galois connection between the poset $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and itself.

□

3.2 Cancellation

Looking at the definition of a Galois connection, two very obvious properties are the *cancellation* laws.

(12) $x \sqsubseteq G.F.x$ for all $x \in \mathcal{B}$,

(13) $F.G.y \sqsubseteq y$ for all $y \in \mathcal{A}$.

We call these "cancellation" laws because they enable one to eliminate occurrences of the adjoints F and G from an expression. They are possibly the most frequently used of all the properties of Galois connections.

Using the cancellation laws one can infer that the adjoints F and G are monotonic. The combination of the cancellation laws and monotonicity is an equivalent way of defining the notion of a Galois connection, in fact the way that Ore [Ore44] originally defined the notion.

Theorem 14 (F, G) is a Galois connection between the posets $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ iff the following two conditions hold.

(a) For all $x \in \mathcal{B}$ and $y \in \mathcal{A}$,

$$x \sqsubseteq G.(F.x) \quad \text{and} \quad F.(G.y) \sqsubseteq y \ .$$

(b) F and G are both monotonic.

□

Definition 8, proposed by J. Schmidt [Sch53], and Ore's definition, contained in theorem 14, 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.

3.3 Pointwise Ordering

Ore's definition is most useful when expressed at function level. Let us define the relation $\dot{\sqsubseteq}$ on functions of the same type by

$$f \mathrel{\dot{\sqsubseteq}} g \ \equiv\ \forall\langle x{::}\ f.x \sqsubseteq g.x\rangle \ .$$

Then we can eliminate the dummies x and y in the cancellation laws to obtain

(15) $\quad I_{\mathcal{B}} \mathrel{\dot{\sqsubseteq}} G{\bullet}F \quad$ and $\quad F{\bullet}G \mathrel{\dot{\sqsubseteq}} I_{\mathcal{A}}$.

Schmidt's definition can also be lifted to function level and, in combination with (15), can be used to construct elegant theorems. Specifically, we have:

Lemma 16 (F,G) is a Galois connection between the posets $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ equivales, for all functions h and k with the same (arbitrary) domain, and ranges respectively $\mathcal{B}$ and $\mathcal{A}$,

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

□

In words,

(17) $\quad (F, G)$ forms a Galois connection $\equiv$ $(F{\bullet}\,, G{\bullet})$ forms a Galois connection.

A property that is somewhat weaker but in a sense dual to this one is the following:

(18) $\quad (F, G)$ forms a Galois connection $\Rightarrow$ $({\bullet}G\,, {\bullet}F)$ forms a Galois connection.

(Take care to note the switch in the order of F and G.) Specifically, we have:

Lemma 19 If (F,G) is a Galois connection between the posets $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ then, for all *monotonic* functions h and k with the same range, and domains respectively $\mathcal{B}$ and $\mathcal{A}$,

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

□

3.4 Poset Isomorphism

A function f with domain $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and range $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ is a *poset monomorphism* if, for all x, y in $\mathcal{A}$, $x \sqsubseteq_{\mathcal{A}} y \equiv f.x \sqsubseteq_{\mathcal{B}} f.y$. A poset *isomorphism* is a surjective poset monomorphism. In this section we prove the following lemma, which is an important component of the "unity-of-opposites" theorem proved in section 5.

Lemma 20 If (F,G) is a Galois connection between the posets $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ then

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

$G \in G.\mathcal{A} \leftarrow F.\mathcal{B}$ is a poset isomorphism.

Moreover, within the given domains, F and G are inverses.

□

In order to prove the lemma, a more convenient, higher-order definition of a poset monomorphism is the following.

Function F with domain $G.\mathcal{A}$ is a *poset monomorphism* if, for all monotonic functions h and k (with range $\mathcal{A}$ and the same domain),

(21) $\quad G \bullet h \mathrel{\dot{\sqsubseteq}} G \bullet k \;\equiv\; F \bullet G \bullet h \mathrel{\dot{\sqsubseteq}} F \bullet G \bullet k$.

To show that F is a poset monomorphism, we begin with an elementary calculation.

$$
\begin{array}{ll}
 & G \bullet F \bullet G \\
\dot{\sqsubseteq} & \quad \{ \quad \text{cancellation of } F \bullet G \quad \} \\
 & G \\
\dot{\sqsubseteq} & \quad \{ \quad \text{cancellation of } G \bullet F \quad \} \\
 & G \bullet F \bullet G \ .
\end{array}
$$

It follows, by anti-symmetry, that $G \bullet F \bullet G = G$. In words, G is the left inverse of $F \in F.\mathcal{B} \leftarrow G.\mathcal{A}$. Dually, $F \bullet G \bullet F = F$. (Thus, F is the left inverse of $G \in G.\mathcal{A} \leftarrow F.\mathcal{B}$.) Now,

$$
\begin{array}{ll}
 & F \bullet G \bullet h \mathrel{\dot{\sqsubseteq}} F \bullet G \bullet k \\
\equiv & \quad \{ \quad (F, G) \text{ is a Galois connection, lemma 16} \quad \} \\
 & G \bullet h \mathrel{\dot{\sqsubseteq}} G \bullet F \bullet G \bullet k \\
\equiv & \quad \{ \quad G \bullet F \bullet G = G \quad \} \\
 & G \bullet h \mathrel{\dot{\sqsubseteq}} G \bullet k \ .
\end{array}
$$

So, $F \in F.\mathcal{B} \leftarrow G.\mathcal{A}$ is a poset monomorphism. It remains to show that $F \in F.\mathcal{B} \leftarrow G.\mathcal{A}$ is surjective. Suppose $y \in \mathcal{B}$. We have to exhibit some $x \in \mathcal{A}$ such that $F.y = F.(G.x)$. The key is the property $F \bullet G \bullet F = F$. By this property, $F.y = F.(G.(F.y))$ so that we may take $F.y$ for x . Dually, $G \in G.\mathcal{A} \leftarrow F.\mathcal{B}$ is a poset isomorphism. This completes the proof.

Exercise 22 Suppose $F \in \mathcal{A} \leftarrow \mathcal{B}$ and $G \in \mathcal{B} \leftarrow \mathcal{A}$ are two functions between posets $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$. *Prove that* F *and* G *are inverse poset monomorphisms is the same as the conjunction of the properties:*

(F, G) is a Galois connection between $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$, and

(G, F) is a Galois connection between $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ and $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$.

□

3.5 Uniqueness of Adjoints

There is a simple but powerful corollary of lemmas 16 and 19. Specifically, suppose $(F_0\,,\,G_0)$ and $(F_1\,,\,G_1)$ are both pairs of Galois connected functions between the same partially ordered sets. Then,

(23) $\quad F_0 \mathrel{\dot{\sqsubseteq}} F_1 \;\equiv\; G_1 \mathrel{\dot{\sqsubseteq}} G_0 \quad ,$

since

$$
\begin{array}{ll}
 & F_0 \mathrel{\dot{\sqsubseteq}} F_1 \\
\equiv & \qquad \{ \qquad \text{lemma 16 with } F,G,h,k := F_0,G_0,I,F_1 \quad \} \\
 & I \mathrel{\dot{\sqsubseteq}} G_0 \bullet F_1 \\
\equiv & \qquad \{ \qquad \text{lemma 19 with } F,G,h,k := F_1,G_1,I,G_0 \quad \} \\
 & G_1 \mathrel{\dot{\sqsubseteq}} G_0 \quad .
\end{array}
$$

(Note this time the switch in the order of subscripts.) Hence, by applying 23 twice and combining the two applications with anti-symmetry,

(24) $\quad F_0 = F_1 \;\equiv\; G_1 = G_0 \quad .$

Thus adjoints are *uniquely* defined. We shall see in the next section how this observation can be put to good use.

3.6 Universal Property

One of the reasons that Galois connections are not easy to spot is that standard mathematical definitions of functions like, for example, the floor function do not take the form of either Schmidt's or Ore's definition of a Galois connection. Rather, they correspond to a hybrid of the two. The hybrid definition is given in the following lemma.

Lemma 25 $\quad (F\,,G)$ is a Galois connection between the posets $(\mathcal{A}\,,\,\sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}\,,\,\sqsubseteq_{\mathcal{B}})$ iff the following conditions hold.

(a) G is monotonic.

(b) For all $x{\in}\mathcal{B}$, $x \sqsubseteq G.(F.x)$.

(c) For all $x{\in}\mathcal{B}$ and $y{\in}\mathcal{A}$, $x \sqsubseteq G.y \Rightarrow F.x \sqsubseteq y$.

$\square$

The hybrid definition of a Galois connection is the least elegant of the three alternatives because the symmetry between the adjoint functions is hidden. An explanation for why it is nevertheless commonly used is that Galois connections are often used to define a function, F, in terms of a known function, G. The hybrid definition focuses on the requirements on F. Indeed, we can put the hybrid definition into words as follows.

> Given a monotonic function G, the lower adjoint, F, of G is defined by the requirement that, for all x, $F.x$ is the least y such that $x \sqsubseteq G.y$.

(Note that requirement (b) in lemma 25 requires that $F.x$ be a y such that $x \sqsubseteq G.y$ and requirement (c) that it be the least such y.) The requirement "for all x, $F.x$ is the least y such that $x \sqsubseteq G.y$" is often referred to as the *universal property* of F.

Examples of this form of definition can be found in Gentzen's [Gen69] formalisation of what he called "natural deduction". Gentzen defined the logical operators (conjunction, implication, etc.) systematically by giving *introduction* and *elimination* rules for each operator. An example is his definition of disjunction. There are two introduction rules, namely:

$$p \Rightarrow p \vee q$$

and

$$q \Rightarrow p \vee q \quad .$$

(Gentzen would have used a turnstile rather than an implication operator. Such subtleties will be ignored in this discussion.) There is one elimination rule for disjunction:

$$(p \Rightarrow r) \wedge (q \Rightarrow r) \;\Rightarrow\; (p \vee q \Rightarrow r) \quad .$$

To see that Gentzen's rules conform to properties (a), (b) and (c) of lemma 25 we first rewrite the elimination rule in the same way as we did for maximum above. Doing so and comparing with requirement (c), we identify G as the doubling function and F as disjunction:

$$((p,q) \Rightarrow^2 (r,r)) \;\Rightarrow\; (\vee.(p,q) \Rightarrow r) \quad .$$

(The relation $\Rightarrow^2$ is defined in just the same way as we defined $\leq^2$ earlier. That is, $(p,q) \Rightarrow^2 (r,s)$ equivales $(p \Rightarrow r) \wedge (q \Rightarrow s)$.) We now check that the required cancellation law ($x \sqsubseteq G.(F.x)$) corresponds to the introduction rules. Formally, it is:

$$(p,q) \Rightarrow^2 (p \vee q \, , \, p \vee q)$$

which is indeed the same as the conjunction of $p \Rightarrow p \vee q$ and $q \Rightarrow p \vee q$. Finally, it is obvious that the doubling function is monotonic.

3.7 Commutativity Properties

In our discussion of elementary examples of Galois-connected functions we observed that inverse functions are Galois connected. A vital property of inverse functions is that they have "inverse" algebraic properties. The exponential function, for instance, has as its inverse the logarithmic function; moreover,

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

whereas

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

In general, if θ and ϕ are inverse functions then, for any functions f and g of appropriate type,

$$\forall\langle x{::}\; \theta.(f.x) = g.(\theta.x)\rangle \;\equiv\; \forall\langle y{::}\; f.(\phi.y) = \phi.(g.y)\rangle \quad .$$

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,

(26) $\quad \theta_0{\bullet}f = g{\bullet}\theta_1 \;\equiv\; f{\bullet}\phi_1 = \phi_0{\bullet}g \quad .$

In general, our goal is to try to predict algebraic properties of functions previously unknown to us. Many such properties have the form of "commutativity properties". Among such properties are, for example, the property $-(2{\cdot}x) = 2{\cdot}(-x)$ which expresses the fact that multiplication by 2 commutes with negation. In addition we include properties like $\ln\frac{1}{x} = -(\ln x)$ in which the order of application of the logarithmic function is "commuted" but in so doing a change occurs in the function with which it is commuted (in this case the reciprocal function becomes negation). In this way distributivity properties also become commutativity properties. For instance the property that $x{\cdot}(y{+}z) = (x{\cdot}y){+}(x{\cdot}z)$ is a commutativity property of addition. Specifically, multiplication by x after addition commutes to addition after the function $(y,z) \mapsto (x{\cdot}y\,,\,x{\cdot}z)$.

In general, for a given function F we are interested in discovering functions g and h for which $F{\bullet}g = h{\bullet}F$. In the case that F is defined by a Galois connection we can often answer this question most effectively by translating it into a question about the commutativity properties of its adjoint — especially in the case that the adjoint is a known function with known properties, or a "trivial" function whose algebraic properties are easily determined. The latter is the case with the floor function: the adjoint function is the function embedding integers into reals.

The rule that is the key to this strategy is the following. Suppose, for numbers m and n, $0 \le m$ and $0 \le n$, and for all i, $0 \le i < m{+}n$, (F_i, G_i) is a Galois-connected pair of functions. Then, assuming the functions are so typed that the compositions and equalities are meaningful,

(27) $F_0 \bullet \ldots \bullet F_{m-1} = F_m \bullet \ldots \bullet F_{m+n-1} \;\equiv\; G_{m+n-1} \bullet \ldots \bullet G_m = G_{m-1} \bullet \ldots \bullet G_0$.

(Note that the cases $m = 0$ and $n = 0$ are included; the composition of zero functions is of course the identity function.) In particular, if for $i = 0..1$, (h_i, k_i) is a Galois-connected pair of functions and so too is (F, G) then

(28) $h_0 \bullet F = F \bullet h_1 \;\equiv\; k_1 \bullet G = G \bullet k_0$.

The rule (27) captures the strategy used to discover algebraic properties of an unknown function F , namely to translate to the discovery of properties of the adjoint function.

The proof of (27) is trivial: according to theorems 10 and 11, the m-fold composition $G_{m-1} \bullet \ldots \bullet G_0$ is an upper adjoint of $F_0 \bullet \ldots \bullet F_{m-1}$ and the n-fold composition $G_{m+n-1} \bullet \ldots \bullet G_m$ is an upper adjoint of $F_m \bullet \ldots \bullet F_{m+n-1}$. Thus, (27) is the instance of (24) obtained by making the instantiations:

$$\begin{aligned} F_0 &:= F_0 \bullet \ldots \bullet F_{m-1} \;, \\ F_1 &:= F_m \bullet \ldots \bullet F_{m+n-1} \;, \\ G_1 &:= G_{m+n-1} \bullet \ldots \bullet G_m \;, \\ G_0 &:= G_{m-1} \bullet \ldots \bullet G_0 \;. \end{aligned}$$

Note that (27) subsumes the composition and identity rules (theorems 10 and 11) and the uniqueness of adjoints (24), the three rules that contributed to its proof. Nevertheless we do not recommend that you should commit it to memory. It's better to remember the individual rules and the general idea of how these may be combined.

Exercise 29 A property that combines (16) and (19) is the following. 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. Let $h \in \mathcal{B}_0 \leftarrow \mathcal{B}_1$ and $k \in \mathcal{A}_0 \leftarrow \mathcal{A}_1$ be arbitrary monotonic functions. Then

(30) $F_0 \bullet h \mathrel{\dot{\sqsubseteq}} k \bullet F_1 \;\equiv\; h \bullet G_1 \mathrel{\dot{\sqsubseteq}} G_0 \bullet k$.

Prove this rule.

□

Exercise 31 A pattern that you may have observed in the properties of the floor function is that if f is an arbitrary function from reals to reals such that

it is the upper adjoint in a Galois connection, and its (lower) adjoint maps integers to integers.

then f commutes with the floor function in the sense that, for all real x,

$$\lfloor f.x \rfloor = \lfloor f.\lfloor x \rfloor \rfloor \quad .$$

Show how this is an instance of (27). In other words, state precisely how to instantiate m, n and the functions F_0, ..., F_{m+n-1} and G_0, ..., G_{m+n-1} in order to obtain the law.

□

4 Existence of Galois Connections

In this section we delve deeper into the theory of Galois connections. The goal is to establish a number of theorems that predict the existence of adjoint functions. These theorems are all intimately tied up with extremum preservation properties of functions, which in turn depend on the notions of supremum and infimum. There are four sections. Section 4.1 defines suprema and infima of a given “shape”, section 4.2 then defines preservation properties and section 4.3 presents the existence theorems.

In order to better motivate what is to follow, let us begin with a brief outline. We recall that a Galois connection is a connection between two functions F and G between two posets $(\mathcal{A}, \sqsubseteq)$ and $(\mathcal{B}, \preceq)$ of the form

$$F.x \sqsubseteq y \equiv x \preceq G.y \quad .$$

Typical accounts of the existence of Galois connections focus on the properties of these *functions*. For example, given a function F, one may ask whether F is a lower adjoint in a Galois connection. The question we want to ask is subtley different.

Note that the statement $F.x \sqsubseteq y$ defines a *relation* between $\mathcal{B}$ and $\mathcal{A}$. So too does $x \preceq G.y$. The existence of a Galois connection states that these two relations are equal. A natural question is therefore: under which conditions does an arbitrary (binary) relation between two posets define a Galois connection between the sets?

Exploring the question in more detail leads to the following question. Suppose R is a relation between posets $\mathcal{B}$ and $\mathcal{A}$ (i.e. $R \subseteq \mathcal{B} \times \mathcal{A}$). What is a necessary and sufficient condition that there exist a function F such that

$$(x,y) \in R \;\equiv\; F.x \sqsubseteq y \qquad ?$$

Such a relation is called a *pair algebra*. If we know the answer to this question then we can answer the question of whether a given function G has a lower adjoint — by defining R to be the relation $x \preceq G.y$ on x and y. We can also answer the dual question of whether there exists a function G such

$$(x,y) \in R \;\equiv\; x \preceq G.y \quad .$$

If, for a given relation R, the answer to both these questions (the original and its dual) is positive, then the relation R clearly defines a Galois connection.

In order to simplify the question, we make an abstraction step. We are required to find —*for each* x— a value $F.x$ such that $(x,y) \in R \equiv F.x \sqsubseteq y$. Suppose we fix x and hide the dependence on x. Then the problem becomes one of determining, for a given predicate p, necessary and sufficient conditions guaranteeing that

$$(32) \qquad p.y \;\equiv\; a \sqsubseteq y$$

for some a. If we can solve this simplified problem, we have also solved the original problem by defining $p.y$ to be $(x,y) \in R$ and $F.x$ to be a.

It is easy to identify a necessary condition for (32) to hold: a must be the *least* y satisfying p. That is, $p.a$ must hold (since $a \sqsubseteq a$) and for all y such that $p.y$ holds, $a \sqsubseteq y$. The question thus becomes: when is there a least element satisfying a given predicate p, and when does this least element a (say) satisfy (32) for all y?

The least element satisfying a given property (if it exists) is characterised by two properties. First, it itself satisfies the property and, second, it is the "infimum" of all values satisfying the property. Section 4.1 defines the notion of an infimum via a Galois connection. The dual notion of "supremum" is also discussed in this section. Infima and suprema are collectively called "extrema".

Among the properties of extrema that we study, the question studied in section 4.2 of whether a function preserves infima is particularly important. The question is of importance in its own right, but we shall also see that it is vital to answering our question about the existence of Galois connections. Indeed, the basic insight is that a predicate p on a (complete) poset preserves infima if and only if there is an a such that, for all y, $p.y$ is $a \sqsubseteq y$. We see in section 4.3 how this simple observation is used to prove the "fundamental theorem" (theorem 48) on the existence of Galois connections.

Infima and suprema need not exist. Section 5 is devoted to showing how the existence of a Galois connection can be used to predict the existence of extrema in a poset.

4.1 Infima and Suprema

4.1.1 Infima and Completeness

In this section we formulate the notion of an *extremum* (an infimum or a supremum) of a certain "shape" (for example, the infimum of a finite set, or the infimum of a countably infinite set) via a Galois connection. The notions of completeness and cocompleteness, relative to a given shape, are also defined.

Suppose $(\mathcal{A},\sqsubseteq)$ and $(\mathcal{B},\preceq)$ are partially ordered sets and $f\in\mathcal{A}\leftarrow\mathcal{B}$ is a monotonic function. Then an *infimum* of f is a solution of the equation:

$$(33)\qquad x ::\ \ \forall\langle a::\ a\sqsubseteq x \equiv \forall\langle b::\ a\sqsubseteq f.b\rangle\rangle\ \ .$$

As an example, consider the function $\langle b{:}\,c\sqsubseteq b{:}\,b\rangle$ where c is some given constant. This has infimum c since

$$\forall\langle a::\ a\sqsubseteq c\ \equiv\ \forall\langle b{:}\,c\sqsubseteq b{:}\,a\sqsubseteq b\rangle\rangle\ \ .$$

There are two ways in which this definition differs from the most common definition of an infimum. The first is that it is more common to define the infimum of a *set* rather than of a *function*. That the two notions are equivalent is not difficult to see. The infimum of a set is the infimum of the identity function on that set, and the infimum of a function can be thought of as the infimum of the range of the function. (The range of a function is a set.) Defining infima on functions is an elegant way of being able to talk about different kinds of infima as we shall see very shortly. The second difference is that an infimum is often defined as a *greatest lower bound*. That is, x is an infimum of f if it is a lower bound:

$$\forall\langle b::\ x\sqsubseteq f.b\rangle\ \ ,$$

and it is greatest among such lower bounds

$$\forall\langle a::\ a\sqsubseteq x \Leftarrow \forall\langle b::\ a\sqsubseteq f.b\rangle\rangle\ \ .$$

This is entirely equivalent to (33), as is easily verified.

The infimum of a function f need not exist but if it does it is certainly unique. This is easily seen. Specifically, for all $a{\in}A$,

$$\begin{array}{ll}
 & a\sqsubseteq x_0 \\
\equiv & \{\qquad \bullet\quad x_0 \text{ solves (33)}\quad \} \\
 & \forall\langle b::\ a\sqsubseteq f.b\rangle \\
\equiv & \{\qquad \bullet\quad x_1 \text{ solves (33)}\quad \} \\
 & a\sqsubseteq x_1\ \ .
\end{array}$$

Hence, by indirect equality,

$$x_0 \text{ and } x_1 \text{ both solve (33)}\ \Rightarrow\ x_0 = x_1\ \ .$$

As mentioned, equation (33) need not have a solution. If it does, for a given f, we denote its solution by $\sqcap f$. By definition, then,

(34) $\forall\langle a::\ a \sqsubseteq \sqcap f \equiv \forall\langle b::\ a \sqsubseteq f.b\rangle\rangle$.

Suppose we fix the posets $\mathcal{A}$ and $\mathcal{B}$ and consider all functions of type $\mathcal{A}\leftarrow\mathcal{B}$. Suppose that there is a function $\sqcap$ mapping all such functions to their infima. Then we recognise (34) as a Galois connection. Specifically,

$$
\begin{array}{ll}
 & \forall\langle b::\ a \sqsubseteq f.b\rangle \\
\equiv & \{ \quad \bullet \quad \text{define the function } K \in (\mathcal{A}\leftarrow\mathcal{B})\leftarrow\mathcal{A} \text{ by } (K.a).b = a \quad \} \\
 & \forall\langle b::\ (K.a).b \sqsubseteq f.b\rangle \\
\equiv & \{ \quad \text{definition of } \dot{\sqsubseteq} \text{ (pointwise ordering on functions)} \quad \} \\
 & K.a \mathrel{\dot{\sqsubseteq}} f \quad .
\end{array}
$$

Thus, our supposition becomes that there is a function $\sqcap$ that is the upper adjoint of the so-called “constant combinator” of type $(\mathcal{A}\leftarrow\mathcal{B})\leftarrow\mathcal{A}$ defined by

$$(K.a).b = a$$

for all $a{\in}\mathcal{A}$ and $b{\in}\mathcal{B}$. That is, for all $a{\in}A$ and $f \in \mathcal{A}\leftarrow\mathcal{B}$,

(35) $a \sqsubseteq \sqcap f \ \equiv\ K.a \mathrel{\dot{\sqsubseteq}} f$.

If this is the case we say that the poset $\mathcal{A}$ is $\mathcal{B}$-*complete*. If a poset $\mathcal{A}$ is $\mathcal{B}$-complete for all $\mathcal{B}$ we say that it is *complete*.

4.1.2 Shape Posets

The poset $\mathcal{B}$ is called the *shape poset*. By varying $\mathcal{B}$ we can consider different “types” or “shapes” of infima. For instance, if we take $\mathcal{B}$ to be $\mathbf{2}$, the two-point set $\{0,1\}$ ordered by equality, then the set of functions to $\mathcal{A}$ from $\mathcal{B}$ is in (1–1) correspondence with pairs of elements (a_0,a_1) (to be precise: $f{\mapsto}(f.0\,,f.1)$ and $(a_0,a_1){\mapsto}f$ where $f.0 = a_0$ and $f.1 = a_1$). Via this correspondence, the function K is the doubling function: $(K.a).b = \triangle a = (a,a)$, and writing $f.0 \sqcap f.1$ instead of $\sqcap f$, the Galois connection (35) simplifies to

$$a \sqsubseteq x \sqcap y \ \equiv\ (a,a) \sqsubseteq (x,y) \quad .$$

That is,

$$a \sqsubseteq x \sqcap y \ \equiv\ a \sqsubseteq x \wedge a \sqsubseteq y \quad .$$

This is the Galois connection defining the infimum of a bag of two elements (of which the Galois connection defining the minimum of two numbers is a special case).

If $\mathcal{B}$ is the empty poset, ϕ, then there is exactly one function of type $\mathcal{A}\leftarrow\mathcal{B}$. The right side of (35) is vacuously true and, thus, for all $a\in\mathcal{A}$ and $f\in\mathcal{A}\leftarrow\phi$,

$$a\sqsubseteq\sqcap f \quad .$$

In words, the poset $\mathcal{A}$ is ϕ-complete equivales $\mathcal{A}$ has a greatest element.

If $\mathcal{B}$ equals $\mathcal{A}$ then the set of functions of type $\mathcal{A}\leftarrow\mathcal{B}$ includes the identity function. The infimum of the identity function is a solution of the equation

$$x :: \ \forall\langle a{::}\ a\sqsubseteq x \equiv \forall\langle b{::}\ a\sqsubseteq b\rangle\rangle \quad .$$

and thus, by instantiating a to x, a solution of

$$x :: \ \forall\langle b{::}\, x\sqsubseteq b\rangle \quad .$$

The infimum of the identity function is thus the least element in the set $\mathcal{A}$.

A final example is the case that $\mathcal{B}$ is $(\mathbb{N},\geq)$ (the natural numbers ordered by the at-least relation). Functions in $\mathcal{A}\leftarrow\mathcal{B}$ are then in (1–1) correspondence with so-called (descending) *chains* — sets of elements a_i $(0\leq i)$ such that $a_0\sqsupseteq a_1\sqsupseteq a_2\sqsupseteq\ldots$. To say that $\mathcal{A}$ is $(\mathbb{N},\geq)$-complete is equivalent to all such chains having an infimum in $\mathcal{A}$.

4.1.3 Suprema and Cocompleteness

The notion dual to infimum is "supremum" and the notion dual to completeness is "cocompleteness". Suppose $(\mathcal{A},\sqsubseteq)$ and $(\mathcal{B},\preceq)$ are partially ordered sets and $f\in\mathcal{A}\leftarrow\mathcal{B}$ is a monotonic function. Then a *supremum* of f is a solution of the equation:

(36) $\quad x :: \ \forall\langle a{::}\ x\sqsubseteq a \equiv \forall\langle b{::}\ f.b\sqsubseteq a\rangle\rangle \quad .$

As for infima, equation (36) need not have a solution. If it does, for a given f, we denote its solution by $\sqcup f$. By definition, then,

(37) $\quad \forall\langle a{::}\ \sqcup f\sqsubseteq a\equiv\forall\langle b{::}\ f.b\sqsubseteq a\rangle\rangle \quad .$

The poset $\mathcal{A}$ is $\mathcal{B}$*-cocomplete* if there is a function $\sqcup$ that is the lower adjoint of the constant combinator of type $(\mathcal{A}\leftarrow\mathcal{B})\leftarrow\mathcal{A}$. That is, for all $a\in A$ and $f\in\mathcal{A}\leftarrow\mathcal{B}$,

(38) $\quad \sqcup f\sqsubseteq a \ \equiv\ f\,\dot{\sqsubseteq}\,K.a \quad .$

If a poset $\mathcal{A}$ is $\mathcal{B}$-cocomplete for all $\mathcal{B}$ we say that it is *cocomplete*. It can be shown, using the techniques developed here, that completeness and cocompleteness of a poset are equivalent. So the term "cocomplete" (used without qualifying shape poset) is redundant.

The fact that extrema are defined via Galois connections immediately suggests a number of useful calculation properties. We mention just two. First, both extremum operators are monotonic. That is,

$$f \mathrel{\dot{\sqsubseteq}} g \;\Rightarrow\; \sqcap f \sqsubseteq \sqcap g \wedge \sqcup f \sqsubseteq \sqcup g \quad .$$

Second, instantiating a to $\sqcap f$ in (34) we get the *cancellation* property:

$$\forall\langle b::\ \sqcap f \sqsubseteq f.b\rangle \quad .$$

In words, $\sqcap f$ is a lower bound on (the range of) f. Dually,

$$\forall\langle b::\ f.b \sqsubseteq \sqcup f\rangle \quad .$$

That is, $\sqcup f$ is an upper bound (on the range of) f.

4.2 Extremum Preservation Properties

Suppose $\mathcal{A}$ and $\mathcal{B}$ are partially ordered sets and suppose $f \in \mathcal{A} \leftarrow \mathcal{B}$. An important consideration is whether f preserves infima (or suprema) of a certain shape. If f preserves infima of shape $\mathcal{C}$ we say that f is $\mathcal{C}$-inf-preserving. If f preserves suprema of shape $\mathcal{C}$ we say that f is $\mathcal{C}$-sup-preserving. The precise formulation that we use in calculations is as follows.

Definition 39 Suppose $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are partially ordered sets such that $\mathcal{B}$ is $\mathcal{C}$-complete. Then function $f \in \mathcal{A} \leftarrow \mathcal{B}$ is $\mathcal{C}$*-inf-preserving* if, for all functions $g \in \mathcal{B} \leftarrow \mathcal{C}$,

(40) $\quad \forall\langle a::\ a \sqsubseteq f.(\sqcap g) \equiv K.a \mathrel{\dot{\sqsubseteq}} f \bullet g\rangle \quad .$

Dually, suppose that $\mathcal{B}$ is $\mathcal{C}$-cocomplete. Then function $f \in \mathcal{A} \leftarrow \mathcal{B}$ is said to be $\mathcal{C}$*-sup-preserving* if, for all functions $g \in \mathcal{B} \leftarrow \mathcal{C}$,

(41) $\quad \forall\langle a::\ f.(\sqcup g) \sqsubseteq a \equiv f \bullet g \mathrel{\dot{\sqsubseteq}} K.a\rangle \quad .$

If $\mathcal{B}$ is complete, we say that f is *inf-preserving* if it satisfies (40) for all $\mathcal{C}$ and all g. *Sup-preserving* is defined dually.

□

The definition of inf-preserving does not require that $\mathcal{A}$ be complete. If that is the case, and we abuse notation by using $\sqcap$ to denote the infimum operator for both $\mathcal{A}$ and $\mathcal{B}$, then $f \in \mathcal{A} \leftarrow \mathcal{B}$ is inf-preserving if for all functions g with range $\mathcal{B}$

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

Although more complicated, we are obliged to use equation (40) if we want to establish properties of inf-preservation, whereas the less complicated equation (42) is the equation we will use when we want to establish properties of a function f that is known to be inf-preserving, and the posets concerned are known to be complete.

A predicate is a function with range Bool, the two element set with elements true and false. Ordering Bool by implication ($\Rightarrow$) the infimum of a (monotonic) predicate p is the universal quantification $\forall p$ (that is $\forall\langle x{::}\,p.x\rangle$). Also that predicate p is $\mathcal{C}$-inf-preserving means that $p.(\sqcap g) \equiv \forall(p\bullet g)$ for all functions g with range the domain of p and domain $\mathcal{C}$. Formally,

$$
\begin{array}{ll}
 & p \in \mathsf{Bool}\leftarrow\mathcal{B} \text{ is } \mathcal{C}\text{-inf-preserving} \\
\equiv & \qquad \{ \qquad \text{definition} \quad \} \\
 & \forall\langle g\colon g\in\mathcal{B}\leftarrow\mathcal{C}\colon \forall\langle a\colon\ a\in\mathsf{Bool}\colon\ a\Rightarrow p.(\sqcap g) \equiv \forall\langle x{::}\ a\Rightarrow p.(g.x)\rangle\rangle\rangle \\
\equiv & \qquad \{ \qquad \text{simplification using } \mathsf{true}\Rightarrow q \ \equiv\ q \\
 & \qquad\qquad\quad \text{and } \mathsf{false}\Rightarrow q \ \equiv\ \mathsf{true} \quad \} \\
 & \forall\langle g\colon g\in\mathcal{B}\leftarrow\mathcal{C}\colon p.(\sqcap g)\equiv\forall\langle x{::}\,p.(g.x)\rangle\rangle \quad .
\end{array}
$$

That is, for predicate p with domain $\mathcal{B}$,

(43) $\quad p$ is $\mathcal{C}$-inf-preserving $\ \equiv\ \forall\langle g\colon g\in\mathcal{B}\leftarrow\mathcal{C}\colon p.(\sqcap g)\equiv\forall\langle x{::}\,p.(g.x)\rangle\rangle$.

The dual of (43) is:

(44) $\quad p$ is $\mathcal{C}$-sup-preserving $\ \equiv\ \forall\langle g\colon g\in\mathcal{B}\leftarrow\mathcal{C}\colon p.(\sqcup g)\equiv\forall\langle x{::}\,p.(g.x)\rangle\rangle$.

Just as for (co)completeness properties, various terminology exists for specific instances of $\mathcal{C}$. Taking $\mathcal{C}$ to be $(\mathsf{Bool},\Rightarrow)$, the booleans ordered by implication, a $\mathcal{C}$-(inf or sup)-preserving function, f, is a function that maps values x and y such that $x\sqsubseteq y$ to values $f.x$ and $f.y$ such that $f.x\sqsubseteq f.y$. Thus a function is $(\mathsf{Bool},\Rightarrow)$-(inf and/or sup)-preserving if and only if it is monotonic.

If $\mathcal{C}$ is the empty set then f is $\mathcal{C}$-sup-preserving means that f maps the least element of $\mathcal{B}$ to the least element of $\mathcal{A}$. The terminology that is often used in the computing science literature is "f is *strict*". Sometimes one says f is *bottom-strict*, "bottom" being a common name in the computing science literature for the least element in a set. "Top-strict" then means preserving the greatest element, i.e. empty set-inf-preserving.

If $\mathcal{C}$ is the set of natural numbers ordered by the usual at-most relation then $\mathcal{C}$-sup-preservation of a function is often called ω-*continuity*.

If $\mathcal{C}$ is a non-empty finite set then $\mathcal{C}$-inf-preservation of f is equivalent to f preserving binary infima (i.e. $f.(x\sqcap y) = f.x\sqcap f.y$). This is sometimes referred to as *positive inf-preservation* of f.

Several examples of finite preservation properties have already been discussed. All of these examples are instances of a fundamental extremum preservation property of adjoint functions which we present in the next section.

4.3 Existence Theorem

In this section we derive a fundamental theorem on the existence of Galois connections. To simplify matters we often make the assumption that we are dealing with complete posets, particularly in the beginning of the discussion. The section ends with a discussion of properties of adjoint functions when this assumption is not valid.

4.3.1 Pair Algebras

Recall the motivating discussion at the beginning of section 4. There we went from the question, given a relation R, when is there a function F such that $(x,y)\in R \;\equiv\; F.x\sqsubseteq y$, to the question, given a predicate p, when is it the case that $p.y \;\equiv\; a\sqsubseteq y$. The answer to the latter question is the subject of lemma 46 below, following which we can immediately answer the former question. It is but a short step from here to the fundamental theorem on the existence of Galois connections. Along the way, we discuss several other examples of the lemmas we have formulated.

We begin with a trivial, but insightful lemma.

Lemma 45 Function f is $\mathcal{C}$-inf-preserving is the same as, for all a, the predicate $\langle x{::}\; a\sqsubseteq f.x\rangle$ is $\mathcal{C}$-inf-preserving.

Proof

$$
\begin{array}{ll}
 & f \text{ is } \mathcal{C}\text{-inf-preserving} \\
\equiv & \{ \quad \text{definition of } \mathcal{C}\text{-inf-preserving: (40),} \\
 & \quad\;\; \text{where } g \text{ ranges over functions with domain } \mathcal{C} \quad \} \\
 & \forall\langle g{::}\; \forall\langle a{::}\; a\sqsubseteq f.(\sqcap g) \;\equiv\; \forall\langle x{::}\; a\sqsubseteq f.(g.x)\rangle\rangle\rangle \\
\equiv & \{ \quad \text{nesting of quantifications} \quad \} \\
 & \forall\langle a{::}\; \forall\langle g{::}\; a\sqsubseteq f.(\sqcap g) \;\equiv\; \forall\langle x{::}\; a\sqsubseteq f.(g.x)\rangle\rangle\rangle \\
\equiv & \{ \quad (43) \quad \} \\
 & \forall\langle a{::}\; \langle x{::}\; a\sqsubseteq f.x\rangle \text{ is } \mathcal{C}\text{-inf-preserving}\rangle \quad .
\end{array}
$$

□

Noting that the identity function is inf-preserving, an immediate corollary of lemma 45 is that, for each a, the predicate $\langle x{::}\; a\sqsubseteq x\rangle$ is inf-preserving. This suggests that we explore whether the converse is true. And, indeed, it is:

Lemma 46 Suppose p is a predicate on a complete poset. Then

$$p \text{ is inf-preserving} \quad \equiv \quad \exists\langle a:: \ p \equiv \langle x:: a \sqsubseteq x\rangle\rangle \ .$$

Furthermore, if p is inf-preserving

$$\sqcap\langle x: p.x: x\rangle$$

is the unique solution of the equation

$$a:: \quad p \equiv \langle x:: a \sqsubseteq x\rangle \ .$$

□

Applying lemma 46 as we indicated we would be doing we have answered our original question about relations R:

Theorem 47 (Pair Algebras) Suppose $\mathcal{B}$ is a set and $(\mathcal{A}, \sqsubseteq)$ is a complete poset. Suppose $R \subseteq \mathcal{B} \times \mathcal{A}$ is a relation between the two sets. Then the following two statements are equivalent.

The function F defined by

$$F.x = \sqcap\langle y: (x,y) \in R: y\rangle$$

satisfies, for all $x \in \mathcal{B}$ and all $y \in \mathcal{A}$,

$$(x,y) \in R \equiv F.x \sqsubseteq y \ .$$

For all $x \in \mathcal{B}$, the predicate $\langle y:: (x,y) \in R\rangle$ is inf-preserving.

□

4.3.2 Examples of Pair Algebras

A simple example of the pair-algebra theorem is provided by the membership relation. For all sets S and all x (in a given universe of which S is a subset) we have:

$$x \in S \equiv \{x\} \subseteq S \ .$$

This statement has the form

$$(x,S) \in R \equiv F.x \subseteq S$$

where the relation R is the membership relation. We thus deduce that the predicate $(x\in)$ is inf-preserving. That is, for all bags of sets $\mathcal{S}$,

$$x \in \cap\mathcal{S} \equiv \forall\langle S: S \in \mathcal{S}: x \in S\rangle \ .$$

A common way to exploit the pair algebra theorem is to take a function or relation and extend it to a relation between sets in such a way that the infimum and supremum preserving properties are automatically satisfied. In fact this is the basis of the so-called “Galois correspondence” between groups and fields put forward by Évariste Galois in 1832. That example involves too much background terminology to usefully include it here. A simpler example is the following.

Consider a relation R on two sets A and B. (Thus $R \subseteq A \times B$.) These sets need not be ordered. Define the relation $\overline{R}$ on subsets X and Y of A and B, respectively, by

$$(X,Y) \in \overline{R} \;\equiv\; X \times Y \subseteq R \quad .$$

Now, the functions $\langle X{::}\,X \times Y\rangle$ and $\langle Y{::}\,X \times Y\rangle$ preserve arbitrary unions of sets, and hence so do the predicates $\langle X{::}\,X \times Y \subseteq R\rangle$ and $\langle Y{::}\,X \times Y \subseteq R\rangle$. Thus, by the pair-algebra theorem (taking care with the direction of the orderings) there are functions F_R and G_R such that for all subsets X and Y of A and B, respectively,

$$F_R.X \supseteq Y \;\equiv\; X \times Y \subseteq R \;\equiv\; X \subseteq G_R.Y \quad .$$

In words, $F_R.X$ is the largest set Y such that every element of X is related by R to every element of Y. Similarly, $G_R.Y$ is the largest set X such that every element of X is related by R to every element of Y. The functions F_R and G_R are called *polarities*; F_R is the *left polar* and G_R is the *right polar* of relation R. This Galois connection is the basis of so-called *concept lattices* [DP90, GW99]. In this application area, the set A is a set of objects, and the set B is a set of attributes (or “concepts”). For example, we may take A to be the set of planets of the sun (Mars, Venus, Jupiter, etc.) and B to be attributes like “has a moon” and “is nearer to the sun than the Earth”. The relation R is then “satisfies the predicate”. We may then ask for all the planets that both have a moon and are nearer to the sun than the Earth.

A more substantial example of the pair-algebra theorem, that stands out in the computing science literature, is provided by conditional correctness assertions of program statements.

Suppose S is a program statement and p and q are predicates on the state space of S. Then, one writes $\{p\}S\{q\}$ if after successful execution of statement S beginning in a state satisfying the predicate p the resulting state will satisfy predicate q. In such a case one says that statement S is *conditionally correct* with respect to precondition p and postcondition q. (“Conditional” refers to the fact that satisfying predicate q is conditional on the termination of statement S.)

In this way each program statement S defines a relation on state predicates. This relation is such that, for all bags of predicates P,

$$\{\exists\langle p{:}\, p{\in}P{:}\, p\rangle\}S\{q\} \;\equiv\; \forall\langle p{:}\, p{\in}P{:}\, \{p\}S\{q\}\rangle \quad .$$

That is, for each state predicate q the predicate $\langle p{::}\{p\}S\{q\}\rangle$ preserves suprema in the poset of predicates ordered by implication (equivalently, preserves infima in the poset of predicates ordered by follows-from). Thus, by the dual of the pair-algebra theorem with implication as the ordering on predicates, for each statement S and predicate q there is a predicate $\mathsf{wlp}(S,q)$ satisfying

$$\{p\}S\{q\} \;\equiv\; p \Rightarrow \mathsf{wlp}(S,q) \quad .$$

The abbreviation "wlp" stands for "weakest liberal precondition".

Before leaving this example, let us note that it is also the case that, for all bags of predicates Q,

$$\{p\}S\{\forall\langle q{:}\, q{\in}Q{:}\, q\rangle\} \;\equiv\; \forall\langle q{:}\, q{\in}Q{:}\, \{p\}S\{q\}\rangle \quad .$$

There is thus also a predicate $\mathsf{slp}(S,\, p)$ satisfying

$$\{p\}S\{q\} \;\equiv\; \mathsf{slp}(S,\, p) \Rightarrow q \quad .$$

Combining this equation with the equation for the weakest liberal precondition, we thus have the Galois connection: for all predicates p and q,

$$\mathsf{slp}(S,\, p) \Rightarrow q \;\equiv\; p \Rightarrow \mathsf{wlp}(S,\, q) \quad .$$

The abbreviation "slp" stands for "strongest liberal postcondition".

4.3.3 The Existence Theorem

Now that we have seen several concrete examples, let us state the fundamental theorem.

Theorem 48 (Fundamental Theorem) Suppose that $\mathcal{B}$ is a poset and $\mathcal{A}$ is a complete poset. Then a monotonic function $G \in \mathcal{B} \leftarrow \mathcal{A}$ is an upper adjoint in a Galois connection equivales G is inf-preserving.

Dually, a monotonic function $F \in \mathcal{A} \leftarrow \mathcal{B}$ is a lower adjoint in a Galois connection equivales F is sup-preserving.

Proof Define the relation R by $(x,y) \in R \equiv x \preceq G.y$ and function F as in theorem 47. Then,

$$\begin{array}{ll} & G \text{ is inf-preserving} \\ \equiv & \{ \quad \text{lemma 45} \quad \} \\ & \forall\langle x{::}\;\; \langle y{::}\; x \preceq G.y\rangle \text{ is inf-preserving}\,\rangle \\ \equiv & \{ \quad \text{theorem 47 with } R \text{ and } F \text{ defined as above} \quad \} \\ & \forall\langle x,y{::}\; F.x \sqsubseteq y \;\equiv\; x \preceq G.y\rangle \quad . \end{array}$$

□

We mentioned a couple of instances of theorem 48 immediately prior to its derivation. There are many other instances we could give. The predicate calculus is a source of several. We saw earlier that for all predicates p the function $(p\wedge)$ is a lower adjoint. Specifically,

$$(p \wedge q \Rightarrow r) \;\equiv\; (p \Rightarrow (q \Rightarrow r)) \quad .$$

The supremum operator in the lattice of predicates ordered by implication is existential quantification since

$$(\exists\langle x{::}\, p.x\rangle \Rightarrow q) \;\equiv\; \forall\langle x{::}\; p.x \Rightarrow q\rangle \quad .$$

Instantiating theorem 48 we thus conclude that $(p\wedge)$ commutes with existential quantification. That is,

$$p \wedge \exists\langle x{::}\, q.x\rangle \equiv \exists\langle x{::}\; p \wedge q.x\rangle \quad .$$

The dual theorem is that $(p\Rightarrow)$ commutes with universal quantification:

$$(p \Rightarrow \forall\langle x{::}\, q.x\rangle) \;\equiv\; \forall\langle x{::}\; p \Rightarrow q.x\rangle \quad .$$

A more enlightening example is afforded by negation. Recall that

$$(\neg p \Rightarrow q) \;\equiv\; (p \Leftarrow \neg q) \quad .$$

Now, in order to instantiate lemma 55 we need to be very clear about the partially ordered sets involved: instantiate $\mathcal{A}$ to the predicates ordered by implication and $\mathcal{C}$ to the predicates ordered by follows-from. Observe that the supremum operator in $\mathcal{A}$ is existential quantification, and in $\mathcal{C}$ is universal quantification. Thus by application of the lemma:

$$\neg\forall\langle x{::}\, p.x\rangle \equiv \exists\langle x{::}\, \neg(p.x)\rangle \quad .$$

This example illustrates why in general it is necessary to take great care with the precise definitions of the orderings on the posets $\mathcal{A}$ and $\mathcal{B}$ when applying theorem 48.

The above examples all illustrate the fact that, given a Galois connection, we can infer that the lower adjoint preserves suprema (and the upper adjoint preserves infima). The converse property is most often used to ascertain that a function is a lower adjoint (or is an upper adjoint) without it being necessary to know what the corresponding upper adjoint of the function is. Indeed it is often the case that the upper adjoint has a clumsy definition that one wishes to avoid using explicitly at all costs.

Exercise 49 Consider an arbitrary set $\mathcal{U}$. For each x in $\mathcal{U}$ the predicate $(x\in)$ maps a subset P of $\mathcal{U}$ to the boolean value true if x is an element of P and otherwise to false. The predicate $(x\in)$ preserves set union and set intersection. That is, for all bags $\mathcal{S}$ of subsets of $\mathcal{U}$,

$$x \in \cup\mathcal{S} \equiv \exists\langle P : P\in\mathcal{S} : x\in P\rangle$$

and

$$x \in \cap\mathcal{S} \equiv \forall\langle P : P\in\mathcal{S} : x\in P\rangle \quad .$$

According to the fundamental theorem and its dual, the predicate $(x\in)$ thus has a lower and an upper adjoint. *Construct a closed formula for the upper and lower adjoints of* $(x\in)$.

□

Exercise 50 Consider the predicate $(\neq\phi)$ (where ϕ denotes the empty set) on subsets of a given set $\mathcal{U}$. *Using lemma 46 (or otherwise) determine in what sense the predicate preserves set union. Hence construct a Galois connection in which the predicate is the lower adjoint.*

□

Exercise 51 Suppose $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ are complete posets. Theorem 48 states that there is a function mapping the sup-preserving functions $F \in \mathcal{A}\leftarrow\mathcal{B}$ to the inf-preserving functions $G \in \mathcal{B}\leftarrow\mathcal{A}$ and, vice-versa, there is a function mapping the inf-preserving functions $G \in \mathcal{B}\leftarrow\mathcal{A}$ to the sup-preserving functions $F \in \mathcal{A}\leftarrow\mathcal{B}$. Denote these functions by a superscript $^\sharp$ and a superscript $^\flat$, respectively, so that $F^\sharp$ denotes the upper adjoint of sup-preserving function F, and $G^\flat$ denotes the lower adjoint of inf-preserving function G. In other words, for all x and y,

$$F.x \sqsubseteq y \equiv x \sqsubseteq F^\sharp.y$$

and

$$G^\flat.x \sqsubseteq y \equiv x \sqsubseteq G.y \quad .$$

Show that $^\sharp$ *and* $^\flat$ *are inverse poset isomorphisms.* (Take care with the orderings!)

□

Exercise 52 *Prove*: a poset is complete is the same as the poset is cocomplete.

□

4.3.4 Existential Extremum Preservation

In the statement of theorems 47 and 48 we have assumed that a poset is complete. Sometimes this assumption is not justified, but we would still like to know what the extremum preservation properties of the adjoints of a Galois connection are. For this purpose, we make use of (27).

Suppose $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are partially ordered sets and both $\mathcal{A}$ and $\mathcal{C}$ are $\mathcal{B}$-cocomplete. We recall that this means that all functions of type $\mathcal{A}{\leftarrow}\mathcal{B}$ or $\mathcal{C}{\leftarrow}\mathcal{B}$ have a supremum. Equivalently, the K-combinators of type $(\mathcal{A}{\leftarrow}\mathcal{B}){\leftarrow}\mathcal{A}$ and $(\mathcal{C}{\leftarrow}\mathcal{B}){\leftarrow}\mathcal{C}$ each have lower adjoints. Denoting the K-combinator of type $(\mathcal{A}{\leftarrow}\mathcal{B}){\leftarrow}\mathcal{A}$ by $\mathsf{K}_{\mathcal{A}}$ and the corresponding supremum operator by $\sqcup_{\mathcal{A}}$, we can instantiate (27) to

(53) $\quad \mathsf{F}_0{\bullet}\sqcup_{\mathcal{C}} = \sqcup_{\mathcal{A}}{\bullet}\mathsf{F}_1 \;\equiv\; \mathsf{K}_{\mathcal{C}}{\bullet}\mathsf{G}_0 = \mathsf{G}_1{\bullet}\mathsf{K}_{\mathcal{A}}$.

This holds for all functions F_0 , G_0 , F_1 and G_1 such that $\mathsf{F}_0 \in \mathcal{A}{\leftarrow}\mathcal{C}$, $\mathsf{G}_0 \in \mathcal{C}{\leftarrow}\mathcal{A}$, $\mathsf{F}_1 \in (\mathcal{A}{\leftarrow}\mathcal{B}){\leftarrow}(\mathcal{C}{\leftarrow}\mathcal{B})$, $\mathsf{G}_1 \in (\mathcal{C}{\leftarrow}\mathcal{B}){\leftarrow}(\mathcal{A}{\leftarrow}\mathcal{B})$, and $(\mathsf{F}_0,\mathsf{G}_0)$ and $(\mathsf{F}_1,\mathsf{G}_1)$ are both Galois pairs. If you ignore the subscripts on $\sqcup$ and K (for instance, when $\mathcal{A}$ and $\mathcal{C}$ are identical) then (53) is seen to be an instance of (28). The extra generality proves to be necessary for some examples to be discussed shortly.

Now, the K-combinator is a particularly simple function. It is relatively easy to explore its commutativity properties. Then, with the aid of (53), we can translate those properties into properties of suprema. Let's do that.

Suppose $\mathsf{G} \in \mathcal{C}{\leftarrow}\mathcal{A}$. It is easy to find a function h such that $\mathsf{K}_{\mathcal{C}}{\bullet}\mathsf{G} = \mathsf{h}{\bullet}\mathsf{K}_{\mathcal{A}}$. Specifically,

(54) $\quad \mathsf{K}_{\mathcal{C}}{\bullet}\mathsf{G} = (\mathsf{G}{\bullet}){\bullet}\mathsf{K}_{\mathcal{A}}$.

Recalling that the function $\mathsf{G}{\bullet}$ has lower adjoint $\mathsf{F}{\bullet}$ given that G has lower adjoint F we can now instantiate (53) obtaining the lemma:

Lemma 55 Suppose that $\mathsf{F} \in \mathcal{A}{\leftarrow}\mathcal{C}$ is a lower adjoint in a Galois connection. Then F preserves suprema. That is, for each partially ordered set $\mathcal{B}$ such that both $\mathcal{A}$ and $\mathcal{C}$ are $\mathcal{B}$-cocomplete,

$$\mathsf{F}{\bullet}\sqcup_{\mathcal{C}} = \sqcup_{\mathcal{A}}{\bullet}(\mathsf{F}{\bullet}) \quad .$$

Equivalently, for each function $\mathsf{h} \in \mathcal{C}{\leftarrow}\mathcal{B}$,

$$\mathsf{F}.(\sqcup_{\mathcal{C}}.\mathsf{h}) = \sqcup_{\mathcal{A}}.(\mathsf{F}{\bullet}\mathsf{h}) \quad .$$

□

Exercise 56 Another consequence of (54) is that for an arbitrary monotonic function f of type $\mathcal{A} \leftarrow \mathcal{C}$

$$\sqcup_{\mathcal{A}}.(f \bullet h) \sqsubseteq f.(\sqcup_{\mathcal{C}}.h) \quad .$$

Prove this property.

□

5 Unity of Opposites

In this section we want to show that adjoint functions create complete posets. Specifically, we want to show that if the pair (F, G) is a Galois connection between posets $\mathcal{A}$ and $\mathcal{B}$, and either one of $\mathcal{A}$ or $\mathcal{B}$ is $\mathcal{C}$-complete, then the images of F and G (the set of elements x that equal $F.y$ for some y, and the set of y that equal $G.x$ for some x) are also $\mathcal{C}$-complete. For brevity we denote the image of F by $F.\mathcal{B}$ and the image of G by $G.\mathcal{A}$.

We make the assumption that $\mathcal{B}$ is $\mathcal{C}$-complete. We begin by showing that $G.\mathcal{A}$ is $\mathcal{C}$-complete and then show that $F.\mathcal{B}$ is $\mathcal{C}$-complete. The dual theorem is that both of $F.\mathcal{B}$ and $G.\mathcal{A}$ are $\mathcal{C}$-complete if $\mathcal{A}$ is $\mathcal{C}$-complete. Thus, both are $\mathcal{C}$-complete if $\mathcal{A}$ or $\mathcal{B}$ is $\mathcal{C}$-complete.

5.1 The Prefix Lemma

The key property that we use is that x is an element of the image of G is the same as $G.(F.x) \sqsubseteq x$. Thus the image of G is the set of so-called "prefix points" of the function $G \bullet F$. Prefix points will figure highly in later sections on fixed point calculus. For the moment, it suffices for us to observe the so-called "prefix lemma" which asserts that the prefix points of a monotonic endofunction on a $\mathcal{C}$-complete poset form a $\mathcal{C}$-complete poset. (The prefix lemma emerges somewhat out of the blue in this development. We encounter the lemma again in section 7 where we hazard a guess as to how the lemma may have been first observed.)

The first task —which we leave to the reader— is to show that

$$(57) \qquad x \in G.\mathcal{A} \;\equiv\; G.(F.x) \sqsubseteq x \quad .$$

Property (57) is important because it identifies the image of G as the set of "prefix points" of the function $G \bullet F$. In general, if h is a monotonic endofunction on some poset $\mathcal{A}$ then $y \in \mathcal{A}$ is said to be a a *prefix point* of h if $h.y \sqsubseteq y$. The reason that this is important is because of the following lemma.

Lemma 58 (Prefix lemma) Suppose $\mathcal{B}$ and $\mathcal{C}$ are posets and suppose $\mathcal{B}$ is $\mathcal{C}$-complete. Suppose also that h is a monotonic endofunction on $\mathcal{B}$. Then the set, $\mathrm{Pre}.h$, of prefix points of h is also $\mathcal{C}$-complete.

Proof Suppose f is any function with range $\mathrm{Pre}.h$ and domain $\mathcal{C}$. Since $\mathcal{B}$ is $\mathcal{C}$-complete and $\mathrm{Pre}.h$ is a subset of $\mathcal{B}$ the function f has an infimum, $\sqcap_{\mathcal{B}}.f$, in $\mathcal{B}$. Letting P denote $\mathrm{Pre}.h$, the required infimum of f is a solution of the equation:

(59) $x\colon\ x{\in}P\colon\ \forall\langle y\colon\ y{\in}P\colon\ y \sqsubseteq x \equiv K.y \mathrel{\dot{\sqsubseteq}} f\rangle$

whereas $\sqcap_{\mathcal{B}}.f$ is a solution of the equation:

$$x\colon\ x{\in}\mathcal{B}\colon\ \forall\langle y\colon\ y{\in}\mathcal{B}\colon\ y \sqsubseteq x \equiv K.y \mathrel{\dot{\sqsubseteq}} f\rangle\ .$$

By specialising the latter universal quantification to elements of P (which is a subset of $\mathcal{B}$) it is thus the case that $\sqcap_{\mathcal{B}}.f$ solves:

(60) $x\colon\ x{\in}\mathcal{B}\colon\ \forall\langle y\colon\ y{\in}P\colon\ y \sqsubseteq x \equiv K.y \mathrel{\dot{\sqsubseteq}} f\rangle$.

Comparing (59) with (60) it is clear that $\sqcap_{\mathcal{B}}.f$ solves (59) if it is an element of P. That is, we must show that $\sqcap_{\mathcal{B}}.f$ is a prefix point of h. This we do as follows.

$\quad\ \sqcap_{\mathcal{B}}.f \in \mathrm{Pre}.h$

$\equiv\quad$ { definition of $\mathrm{Pre}.h$ }

$\quad\ h.(\sqcap_{\mathcal{B}}.f) \sqsubseteq \sqcap_{\mathcal{B}}.f$

$\equiv\quad$ { definition of $\sqcap_{\mathcal{B}}.f$,

$\qquad\quad$ where dummy c ranges over $\mathcal{C}$ }

$\quad\ \forall\langle c{::}\ h.(\sqcap_{\mathcal{B}}.f) \sqsubseteq f.c\rangle$

$\Leftarrow\quad$ { range of f is $\mathrm{Pre}.h$. Thus $\forall\langle c{::}\ h.(f.c) \sqsubseteq f.c\rangle$.

$\qquad\quad$ Transitivity of $\sqsubseteq$. }

$\quad\ \forall\langle c{::}\ h.(\sqcap_{\mathcal{B}}.f) \sqsubseteq h.(f.c)\rangle$

$\Leftarrow\quad$ { h is monotonic }

$\quad\ \forall\langle c{::}\ \sqcap_{\mathcal{B}}.f \sqsubseteq f.c\rangle$

$\equiv\quad$ { cancellation }

$\quad\ \mathsf{true}$.

□

By instantiating h to $G{\bullet}F$, it follows immediately from the prefix lemma that if $\mathcal{B}$ is $\mathcal{C}$-complete, $\mathrm{Pre}.(G{\bullet}F)$ is $\mathcal{C}$-complete. Combining this with (57), we get that if $\mathcal{B}$ is a $\mathcal{C}$-complete poset then $G.\mathcal{A}$ is also a $\mathcal{C}$-complete poset.

5.2 The Theorem

The proof of the prefix lemma shows that the infimum operator in $G.\mathcal{A}$ coincides with the infimum operator in $\mathcal{B}$. The supremum operators of $G.\mathcal{A}$ and $\mathcal{B}$ do not necessarily coincide. Let f be any function with range $G.\mathcal{A}$ and domain $\mathcal{C}$. The supremum of f in $G.\mathcal{A}$ is a solution of the equation

$$X\colon X\in G.\mathcal{A}\colon \forall\langle x\colon x\in G.\mathcal{A}\colon X\sqsubseteq x \equiv \forall\langle y{::}\ f.y\sqsubseteq x\rangle\rangle$$

whereas the supremum of f in $\mathcal{B}$ is a solution of the equation

$$Y\colon\ Y\in\mathcal{B}\colon\ \forall\langle x\colon x\in\mathcal{B}\colon Y\sqsubseteq x \equiv \forall\langle y{::}\ f.y\sqsubseteq x\rangle\rangle\quad .$$

Suppose Y is the supremum of f in $\mathcal{B}$ (that is, $Y=\sqcup_{\mathcal{B}}.f$) and suppose $x\in G.\mathcal{A}$. Then, we construct the supremum of f in $G.\mathcal{A}$ as follows. The goal is to construct X such that $X\in G.\mathcal{A}$ and

$$\forall\langle x\colon x\in G.\mathcal{A}\colon X\sqsubseteq x \equiv \forall\langle y{::}\ f.y\sqsubseteq x\rangle\rangle\quad .$$

Equivalently, $X=G.a$ where $a\in\mathcal{A}$ and

$$\forall\langle x\colon\ x\in G.\mathcal{A}\colon\ G.a\sqsubseteq x \equiv \forall\langle y{::}\ f.y\sqsubseteq x\rangle\rangle\quad .$$

We now calculate a:

$$\begin{array}{ll}
 & G.a\sqsubseteq x \equiv \forall\langle y{::}\ f.y\sqsubseteq x\rangle \\
\equiv & \{\quad \text{definition of } Y,\ x\in G.\mathcal{A}\Rightarrow x\in\mathcal{B}\quad\} \\
 & G.a\sqsubseteq x \equiv Y\sqsubseteq x \\
\equiv & \{\quad \text{propositional calculus}\quad\} \\
 & (G.a\sqsubseteq x \Leftarrow Y\sqsubseteq x)\wedge(G.a\sqsubseteq x\Rightarrow Y\sqsubseteq x) \\
\equiv & \{\quad x\in G.\mathcal{A}\equiv G.(F.x)=x\quad\} \\
 & (G.a\sqsubseteq G.(F.x) \Leftarrow Y\sqsubseteq x)\wedge(G.a\sqsubseteq x\Rightarrow Y\sqsubseteq x) \\
\Leftarrow & \{\quad \text{properties of Galois-connected functions:} \\
 & \qquad G\bullet F \text{ is monotonic and } Y\sqsubseteq G.(F.Y)\quad\} \\
 & a=F.Y\quad .
\end{array}$$

In summary, we have calculated that

(61) $\quad\sqcup_{G.\mathcal{A}}.f = G.(F.(\sqcup_{\mathcal{B}}.f))\quad .$

The final task is to show that $F.\mathcal{B}$ is complete. This we do by expressing the supremum and infimum operators in $F.\mathcal{B}$ in terms of those in $\mathcal{B}$. From section 3.4 we know that the functions F and G witness a poset isomorphism between $F.\mathcal{B}$ and $G.\mathcal{A}$. We have also just shown that $G.\mathcal{A}$ is complete. It follows immediately that $F.\mathcal{B}$ is complete. That is, all supremum and infima exist in $F.\mathcal{B}$ and we can use simple equational reasoning to express them in terms of the suprema and infima in $\mathcal{B}$. Let f be a function with range $F.\mathcal{B}$. Then,

$$
\begin{array}{ll}
 & \sqcup_{F.\mathcal{B}}.f \\
= & \{ \quad x \in F.\mathcal{B} \equiv x = (F\bullet G).x \quad \} \\
 & \sqcup_{F.\mathcal{B}}.(F\bullet G\bullet f) \\
= & \{ \quad F \in F.\mathcal{B} \leftarrow G.\mathcal{A} \text{ is a poset isomorphism,} \\
 & \quad\ \ \text{fundamental theorem} \quad \} \\
 & F.(\sqcup_{G.\mathcal{A}}.(G\bullet f)) \\
= & \{ \quad (61) \quad \} \\
 & (F\bullet G\bullet F).(\sqcup_{\mathcal{B}}.(G\bullet f)) \\
= & \{ \quad F\bullet G\bullet F = F \quad \} \\
 & F.(\sqcup_{\mathcal{B}}.(G\bullet f))
\end{array}
$$

and

$$
\begin{array}{ll}
 & \sqcap_{F.\mathcal{B}}.f \\
= & \{ \quad x \in F.\mathcal{B} \equiv x = (F\bullet G).x \quad \} \\
 & (F\bullet G).(\sqcap_{F.\mathcal{B}}.f) \\
= & \{ \quad F \in F.\mathcal{B} \leftarrow G.\mathcal{A} \text{ is a poset isomorphism,} \\
 & \quad\ \ \text{fundamental theorem} \quad \} \\
 & F.(\sqcap_{G.\mathcal{A}}.(G\bullet f)) \\
= & \{ \quad \text{infimum in } G.\mathcal{A} \text{ coincides with} \\
 & \quad\ \ \text{the infimum in } \mathcal{B} \quad \} \\
 & F.(\sqcap_{\mathcal{B}}.(G\bullet f)) \quad .
\end{array}
$$

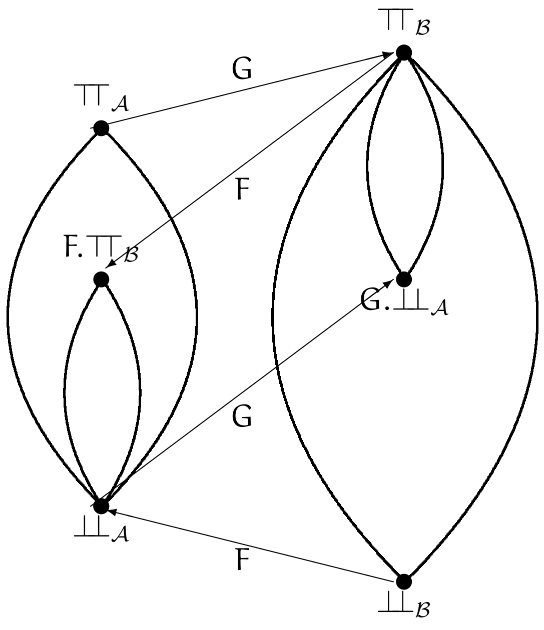
Summarising, we have proved the following theorem.

Theorem 62 (Unity of Opposites) Suppose $F \in \mathcal{A} \leftarrow \mathcal{B}$ and $G \in \mathcal{B} \leftarrow \mathcal{A}$ are Galois connected functions, F being the lower adjoint and G being the upper adjoint. Then $F.\mathcal{B}$ and $G.\mathcal{A}$ are isomorphic posets. Moreover, if one of $\mathcal{A}$ or $\mathcal{B}$ is $\mathcal{C}$-complete, for some shape poset $\mathcal{C}$, then $F.\mathcal{B}$ and $G.\mathcal{A}$ are also $\mathcal{C}$-complete. Assuming that $\mathcal{B}$ is $\mathcal{C}$-complete the supremum and infimum operators are given by

$$\begin{array}{rcl}
\sqcap_{G.\mathcal{A}}.f & = & \sqcap_{\mathcal{B}}.f \\
\sqcup_{G.\mathcal{A}}.f & = & G.(F.(\sqcup_{\mathcal{B}}.f)) \\
\sqcap_{F.\mathcal{B}}.f & = & F.(\sqcap_{\mathcal{B}}.(G\bullet f)) \\
\sqcup_{F.\mathcal{B}}.f & = & F.(\sqcup_{\mathcal{B}}.(G\bullet f)) \quad .
\end{array}$$

□

Picturing the posets $\mathcal{A}$ and $\mathcal{B}$ as sets in which larger elements are above smaller elements, the unity-of-opposites theorem is itself summarised in the following diagram. The two larger lenses picture the sets $\mathcal{A}$ (on the left) and $\mathcal{B}$ (on the right); the bottom-left lens pictures $F.\mathcal{B}$ and the top-right lens $G.\mathcal{A}$. The latter two sets are pictured as having the same size because they are isomorphic, whereas $\mathcal{A}$ and $\mathcal{B}$ are pictured as having different size because they will not be isomorphic in general. Note that F maps the least element of $\mathcal{B}$ (denoted $\perp\!\!\perp_{\mathcal{B}}$ in the diagram) to the least element of $\mathcal{A}$ (denoted $\perp\!\!\perp_{\mathcal{A}}$). Further G maps the greatest element of $\mathcal{A}$ (denoted $\top\!\!\top_{\mathcal{A}}$ in the diagram) to the greatest element of $\mathcal{B}$ (denoted $\top\!\!\top_{\mathcal{B}}$). The posets $F.\mathcal{B}$ and $G.\mathcal{A}$ are "opposites" in the sense that the former contains small elements whereas the latter contains large elements. In particular, $F.\mathcal{B}$ includes the least element of $\mathcal{A}$ and $G.\mathcal{A}$ includes the greatest element of $\mathcal{B}$. They are unified, however, by the fact that they are isomorphic.



$\top\!\!\top_{\mathcal{B}}$
G
$\top\!\!\top_{\mathcal{A}}$
F
$F.\top\!\!\top_{\mathcal{B}}$
$G.\bot\!\!\bot_{\mathcal{A}}$
G
$\bot\!\!\bot_{\mathcal{A}}$
F
$\bot\!\!\bot_{\mathcal{B}}$

Exercise 63 (Closure Operators) Closure operators play an important role in many areas of mathematics. This exercise relates Galois connections to closure operators.

There are two definitions of closure operators. According to the most common definition, a *closure operator* is an endofunction, f, on a poset $(A, \sqsubseteq)$ that is *reflexive*: for all x,

$$x \sqsubseteq f.x \quad ,$$

idempotent: for all x,

$$f.x = f.(f.x) \quad ,$$

and monotonic.

According to the second definition, a closure operator is an endofunction, f, such that

$$x \sqsubseteq f.y \;\equiv\; f.x \sqsubseteq f.y$$

for all x and y.

An example of a closure operator is the function $(a\sqcup)$ where a is some element of A. The identity function is also (trivially) a closure operator.

(a) *Prove that the two definitions of closure operator given above are equivalent.*

Expressing the second definition in point-free form, f is a closure operator if and only if

$$g \,\dot{\sqsubseteq}\, f\bullet h \;\equiv\; f\bullet g \,\dot{\sqsubseteq}\, f\bullet h$$

for all monotonic functions g and h with range A and the same domain. You may find this the more useful definition in the final part of this exercise.

(b) Suppose that $f \in (A,\sqsubseteq)\leftarrow(A,\sqsubseteq)$ is a closure operator and suppose the pair of functions $G \in (A,\sqsubseteq)\leftarrow(B,\preceq)$ and $H \in (B,\preceq)\leftarrow(A,\sqsubseteq)$ forms a Galois connection (with G being the lower adjoint). *Show that* $H\bullet f\bullet G$ *is a closure operator.*

□

Exercise 64 Suppose the conditions on F and G stated in theorem 62 hold. Then the supremum operator in $F.\mathcal{B}$ defines an ordering $\preceq$ on the elements of $F.\mathcal{B}$ by the equation

$$x \preceq y \;\;\equiv\;\; x \sqcup_{F.\mathcal{B}} y = y \;\;.$$

Show that this ordering is the same as the poset ordering on elements of $\mathcal{A}$. *That is, show that for all* x *and* y *in* $F.\mathcal{B}$

$$x \sqsubseteq_{\mathcal{A}} y \;\;\equiv\;\; x \sqcup_{F.\mathcal{B}} y = y \;\;.$$

□

6 Fixed Points

6.1 Prefix Points

We begin our study of fixed points by introducing the notion of a "prefix" point. As an instance of where the notion commonly occurs, consider the following example from language theory. The set $L = \{n : 0 \leq n : a^n b c^n\}$ over the alphabet $\{a,b,c\}$ is sometimes specified in the following way.

The word b is in the set L.

If w is in the set L then so is the word awc.

Nothing else is in the set L.

Expressing the first two clauses in terms of set inclusion we see that L is required to satisfy the equation:

$$X{::}\ \{b\}\subseteq X \wedge \{a\}{\cdot}X{\cdot}\{c\}\subseteq X \quad ,$$

which is equivalent to the equation:

$$X{::}\ \{b\}\cup\{a\}{\cdot}X{\cdot}\{c\} \subseteq X \quad .$$

Now consider the function f mapping sets of words to sets of words defined by:

$$f.X = \{b\}\cup\{a\}{\cdot}X{\cdot}\{c\} \quad .$$

Then the requirement is that L be a so-called "prefix point" of f, i.e. L should satisfy the equation:

$$X{::}\ f.X\subseteq X \quad .$$

What about the third clause: "Nothing else is in the set L"? Note that this clause is necessary to specify L since, for example, the set of all words over the alphabet $\{a,b,c\}$ is a prefix point of the function f. One way to understand this clause is as the requirement that L be the *least* prefix point of the function f. Thus the complete specification of L is the equation

$$X{::}\ f.X\subseteq X \wedge \forall\langle Y{:}\ f.Y\subseteq Y{:}\ X\subseteq Y\rangle \quad .$$

Here, the first conjunct is the requirement that X be a prefix point of f, and the second conjunct that f be at most all prefix points Y of f.

In this section we define the notions of least prefix point and least fixed point in a general setting, and we establish the most basic properties of these notions. You are recommended to instantiate the properties we establish on the example just discussed as you read in order to confirm your understanding.

Suppose $\mathcal{A}=(A,\sqsubseteq)$ is a partially ordered set and suppose f is a monotonic endofunction on $\mathcal{A}$. Then a *prefix point* of f is an element x of the carrier set A such that $f.x\sqsubseteq x$. A *least prefix point* of f is a solution of the equation

$$x{::}\ f.x\sqsubseteq x \wedge \forall\langle y{:}\ f.y\sqsubseteq y{:}\ x\sqsubseteq y\rangle \quad .$$

A least prefix point of f is thus a prefix point of f that is smaller than all other prefix points of f. A *least fixed point* of f is a solution of the equation

(65) $$x{::}\ f.x=x \wedge \forall\langle y{:}\ f.y{=}y{:}\ x\sqsubseteq y\rangle \quad .$$

Rather than study *fixed* points of f we are going to study *prefix* points of f since the latter are mathematically more manageable than the former.

Exercise 66 Relations are often defined by just saying which pairs are in the relation. Implicitly the relation in question is defined to be the least solution of a certain equation. (Recall that relations are sets of pairs and thus ordered by set inclusion.)

An example is the following definition of the "at-most" relation on natural numbers (denoted as usual by the infix operator $\leq$). The "at-most" relation is the least relation satisfying, first, that for all natural numbers n,

$$0 \leq n$$

and, second, for all natural numbers m and n,

$$m+1 \leq n+1 \Leftarrow m \leq n \quad .$$

This defines the "at-most" relation as a least prefix point of a function from relations to relations. *What is this function?*

□

It is easy to see that a least prefix point of f is unique. Let x and x' be least prefix points of f. Then,

$$\begin{array}{ll}
 & x = x' \\
\equiv & \{ \quad \text{anti-symmetry} \quad \} \\
 & x \sqsubseteq x' \wedge x' \sqsubseteq x \\
\Leftarrow & \{ \quad \forall\langle y\text{: } f.y \sqsubseteq y\text{: } x \sqsubseteq y\rangle \text{ with the instantiation } y := x' \\
 & \quad\;\; \forall\langle y\text{: } f.y \sqsubseteq y\text{: } x' \sqsubseteq y\rangle \text{ with the instantiation } y := x \quad \} \\
 & f.x' \sqsubseteq x' \wedge f.x \sqsubseteq x \\
\equiv & \{ \quad x \text{ and } x' \text{ are prefix points of } f \quad \} \\
 & \mathsf{true} \quad .
\end{array}$$

A similar argument establishes that a least fixed point of f is unique.

Nothing is lost by studying least prefix points rather than least fixed points since, by the following calculation, a least prefix point of a monotonic function f is a fixed point of f.

$$\begin{array}{ll}
 & x = f.x \\
\equiv & \{ \quad \bullet \quad f.x \sqsubseteq x \quad \} \\
 & x \sqsubseteq f.x \\
\Leftarrow & \{ \quad \bullet \quad \forall\langle y\text{: } f.y \sqsubseteq y\text{: } x \sqsubseteq y\rangle \text{ , } y := f.x \quad \}
\end{array}$$

$$\begin{array}{ll} & f.(f.x) \sqsubseteq f.x \\ \Leftarrow & \{ \quad f \text{ is monotonic} \quad \} \\ & f.x \sqsubseteq x \\ \equiv & \{ \quad \bullet \quad f.x \sqsubseteq x \quad \} \\ & \mathsf{true} \quad . \end{array}$$

Since all fixed points are also prefix points it follows that a least prefix point of f is also a least fixed point of f.

Dual to the notion of least prefix point we can define the notion of "greatest postfix point": a *greatest postfix point* of endofunction $f \in (A\,,\,\sqsubseteq) \leftarrow (A\,,\,\sqsubseteq)$ is a least prefix point of the function $f \in (A\,,\,\sqsupseteq) \leftarrow (A\,,\,\sqsupseteq)$. Spelling this out in detail, a *postfix point* of f is an element x of the carrier set A such that $x \sqsubseteq f.x$ and a *greatest postfix point* of f is a solution of the equation

(67) $\quad x\!:: \; x \sqsubseteq f.x \wedge \forall\langle y\!: y \sqsubseteq f.y\!: y \sqsubseteq x\rangle \quad .$

A *greatest fixed point* of f is a solution of the equation

$$x\!:: \; f.x = x \wedge \forall\langle y\!: f.y = y\!: y \sqsubseteq x\rangle \quad .$$

Since a least prefix point is a least fixed point, we immediately have the dual result that a greatest postfix point is also a greatest fixed point. A simple but important corollary is that function f has a unique fixed point if and only if the least prefix point of f equals the greatest postfix point of f .

Let us summarise what has been learnt in this section. Introducing the notation μf for a solution of the equation (65) and νf for a solution of the equation (67) we have:

Theorem 68 (Least Prefix Point) Suppose $(A, \sqsubseteq)$ is an ordered set and the function f of type $(A, \sqsubseteq) \leftarrow (A, \sqsubseteq)$ is monotonic. Then f has at most one least prefix point, μf , characterised by the two properties:

(69) $\quad f.\mu f \sqsubseteq \mu f$

and, for all $x \in A$,

(70) $\quad \mu f \sqsubseteq x \;\Leftarrow\; f.x \sqsubseteq x \quad .$

Moreover, the least prefix point of f is a fixed point of f :

(71) $\quad f.\mu f = \mu f \quad .$

□

Note that least fixed points are characterised by the combination of (69) and (70) or (71) and (70). Because (69) is weaker than (71) it is usual to use the former characterisation when it is required to *establish* that a particular value is a least prefix point, and the latter characterisation when it is required to *exploit* the fact that a particular value is a least prefix point.

Theorem 72 (Greatest Postfix Point) Suppose $(A,\sqsubseteq)$ is an ordered set. Suppose, also, that the function f of type $(A,\sqsubseteq)\leftarrow(A,\sqsubseteq)$ is monotonic. Then f has at most one greatest postfix point, νf, characterised by the two properties:

(73) $\nu f \sqsubseteq f.\nu f$

and, for all $x{\in}A$,

(74) $x \sqsubseteq \nu f \ \Leftarrow\ x \sqsubseteq f.x$.

Moreover, the greatest postfix point of f is a fixed point of f:

(75) $f.\nu f = \nu f$.

□

Use of the rules (70) or (74) corresponds to using induction. We therefore refer to their use in calculations as *fixed point induction*. (Strictly we should say *least* fixed point induction or *greatest* fixed point induction but it will invariably be clear which of the two rules is meant.) Rules (71) and (75) are called *computation* rules because they are often used in programs as left-right rewrite rules, repeated application of which is used to compute a fixed point.

Both induction rules can of course be written as an implication rather than a follows-from. For example, (70) can be written as

$$f.x \sqsubseteq x \ \Rightarrow\ \mu f \sqsubseteq x \ .$$

This is the way many would write the rule. We deviate from this practice because in almost all our calculations we will apply the rule from left to right in the form given in (70). This is independent of whether our goal is to *verify* a statement of the form $\mu f \sqsubseteq x$ or, given x, to *construct* an f satisfying $\mu f \sqsubseteq x$, or, given f, to *construct* an x satisfying $\mu f \sqsubseteq x$. The benefit of proceeding in this way is the replacement of complification steps by simplification steps.

Exercise 76 *Show that* $\mu f \sqsubseteq \nu f$.

□

6.2 A First Example

In this section we present a simple example of the Least Prefix Point theorem (theorem 68). The example illustrates why (70) really is an induction rule. It also illustrates the use of the computation rule (71) in a simple example of algorithm design.

Let L be the least solution of the equation

$$X{::}\ \{a\}\cup\{b\}\cdot X\cdot X \subseteq X \quad .$$

This corresponds to a grammar having two productions $X ::= a$ and $X ::= bXX$. We first use (70) to prove that, for all words w in L, the number of a's in w is one more than the number of b's in w.

Let M be the set of all words w such that the number of a's in w is one more than the number of b's in w. Let $\#_a w$ denote the number of a's in w, and $\#_b w$ denote the number of b's in w. We have to show that $L\subseteq M$. We use fixed point induction:

$L\subseteq M$

$\Leftarrow$ { by definition $L=\mu f$ where $f=\langle X{::}\ \{a\}\cup\{b\}\cdot X\cdot X\rangle$, induction: (70) }

$\{a\}\cup\{b\}\cdot M\cdot M \subseteq M$

$\equiv$ { set theory, definition of concatenation }

$a\in M \wedge \forall\langle x,y{:}\ x\in M\wedge y\in M{:}\ bxy \in M\rangle$

$\equiv$ { definition of M }

$\#_a a = \#_b a + 1$

$\wedge\ \forall\langle x,y{:}\ \#_a x = \#_b x + 1 \wedge \#_a y = \#_b y + 1{:}\ \#_a(bxy) = \#_b(bxy) + 1\rangle$

$\equiv$ { definition of $\#_a$ and $\#_b$ }

true

$\wedge\ \forall\langle x,y{:}\ \#_a x = \#_b x + 1 \wedge \#_a y = \#_b y + 1{:}\ \#_a x + \#_a y = 1 + \#_b x + \#_b y + 1\rangle$

$\equiv$ { arithmetic }

true .

Note that this proof is not a proof by induction on the length of words in L. Introducing the length of words in L into the proof is spurious and only obscures the properties that are truly vital to the proof.

Now let us illustrate the use of the computation rule. The problem we consider is how to write a simple loop that determines whether a given word W is in the language

L. At the same time we illustrate one of the most basic elements of the mathematics of program construction, namely the use of loop invariants.

For the purposes of the discussion we need two functions on words, fst and rest. The fst function returns the first symbol of a word of length at least one, and the rest function returns the remainder of the word. For example, $\mathrm{fst}.(aba) = a$ and $\mathrm{rest}.(aba) = ba$. We also use ε to denote the empty word (the word of length zero).

The problem is to determine the truth or falsity of $W{\in}L$. As L is a *fixed* point of its defining equation, we can expand this requirement as follows:

$$
\begin{array}{ll}
 & W \in L \\
\equiv & \{ \quad \text{by (71), } L = \{a\} \cup \{b\}{\cdot}L{\cdot}L \quad \} \\
 & W \in \{a\} \cup \{b\}{\cdot}L{\cdot}L \\
\equiv & \{ \quad \text{set calculus} \quad \} \\
 & W = a \lor (\mathrm{fst}.W = b \land \mathrm{rest}.W \in L{\cdot}L) \quad .
\end{array}
$$

The two tests $W = a$ and $\mathrm{fst}.W = b$ can, of course, easily be implemented. This then leaves us with the task of implementing the test $\mathrm{rest}.W \in L{\cdot}L$. Were we to repeat the same small calculation beginning with the latter then we would end up with the task of implementing the test $\mathrm{rest}.(\mathrm{rest}.W) \in L{\cdot}L{\cdot}L$. This clearly suggests a generalisation of the original problem, namely, given natural number n and word w determine the truth or falsity of the statement $w{\in}L^n$ where, by definition, $L^0 = \{\varepsilon\}$ and $L^{n+1} = L{\cdot}L^n$.

With this generalised problem we begin our calculation again. First, by definition of L^0,

(77) $\quad w{\in}L^0 \equiv w = \varepsilon \quad .$

Second,

$$
\begin{array}{ll}
 & w{\in}L^{n+1} \\
\equiv & \{ \quad L^{n+1} = L{\cdot}L^n \text{ and, by (71), } L = \{a\} \cup \{b\}{\cdot}L{\cdot}L \quad \} \\
 & w \in (\{a\} \cup \{b\}{\cdot}L{\cdot}L){\cdot}L^n \\
\equiv & \{ \quad \text{concatenation distributes through union,} \\
 & \quad\;\; \text{concatenation is associative} \quad \} \\
 & w \in \{a\}{\cdot}L^n \cup \{b\}{\cdot}L{\cdot}L{\cdot}L^n \\
\equiv & \{ \quad \text{set calculus} \quad \} \\
 & (\mathrm{fst}.w = a \land \mathrm{rest}.w \in L^n) \lor (\mathrm{fst}.w = b \land \mathrm{rest}.w \in L^{n+2}) \quad .
\end{array}
$$

This calculation provides enough information to construct the required loop. The loop uses two variables, a word w and a natural number k, satisfying the loop invariant

$$W \in L \;\equiv\; w \in L^k \quad .$$

This invariant is established by the assignment:

$$w, k \;:=\; W, 1 \quad .$$

Also, by the above calculation, it is maintained when $k = n+1$ for some number n by the assignment

```
if  fst.$w = a$ → $w, k$ := rest.$w, k-1$
□   fst.$w = b$ → $w, k$ := rest.$w, k+1$
fi  .
```

(Here we make the implicit assumption that the given word W is a string of a's and b's and contains no other characters, thus avoiding stipulating the semantics of the statement when neither of the two tests evaluates to true.) This statement also makes progress by reducing the length of w by 1 at each iteration. Finally, if we impose the termination condition $w = \varepsilon \vee k = 0$, then the conjunction of the termination condition and the invariant

$$(w = \varepsilon \vee k = 0) \wedge (W \in L \;\equiv\; w \in L^k)$$

implies, using (77) and the calculation immediately following it, that

$$W \in L \equiv w = \varepsilon \wedge k = 0 \quad .$$

This then is the complete program:

```
    $w, k$ := $W, 1$
    { Invariant: $W \in L \equiv w \in L^k$.
      Bound function: length of $w$ }
;   do $\neg(w = \varepsilon \vee k = 0)$ →   if  fst.$w = a$ → $w, k$ := rest.$w, k-1$
                                    □   fst.$w = b$ → $w, k$ := rest.$w, k+1$
                                    fi
    od
    { $(w = \varepsilon \vee k = 0) \wedge (W \in L \equiv w \in L^k)$ }
    { $W \in L \equiv w = \varepsilon \wedge k = 0$ }
```

6.3 Kleene Algebra

The purpose of this section is to provide some practice in the use of the characterising properties of least fixed points, in particular the induction rule. For this purpose we consider an algebraic structure called a "Kleene algebra" by Kozen [Koz94].

A Kleene algebra has two constants 0 and 1, two binary operators $+$ and $\cdot$, and a unary operator *. These operators satisfy a number of axioms which we shall give shortly. The name "Kleene algebra" is a tribute to S. C. Kleene [Kle56] who postulated an algebraic structure as a basis for studying the behaviour of finite state machines. He called it the "algebra of regular events", which nowadays is most commonly abbreviated to "regular algebra".

Regular algebra is central to the mathematics of program construction because the three operators capture the essential properties of the three main ingredients of programming languages, choice ($+$), sequencing ($\cdot$) and iteration (*). The axioms of a Kleene algebra are not sufficient to capture all the properties of choice, sequencing and iteration in programming languages and we shall need to add to the axioms later on. Any Kleene algebra is thus also a regular algebra but it is not necessarily the case that any regular algebra is a Kleene algebra. See sections 7.3 and 7.4 for further discussion.

6.3.1 The Axioms

It is convenient to discuss the axioms of a Kleene algebra in two stages, first the axiomatisation of the $+$ and $\cdot$ operators, and then the axiomatisation of the * operator. In presenting the axioms we use variables a, b, x, y and z to range over the carrier set of the algebra. These variables are universally quantified in the usual way.

The notation "$+$" and "$\cdot$" is chosen to suggest a connection with addition and multiplication in ordinary arithmetic. Indeed, the $+$ operator of a Kleene algebra is required to be associative and symmetric and to have 0 as unit element:

$$(x+y)+z = x+(y+z) \quad ,$$

$$x+y = y+x \quad ,$$

and $$x+0 = x = 0+x \quad .$$

Also, just as in arithmetic, multiplication is required to be associative, to distribute over addition, and to have 1 as unit and 0 as zero element:

$$x\cdot(y\cdot z) = (x\cdot y)\cdot z \quad ,$$

$$x\cdot(y+z) = (x\cdot y)+(x\cdot z) \quad ,$$

$$(y+z)\cdot x = (y\cdot x)+(z\cdot x) \quad ,$$

$$x{\cdot}0 \;=\; 0 \;=\; 0{\cdot}x \quad ,$$

and $$1{\cdot}x = x = x{\cdot}1 \quad .$$

Use of these rules in calculations will be referred to simply as "arithmetic".

The addition and multiplication operators of a Kleene algebra deviate from the operators of normal arithmetic in two respects. First, multiplication is *not* assumed to be commutative. Second, addition is assumed to be idempotent. That is,

$$x{+}x = x$$

for all x. (Equivalently, $1{+}1 = 1$.)

Because addition is idempotent, associative and symmetric, the relation $\leq$ defined by

$$x{\leq}y \;\equiv\; x{+}y = y$$

is reflexive, transitive and anti-symmetric. In short, it is a partial ordering relation. Moreover, because 0 is the unit of addition, it is the least element in the ordering:

$$0{\leq}x$$

for all x. Finally, addition and multiplication are both monotonic with respect to the $\leq$ ordering.

Exercise 78 *Verify all the claims made in the last paragraph. Prove also that* $+$ *is the binary supremum operator.* That is,

$$x{+}y{\leq}z \equiv x{\leq}z{\wedge}y{\leq}z \quad .$$

□

The * operator will be the focus of our attention. The axioms for * require that $a^*\cdot b$ be the least prefix point of the (monotonic) function mapping x to $b + a{\cdot}x$ and that $b \cdot a^*$ be the least prefix point of the (monotonic) function mapping x to $b + x{\cdot}a$. Formally, $a^* \cdot b$ is a prefix point of the function mapping x to $b + a{\cdot}x$:

(79) $$b + a{\cdot}(a^*{\cdot}b) \leq a^*{\cdot}b \quad ,$$

and is the least among all such prefix points:

(80) $$a^*{\cdot}b \leq x \Leftarrow b + a{\cdot}x \leq x \quad .$$

That is,

(81) $$a^*{\cdot}b = \mu\langle x{::}\; b + a{\cdot}x\rangle \quad .$$

Similarly, $b \cdot a^*$ is a prefix point of the function mapping x to $b+x{\cdot}a$:

(82) $\quad b+(b{\cdot}a^*){\cdot}a \le b{\cdot}a^*$,

and is the least among all such prefix points:

(83) $\quad b{\cdot}a^* \le x \Leftarrow b+x{\cdot}a \le x$.

That is,

(84) $\quad b{\cdot}a^* = \mu\langle x{::}\ b+x{\cdot}a\rangle$.

The axioms (79) and (82) are thus instances of (69) and the axioms (80) and (83) are instances of (70), the two properties characterising least prefix points. An immediate corollary is that $a^*{\cdot}b$ and $b{\cdot}a^*$ are *fixed* points of the relevant functions:

(85) $\quad b+a{\cdot}(a^*{\cdot}b) = a^*{\cdot}b$,

(86) $\quad b+(b{\cdot}a^*){\cdot}a = b{\cdot}a^*$.

This concludes the axiomatisation of a Kleene algebra. There are several interpretations of the operators in a Kleene algebra that fulfill the axioms. Table 6.3.1 summarises a few. The first interpretation in the table, in which the carrier consists of sets of words over a given, fixed alphabet, is the original application of Kleene algebra. Here the addition operation is set union, the multiplication operation is concatenation of sets of words, 0 is the empty set, 1 is the set whose sole element is the empty word, and a^* is the set of words formed by concatenating together an arbitrary number of words in the set a.

The second interpretation is the one we will return to most often in these lecture notes. In this interpretation the carrier set of the algebra is the set of binary relations over some state space, the addition operator is set union, multiplication is the composition operator on relations, 0 is the empty relation, 1 is the identity relation, and the iteration operator is the reflexive, transitive closure operator on relations.

The remaining three interpretations all concern path-finding problems; in each case the iteration operator (*) is uninteresting. These interpretations only become interesting when one considers graphs with edge labels drawn from these primitive Kleene algebras, the crucial theorem being that it is possible to define a Kleene algebra of graphs given that the edge labels are elements of a Kleene algebra [BC75, Koz94].

Table 1: Kleene algebras

	carrier	$+$	$\cdot$	0	1	a^*	$\leq$
Languages	sets of words	$\cup$	$\cdot$	ϕ	$\{\varepsilon\}$	a^*	$\subseteq$
Programming	binary relations	$\cup$	$\circ$	ϕ	id	a^*	$\subseteq$
Reachability	booleans	$\vee$	$\wedge$	false	true	true	$\Rightarrow$
Shortest paths	nonnegative reals	max	$+$	∞	0	0	$\geq$
Bottlenecks	nonnegative reals	max	min	0	∞	∞	$\leq$

6.3.2 Reflexive, Transitive Closure

It is common to call a^* *the reflexive*, *transitive closure* of a. This is because a^* is reflexive, in the sense that

$$1 \leq a^* \quad ,$$

transitive, in the sense that

$$a^* = a^* \cdot a^* \quad ,$$

and * is a closure operator, i.e.

$$a \leq b^* \equiv a^* \leq b^* \quad .$$

In this section we shall verify, from the axioms given above, that a^* does indeed have these and other related properties. The main tool will be fixed point induction, i.e. axioms (80) and (83).

The reflexivity of a^* is immediate from (85) or (86). Instantiating b to 1 in both these rules and using $1 \cdot x = x = x \cdot 1$ we get

$$1 + a \cdot a^* = a^* = 1 + a^* \cdot a \quad .$$

Thus, since $+$ is the binary supremum operator, $1 \leq a^*$. It then follows that $a \leq a^*$ since

$$a^* = 1 + a \cdot a^* = 1 + a \cdot (1 + a \cdot a^*) = 1 + a \cdot 1 + a \cdot (a \cdot a^*) = 1 + a + a \cdot a \cdot a^* \quad .$$

Now for the transitivity of a^* we use a ping-pong argument:

$$
\begin{array}{ll}
 & a^* = a^*\cdot a^* \\
\equiv & \{ \quad \text{antisymmetry} \quad \} \\
 & a^* \le a^*\cdot a^* \;\wedge\; a^*\cdot a^* \le a^* \quad .
\end{array}
$$

The first inequality does not involve induction:

$$
\begin{array}{ll}
 & a^* \le a^*\cdot a^* \\
\Leftarrow & \{ \quad \text{transitivity} \quad \} \\
 & a^* \le 1\cdot a^* \le a^*\cdot a^* \\
\equiv & \{ \quad \text{arithmetic for the first inequality;} \\
 & \quad\;\; 1 \le a^* \text{ and multiplication is monotonic for the second} \quad \} \\
 & \mathsf{true} \quad .
\end{array}
$$

The second inequality is where we need to use induction:

$$
\begin{array}{ll}
 & a^*\cdot a^* \le a^* \\
\Leftarrow & \{ \quad (80) \text{ with } a,b,x \;:=\; a\,,a^*\,,a^* \quad \} \\
 & a^* + a\cdot a^* \;\le\; a^* \\
\equiv & \{ \quad + \text{ is the binary supremum operator} \quad \} \\
 & a^* \le a^* \;\wedge\; a\cdot a^* \le a^* \\
\equiv & \{ \quad \text{reflexivity of } \le\,;\;\; a^* \;=\; 1 + a\cdot a^* \quad \} \\
 & \mathsf{true} \quad .
\end{array}
$$

Thus we have established that $a^* = a^*\cdot a^*$.

We establish that $*$ is a closure operator by the following ping-pong argument.

$$
\begin{array}{ll}
 & a^* \le b^* \\
\Leftarrow & \{ \quad (80) \text{ with } a,b,x \;:=\; a\,,a\,,b^* \quad \} \\
 & 1 + a\cdot b^* \;\le\; b^* \\
\equiv & \{ \quad + \text{ is the binary supremum operator} \quad \} \\
 & 1 \le b^* \;\wedge\; a\cdot b^* \le b^* \\
\Leftarrow & \{ \quad b^* \;=\; 1 + b\cdot b^*\,,\;\; b^* = b^*\cdot b^*\,,\;\; \cdot \text{ is monotonic} \quad \} \\
 & a \le b^* \\
\Leftarrow & \{ \quad a \le a^* \quad \} \\
 & a^* \le b^* \quad .
\end{array}
$$

Exercise 87 *Prove the following properties*:

(a) $a \cdot b^* \leq c^* \cdot a \Leftarrow a{\cdot}b \leq c{\cdot}a$

(b) $c^* \cdot a \leq a \cdot b^* \Leftarrow c{\cdot}a \leq a{\cdot}b$

(c) $a \cdot (b{\cdot}a)^* = (a{\cdot}b)^* \cdot a$

(d) $(a{+}b)^* = b^* \cdot (a \cdot b^*)^* = (b^* \cdot a)^* \cdot b^*$

(e) $(a^*)^* = a^*$

Properties (a) and (b) are called *leapfrog* rules (because a "leapfrogs" from one side of a star term to the other). Both have the immediate corollary that * is monotonic (by taking a to be 1). Properties (c) and (d) are called the *mirror* rule and *star decomposition* rule, respectively. Property (e) states that * is idempotent.

□

Exercise 88 The goal of this exercise is to show that a^* is the least reflexive and transitive element of the algebra that is at least a.

Let a^+ denote $a \cdot a^*$. We call a^+ the *transitive closure* of a. *Prove the following properties of* a^+.

(a) $a \leq a^+$

(b) $a^+ = a \cdot a^* = a^* \cdot a$

(c) $a^+ \cdot a^+ \leq a^+$

(d) $a^+ \leq x \Leftarrow a + x{\cdot}x \leq x$

Note that the combination of (a), (c) and (d) expresses the fact that a^+ is the least prefix point, and hence the least fixed point, of the function $\langle x{::}\ a + x{\cdot}x \rangle$. That is, a^+ is the least transitive element that is at least a.

(e) *Show that* a^* *is the least fixed point of the function* $\langle x{::}\ 1 + a + x{\cdot}x \rangle$.

□

7 Fixed Point Calculus

The least fixed point of a monotonic function is, as we have seen in theorem 68, characterised by two properties. It is a fixed point, and it is least among all prefix points of the functions. This gives us two calculational rules for reasoning about the least fixed point μf of monotonic function f: the *computation rule*

$$\mu f = f.\mu f$$

and the *induction rule*: for all x,

$$\mu f \sqsubseteq x \Leftarrow f.x \sqsubseteq x \quad .$$

In principle these are the only rules one needs to know about least fixed points. Every calculation can be reduced to one involving just these two rules (together with the rules pertaining to the specific problem in hand). This indeed is commonly what happens. Many postulates are verified by induction. But this is often not the most effective way of reasoning about fixed points.

The problem is that this basic characterisation of least fixed points typically invites proof by mutual inclusion. That is, given two expressions, E and F, involving fixed points (or prefix points) the obvious strategy to determine conditions under which they are equal is to determine conditions under which the two inclusions $E \sqsubseteq F$ and $F \sqsubseteq E$ hold. This can lead to long and clumsy calculations. To avoid this we state a number of equational properties of least fixed points. Their proofs, which are mostly straightforward, are left as exercises. We conclude the section with the mutual recursion theorem, which expresses the solution of mutually recursive fixed point equations in terms of the successive solution of individual equations. The proof, which we do include, is a fine illustration of the earlier properties.

Unless otherwise stated, we assume throughout this section that f and g are monotonic endofunctions on the complete lattice $\mathcal{A} = (A , \sqsubseteq)$. (The assumption of completeness can be avoided but is convenient and practical.) The set of such functions also forms a complete lattice under the *pointwise ordering* $\dot{\sqsubseteq}$ defined by

$$f \dot{\sqsubseteq} g \;\equiv\; \forall\langle x{::}\; f.x \sqsubseteq g.x\rangle \quad .$$

The set of prefix points of f is denoted by $\mathsf{Pre}.f$ and its set of postfix points is denoted by $\mathsf{Post}.f$. The set $\mathsf{Pre}.f$ is also a complete lattice, as is $\mathsf{Post}.f$.

7.1 Basic Rules

The most basic fixed point rule is that the least-fixed-point operator is monotonic:

(89) $\quad \mu f \sqsubseteq \mu g \Leftarrow f \dot{\sqsubseteq} g \quad .$

The second rule is called the *rolling rule*:

(90) $g.\mu(f\bullet g) = \mu(g\bullet f)$

(where $f \in (A, \sqsubseteq) \leftarrow (B, \preceq)$ and $g \in (B, \preceq) \leftarrow (A, \sqsubseteq)$ for some posets $(A, \sqsubseteq)$ and $(B, \preceq)$). Note that the rolling rule subsumes the rule that a least prefix point is also a fixed point. Just instantiate f to the identity function. The proof we gave of this fact in the previous section can be adapted to give a proof of the rolling rule.

An immediate corollary of the rolling role is that, for a monotonic endofunction f, $\mu(f^2) = f.\mu(f^2)$ where f^2 denotes $f\bullet f$. This suggests that $\mu(f^2)$ and μf are equal. This is indeed the case, and is called the *square rule*:

(91) $\mu f = \mu(f^2)$.

Consider now a monotonic, binary function $\oplus$ on the poset $\mathcal{A}$. Then, for each x, the function $(x\oplus)$ (which maps y to $x\oplus y$) is monotonic and so has a least fixed point $\mu(x\oplus)$. Abstracting from x, the function $\langle x{::}\ \mu(x\oplus)\rangle$ is a monotonic endofunction on $\mathcal{A}$ and its least fixed point is $\mu\langle x{::}\ \mu\langle y{::}\ x\oplus y\rangle\rangle$. The *diagonal rule* removes the nesting in this expression:

(92) $\mu\langle x{::}\, x\oplus x\rangle \;=\; \mu\langle x{::}\ \mu\langle y{::}\ x\oplus y\rangle\rangle$.

The diagonal rule is a very important rule because it is the basis of methods for finding fixed points step by step. Suppose one wants to solve a fixed point equation $x{::}\ x = x\oplus x$ where "$x\oplus x$" is a large expression involving several occurrences of the identifier x. The diagonal rule allows one to eliminate the x's one-by-one. This is what happens when one solves a number of equations defining a set of values by mutual recursion (sometimes called a set of "simultaneous equations"). Such a set of equations can be viewed as one equation of the form $\underline{x} = f.\underline{x}$ where $\underline{x}$ is a vector of values. Each occurrence of an element x_i in the set of equations is in fact an occurrence of $\Pi_i.\underline{x}$, where Π_i denotes the function that "projects" the vector $\underline{x}$ onto its ith component. It is thus an occurrence of "$\underline{x}$" itself and the standard practice is to eliminate such occurrences individually. Mutual recursion is discussed further in section 7.5.

Exercise 93 *For arbitrary elements* a *and* b *of a Kleene algebra , prove the following properties. Use only equational reasoning. (That is, do not prove any of the properties by mutual inclusion.)* (Hint: Take advantage of exercises 87 and 88.)

(a) $\mu\langle X{::}\ a\cdot X^*\rangle \;=\; a^+$,

(b) $\mu\langle X{::}\ (a+X)^*\rangle \;=\; \mu\langle X{::}\ (a\cdot X)^*\rangle \;=\; \mu\langle X{::}\ a+X^*\rangle \;=\; a^*$,

(c) $\mu\langle X{::}\ a+X\cdot b\cdot X\rangle \;=\; a\cdot(b\cdot a)^*$,

(d) $\mu\langle X{::}\ 1+a{\cdot}X{\cdot}b{\cdot}X+b{\cdot}X{\cdot}a{\cdot}X\rangle \;=\; \mu\langle X{::}\ 1+a{\cdot}X{\cdot}b+b{\cdot}X{\cdot}a+X{\cdot}X\rangle$.

□

Exercise 94 Verify each of the rules given in this subsection.

□

7.2 Fusion

This section is about perhaps the most important fixed point rule of all, the *fusion* rule. In words, the fusion rule provides a condition under which the application of a function to a least fixed point can be expressed as a least fixed point. The rule is important because many computational problems are initially specified as such a function application, but are solved by computing the (least or greatest) solution to an appropriate fixed point equation.

Several examples arise as path-finding problems. Given a graph with "weighted" edges (formally, a function from the edges to the real numbers), the *length* of a path is defined to be the *sum* of the edge weights, and the *width* of a path is defined to be the *minimum* of the edge weights. The shortest *distance* between two nodes is the *minimum*, over all paths between the nodes, of the length of the path whilst the largest *gap* between two nodes is the *maximum*, over all paths between the nodes, of the width of the path. Thus both the shortest distance and the largest gap are defined as functions applied to the set of all paths between two given nodes; in the case of shortest distance, the function is "minimize the length", and, in the case of the largest gap, the function "maximize the width". However, both these problems are typically solved not by first enumerating all paths (an impossible task in the general case because there are infinitely many of them) but by solving directly recursive equations defining the shortest distance and largest gap functions. The formal justification for this process relies on the fusion theorem below.

The most powerful of the two rules characterising least prefix points is the induction rule. Its power is, however, somewhat limited because it only allows one to calculate with orderings in which the μ operator is the *principal* operator on the *lower* side of the ordering (i.e. orderings of the form $\mu f \sqsubseteq \cdots$). Formally the fusion rule overcomes this restriction on the induction rule by combining the calculation properties of Galois connections with those of fixed points.

Theorem 95 (μ-fusion) Suppose $f \in A \leftarrow B$ is the lower adjoint in a Galois connection between the posets $(A\,,\,\sqsubseteq)$ and $(B\,,\,\preceq)$. Suppose also that $g \in (B\,,\,\preceq) \leftarrow (B\,,\,\preceq)$ and $h \in (A\,,\,\sqsubseteq) \leftarrow (A\,,\,\sqsubseteq)$ are monotonic functions. Then

(a) $\quad f.\mu g \sqsubseteq \mu h \Leftarrow f\bullet g \mathrel{\dot{\sqsubseteq}} h\bullet f$,

(b) $\quad f.\mu g = \mu h \Leftarrow f\bullet g = h\bullet f$.

Indeed, if the condition

$$f\bullet g = h\bullet f$$

holds, f is the lower adjoint in a Galois connection between the posets $(\mathsf{Pre}.h\,,\,\sqsubseteq)$ and $(\mathsf{Pre}.g\,,\,\preceq)$.

□

We call this theorem μ-“fusion” because it states when application of function, f, can be “fused” with a fixed point, μg, to form a fixed point, μh. (The rule is also used, of course, to “defuse” a fixed point into the application of a function to another fixed point.) Another reason for giving the rule this name is because it is the basis of so-called “loop fusion” techniques in programming: the combination of two loops, one executed after the other, into a single loop. The rule is also called the *transfer* lemma because it states when function f “transfers” the computation of one fixed point (μg) into the computation of another fixed point (μh).

Exercise 96 *Prove the fusion rule.* (Hint: first show that the function $k \in B \leftarrow A$ maps prefix points of h into prefix points of g if $g\bullet k \mathrel{\dot{\sqsubseteq}} k\bullet h$. Use this to show that f is the lower adjoint in a Galois connection between the posets $(\mathsf{Pre}.h\,,\,\sqsubseteq)$ and $(\mathsf{Pre}.g\,,\,\preceq)$.)

□

Note that in order to apply μ-fusion we do not need to *know* the upper adjoint of function f, we only need to know that it exists. The fundamental theorem of Galois connections is the most commonly used tool to establish the existence of an upper adjoint.

As an aid to memorising the μ-fusion theorems note that the order of f, g, $\sqsubseteq$ or $=$, and h is the same in the consequent and the antecedent, be it that in the antecedent the lower adjoint occurs twice, in different argument positions of $\bullet$.

The conclusion of μ-fusion — $\mu h = f.\mu g$ — involves *two* premises, that f be a lower adjoint, and that $f\bullet g = h\bullet f$. The rule is nevertheless very versatile since being a lower adjoint is far from being uncommon, and many algebraic properties take the form $f\bullet g = h\bullet f$ for some functions f, g and h. The next lemma also combines fixed points with Galois connections. We call it the *exchange rule* because it is used to exchange one lower adjoint for another. Its proof is left as an exercise. (Hint: for part (a) use induction followed by the rolling rule.)

Lemma 97 (Exchange Rule) Suppose f has type $(A\,,\sqsubseteq)\leftarrow(B\,,\preceq)$, and g and h both have type $(B\,,\preceq)\leftarrow(A\,,\sqsubseteq)$. Then, if g is a lower adjoint in a Galois connection,

(a) $\quad \mu(g\bullet f)\sqsubseteq\mu(h\bullet f)\;\wedge\;\mu(f\bullet g)\sqsubseteq\mu(f\bullet h)\;\Leftarrow\; g\bullet f\bullet h\,\dot{\sqsubseteq}\,h\bullet f\bullet g$.

Furthermore, if both g and h are lower adjoints, then

(b) $\quad \mu(g\bullet f)=\mu(h\bullet f)\;\wedge\;\mu(f\bullet g)=\mu(f\bullet h)\;\Leftarrow\; g\bullet f\bullet h=h\bullet f\bullet g$.

□

The two conclusions of (b), $\mu(g\bullet f)=\mu(h\bullet f)$ and $\mu(f\bullet g)=\mu(f\bullet h)$, say that g and h can be exchanged within a μ term in which they are composed after or before an arbitrary function f.

Exercise 98 *Prove the exchange rule.*

The two conjuncts in the consequent of 97(b) can be combined into one equation. The resulting lemma states when lower adjoint g can be replaced by lower adjoint h in an arbitrary context. The lemma has four variables rather than three. *Find the lemma and prove its validity.*

□

7.2.1 Applications

We mentioned path-finding problems earlier as an example of the use of fusion. Space does not allow us to expand this in detail. Instead, this section illustrates the use of the rule by some examples involving languages.

Shortest Word As explained earlier, a context-free grammar defines a function from languages to languages. The language defined by the grammar is the least fixed point of this function. There is a number of computations that we may wish to perform on the grammar. These include testing whether the language generated is nonempty or not, and determining the length of a shortest word in the grammar. Let us consider the latter problem. (In fact, the problem is intimately related to the shortest path problem on graphs.) For concreteness we will demonstrate how to solve the problem using the grammar with productions

$$S \quad ::= \quad aS \;\mid\; SS \;\mid\; \varepsilon \;.$$

The language defined by the grammar is thus

$$\mu\langle X::\{a\}\cdot X\cup X\cdot X\cup\{\varepsilon\}\rangle$$

(Of course the problem is very easily solved for a specific example as simple as this one. Try to imagine however that we are dealing with a grammar with a large number of nonterminals and a large number of productions.)

Given a language L defined in this way, the general problem is to find $\#L$ where the function $\#$ is defined by extending the length function on words to a shortest length function on languages. Specifically, letting $\mathsf{length}.w$ denote the length of word w (the number of symbols in the word),

$$\#L \;=\; \Downarrow\langle w\colon w{\in}L\colon \mathsf{length}.w\rangle$$

(where $\Downarrow$ denotes the minimum operator — the minimum over an empty set is defined to be infinity). Now, because $\#$ is the infimum of the length function it is the lower adjoint in a Galois connection. Indeed,

$$\#L \geq k \;\equiv\; L \subseteq \Sigma^{\geq k}$$

where $\Sigma^{\geq k}$ is the set of all words (in the alphabet Σ) whose length is at least k. Noting the direction of the ordering on the left-hand side of this Galois connection, we get that

$$\#(\mu_{\subseteq} f) = \mu_{\geq} g \quad\Leftarrow\quad \#\bullet f \;=\; g\bullet\# \quad .$$

Applying this to our example grammar, we fill in f and calculate g:

$$\begin{array}{ll}
 & \#\bullet\langle X\!::\{a\}{\cdot}X\cup X{\cdot}X\cup\{\varepsilon\}\rangle \;=\; g\bullet\# \\
\equiv & \qquad\{\qquad \text{definition of composition} \quad\} \\
 & \forall\langle X\!::\ \#(\{a\}{\cdot}X\cup X{\cdot}X\cup\{\varepsilon\}) \;=\; g.(\#X)\rangle \\
\equiv & \qquad\{\qquad \#\ \text{is a lower adjoint and so distributes over}\ \cup\,, \\
 & \qquad\qquad \text{definition of}\ \# \quad\} \\
 & \forall\langle X\!::\ \#(\{a\}{\cdot}X)\downarrow\#(X{\cdot}X)\downarrow\#\{\varepsilon\} \;=\; g.(\#X)\rangle \\
\equiv & \qquad\{\qquad \#(Y{\cdot}Z) = \#Y+\#Z\,,\ \#\{a\}{=}1\,,\ \#\{\varepsilon\}{=}0 \quad\} \\
 & (1+\#X){\downarrow}(\#X+\#X){\downarrow}0 \;=\; g.(\#X) \\
\Leftarrow & \qquad\{\qquad \text{instantiation} \quad\} \\
 & \forall\langle k\!::\ (1{+}k){\downarrow}(k{+}k){\downarrow}0 \;=\; g.k\rangle \quad .
\end{array}$$

In this way we have determined that, for the language L defined by the above grammar,

$$\#L \;=\; \mu_{\geq}\langle k\!::(1{+}k){\downarrow}(k{+}k){\downarrow}0\rangle \quad .$$

That is, the length of a shortest word in the language defined by the grammar with productions

$$S \quad ::= \quad aS \;\mid\; SS \;\mid\; \varepsilon$$

is the greatest value k satisfying

$$k \;=\; (1{+}k) \downarrow (k{+}k) \;\downarrow\; 0 \;\; .$$

Note how the structure of the equation for k closely follows the structure of the original grammar — the alternate operator has been replaced by minimum, and concatenation of languages has been replaced by addition. What the example illustrates is that the length of the shortest word in the language generated by a context-free grammar can always be computed by determining the *greatest* fixed point (in the usual $\leq$ ordering on numbers — that is, the least fixed point if numbers are ordered by $\geq$) of an equation (or system of simultaneuous equations) obtained by replacing all occurrences of alternation in the grammar by minimum and all occurrences of concatenation by addition. Moreover, the key to the proof of this theorem is the fusion rule.

Membership of a language To illustrate the use of fusion yet further we consider two examples. The first is an example of how the fusion theorem is applied; the second illustrates how the fusion theorem need not be *directly* applicable. We discuss briefly how the second example is generalised in such a way that the fusion theorem does become applicable.

Both examples are concerned with membership of a set. So, let us consider an arbitrary set $\mathcal{U}$. For each x in $\mathcal{U}$ the predicate $(x{\in})$ maps a subset P of $\mathcal{U}$ to the boolean value true if x is an element of P and otherwise to false. The predicate $(x{\in})$ preserves set union. That is, for all bags $\mathcal{S}$ of subsets of $\mathcal{U}$,

$$x \in \cup\mathcal{S} \equiv \exists\langle P{:}\, P{\in}\mathcal{S}{:}\, x{\in}P\rangle \;\; .$$

According to the fundamental theorem, the predicate $(x{\in})$ thus has an upper adjoint. Indeed, we have, for all booleans b,

$$x{\in}S \Rightarrow b \quad\equiv\quad S \subseteq \text{if } b \rightarrow \mathcal{U} \;\square\; \neg b \rightarrow \mathcal{U}\backslash\{x\} \text{ fi} \;\; .$$

Now suppose f is a monotonic function on sets. Let μf denote its least fixed point. The fact that $(x{\in})$ is a lower adjoint means that we may be able to apply the fusion theorem to reduce a test for membership in μf to solving a recursive equation. Specifically

$$(x \in \mu f \;\equiv\; \mu g) \quad\Leftarrow\quad \forall\langle S{::}\; x \in f.S \;\equiv\; g.(x \in S)\rangle \;\; .$$

That is, the recursive equation with underlying endofunction f is replaced by the equation with underlying endofunction g (mapping booleans to booleans) if we can establish the property

$$\forall\langle S{::}\; x \in f.S \;\equiv\; g.(x \in S)\rangle \;\; .$$

An example of where this is always possible is testing whether the empty word is in the language defined by a context-free grammar. For concreteness, consider again the grammar with just one nonterminal S and productions

$$S \quad ::= \quad aS \quad | \quad SS \quad | \quad \varepsilon$$

Then the function f maps set X to

$$\{a\}{\cdot}X \cup X{\cdot}X \cup \{\varepsilon\} \ .$$

We compute the function g as follows:

$$\begin{array}{ll}
 & \varepsilon \in f.S \\
= & \{ \quad \text{definition of } f \quad \} \\
 & \varepsilon \in (\{a\}{\cdot}S \cup S{\cdot}S \cup \{\varepsilon\}) \\
= & \{ \quad \text{membership distributes through set union} \quad \} \\
 & \varepsilon \in \{a\}{\cdot}S \ \vee \ \varepsilon \in S{\cdot}S \ \vee \ \varepsilon \in \{\varepsilon\} \\
= & \{ \quad \varepsilon \in X{\cdot}Y \ \equiv \ \varepsilon \in X \wedge \varepsilon \in Y \quad \} \\
 & (\varepsilon \in \{a\} \wedge \varepsilon \in S) \ \vee \ (\varepsilon \in S \wedge \varepsilon \in S) \ \vee \ \varepsilon \in \{\varepsilon\} \\
= & \{ \quad \bullet \quad g.b \ = \ (\varepsilon \in \{a\} \wedge b) \vee (b \wedge b) \vee \varepsilon \in \{\varepsilon\} \ , \\
 & \quad \text{see below for why the rhs has not been} \\
 & \quad \text{simplified further} \quad \} \\
 & g.(\varepsilon \in S) \ .
\end{array}$$

We have thus derived that

$$\varepsilon \in \mu\langle X{::}\ \{a\}{\cdot}X \cup X{\cdot}X \cup \{\varepsilon\}\rangle \ \equiv \ \mu\langle b{::}\ (\varepsilon \in \{a\} \wedge b) \vee (b \wedge b) \vee \varepsilon \in \{\varepsilon\}\rangle \ .$$

Note how the definition of g has the same structure as the definition of f. Effectively set union has been replaced by disjunction and concatenation has been replaced by conjunction. Of course, g can be simplified further (to the constant function true) but that would miss the point of the example.

Now suppose that instead of taking x to be the empty word we consider any word other than the empty word. Then, repeating the above calculation with "$\varepsilon \in$" replaced everywhere by "$x \in$", the calculation breaks down at the second step. This is because the empty word is the only word x that satisfies the property

$$x \in X{\cdot}Y \ \equiv \ x \in X \wedge x \in Y$$

for all X and Y. Indeed, taking x to be a for illustration purposes, we have

$$a \in \mu\langle X\!:: \{a\}\cdot X \cup X\cdot X \cup \{\varepsilon\}\rangle \;\equiv\; \mathsf{true}$$

but

$$\mu\langle b\!:: (a\in\{a\} \wedge b) \vee (b \wedge b) \vee a\in\{\varepsilon\}\rangle \;\equiv\; \mathsf{false}\ \ .$$

This second example emphasises that the conclusion of μ-fusion — $\mu h = f.\mu g$ — demands *two* properties of f, g and h, namely that f be a lower adjoint, and that $f\bullet g = h\bullet f$. The rule is nevertheless very versatile since being a lower adjoint is far from being uncommon, and many algebraic properties take the form $f\bullet g = h\bullet f$ for some functions f, g and h. In cases when the rule is not immediately applicable we have to seek generalisations of f and/or g that do satisfy both properties. For the membership problem above, this is achieved by generalising the problem of applying the function $(x\in)$ to the language but the *matrix* of functions $(u\in)$ where u is a segment of x. This is the basis of the Cocke-Younger-Kasami algorithm for general context-free language recognition.

7.3 Uniqueness

An important issue when confronted with a fixed point equation is whether or not the equation has a *unique* solution.

Uniqueness is a combination of there being at least one solution and at most one solution. In a complete lattice every monotonic function has a least and a greatest fixed point, so the question is whether the least and greatest fixed points are equal.

A very important special case is when the equation

$$x\!::\ x = a + b\cdot x$$

has a unique solution in a Kleene algebra.

In order to give a general answer to this question, it is necessary to make a further assumption about the partial ordering relation on elements in the algebra. The assumption we make is that the elements in the algebra form a complete lattice under the partial ordering relation and, furthermore, this ordering is such that, for each a, the section $(a+)$ is the upper adoint in a Galois connection. This important property is one that distinguishes a *regular algebra* from a Kleene algebra. The other additional property of a regular algebra (which we do not need to consider until later) is that each section $(b\cdot)$ is a lower adjoint in a Galois connection.

An ordering relation that is complete and is such that sections of the binary supremum operator are upper adjoints in a Galois connection is called a *complete, completely distributive lattice.* (It is called completely distributive because being an upper adjoint is equivalent to being universally distributive over all infima.) A *regular algebra* is thus a Kleene algebra for which underlying poset is a complete, completely distributive lattice and such that each section $(\,b\cdot\,)$ is a lower adjoint in a Galois connection.

The assumption of being completely distributive clearly holds for languages and relations, since in both cases the ordering relation is the subset relation and addition is set union and we have the shunting rule:

$$b \subseteq a \cup c \;\equiv\; \neg a \cap b \subseteq c \quad .$$

The assumption that $(\,a+\,)$ is the upper adoint in a Galois connection allows us to use fusion as follows:

Theorem 99 If $(\,y+\,)$ is an upper adjoint, then we have, for all a and b,

$$\nu\langle x::\, a+x\cdot b\rangle = y+\nu\langle x::x\cdot b\rangle \;\Leftarrow\; y = a+y\cdot b \quad .$$

Proof

$$\begin{array}{ll}
 & \nu\langle x::\, a+x\cdot b\rangle = y+\nu\langle x::x\cdot b\rangle \\
\Leftarrow & \qquad \{ \qquad (\,y+\,) \text{ is upper adjoint: } \nu\text{-fusion} \quad \} \\
 & \forall\langle x::\; a+(y+x)\cdot b = y+x\cdot b\rangle \\
\Leftarrow & \qquad \{ \qquad (\,\cdot b\,) \text{ distributes over } +\text{, associativity of } + \quad \} \\
 & a+y\cdot b = y \quad .
\end{array}$$

□

As a consequence, in a regular algebra, the *largest* solution of the equation $x::\; x = a+x\cdot b$ is the sum (i.e. supremum) of an arbitrary solution and the largest solution of the equation $x::\, x = x\cdot b$. Note that a special choice for y in theorem 99 is $y = a\cdot b^*$.

An immediate corollary of theorem 99 is that if $\nu\langle x::x\cdot b\rangle = 0$, function $\langle x::\, a+x\cdot b\rangle$ has a unique fixed point. This is the rule we call the *unique extension property* (UEP) of regular algebra.

Theorem 100 (The unique extension property (UEP)) Suppose b is an element of a regular algebra. If $\nu\langle x::x\cdot b\rangle = 0$, then, for all a and x,

$$a+x\cdot b = x \;\;\equiv\;\; x = a\cdot b^* \quad .$$

□

The UEP draws attention to the importance of property $\nu\langle x::x\cdot b\rangle = 0$. In language theory it is equivalent to $\varepsilon\notin b$ since if, on the contrary, x is a non-empty set such that $x = x\cdot b$ then the length of the shortest word in x must be equal to the length of the shortest word in b plus the length of the shortest word in x. That is, the length of the shortest word in b is zero. The terminology that is often used is "b does not possess the *empty-word property*". In relation algebra we say "b is well-founded": the property expresses that there are no infinite sequences of b-related elements (thus, if relation b represents a finite directed graph, $\nu\langle x{::}\,x{\cdot}b\rangle = 0$ means that the graph is acyclic).

Exercise 101 Consider the final three instances of a Kleene algebra shown in table 6.3.1. In each of these instances, suppose b is a square matrix with entries drawn from the carrier set of the relevant instance. (Equivalently, suppose b is a graph with edge labels drawn from the carrier set. If the (i,j)th matrix entry is 0 then there is no edge from i to j in the graph.) *What is the interpretation of* $\nu\langle x{::}\,x{\cdot}b\rangle = 0$ *in each case?*

□

7.4 Parameterised Prefix Points

Often fixed point equations are parameterised, sometimes explicitly, sometimes implicitly. An example of implicit parameters is the definition of the star operator discussed in section 6.3. In the defining equations given there (see for example equation (79)) a and b are parameters; the equations do not depend on how a and b are instantiated. The equations define the *function* mapping a to a^*. The rules we present in this section are concerned with such parameterised fixed point equations.

The first rule we present, the abstraction rule, is rarely given explicitly although it is very fundamental. The second rule, called the "beautiful theorem" because of its power and generality, is well-known in one particular form —its application to proving continuity properties of functions defined by fixed point equations— but is less well known at the level of generality given here.

7.4.1 The Abstraction Rule

The definition of a^* in section 6.3 is a very good example of the "abstraction rule". Recall that a^* is defined to be the least prefix point of the function mapping x to $1+a{\cdot}x$. Thus a is a *parameter* in the definition of the star operator. The remarkable, and very fundamental, property is that the star operator is itself a least prefix point. Indeed, suppose we consider the function

$$\langle g{::}\ \langle a{::}\ 1 + a\cdot g.a\rangle\rangle\quad .$$

(Note that this function maps an endofunction g on the Kleene algebra to an endofunction on the Kleene algebra and so is a (higher order) endofunction.) Then the star operator is the least fixed point of this function. That is,

$$\langle a:: a^*\rangle \;=\; \mu\langle g:: \langle a:: 1 + a\cdot g.a\rangle\rangle \quad .$$

We leave the verification of this fact as an exercise. (We prove a more general theorem shortly.)

The general theorem that this illustrates is obtained in the following way. We begin by making explicit the fact that a is a parameter in the definition of the star operator. This we do by considering the binary operator $\oplus$ defined by

$$a{\oplus}x = 1 + a{\cdot}x \quad .$$

Then, by definition, $a^* \;=\; \mu\langle x:: 1 + a{\cdot}x\rangle$. So

$$\langle a:: a^*\rangle \;=\; \langle a:: \mu\langle x:: a{\oplus}x\rangle\rangle \quad .$$

This is just a repetition of the definition in section 6.3 but slightly more roundabout. But now we consider the function F from functions to functions defined by

$$F \;=\; \langle g:: \langle a:: a\oplus g.a\rangle\rangle \quad .$$

(That is, $(F.g).a \;=\; a\oplus g.a$.) So for the particular definition of $\oplus$ given above,

$$F \;=\; \langle g:: \langle a:: 1 + a\cdot g.a\rangle\rangle \quad .$$

Then the theorem is that

$$\langle a:: \mu\langle x:: a{\oplus}x\rangle\rangle \;=\; \mu F$$

for all (monotonic) binary operators $\oplus$ and all F defined as above. That is, for all monotonic binary operators $\oplus$,

(102) $\quad \langle a:: \mu\langle x:: a{\oplus}x\rangle\rangle \;=\; \mu\langle g:: \langle a:: a\oplus g.a\rangle\rangle \quad .$

This rule we call the *abstraction rule*.

In order to verify the validity of the rule we have to verify that $\langle a:: \mu\langle x:: a{\oplus}x\rangle\rangle$ is a prefix point of the function F, and that it is at most any prefix point of F. This turns out to be straightforward. First, it is a prefix point by the following argument:

$$\begin{array}{ll}
 & F.\langle a:: \mu\langle x:: a{\oplus}x\rangle\rangle \;\dot{\sqsubseteq}\; \langle a:: \mu\langle x:: a{\oplus}x\rangle\rangle \\
\equiv & \qquad \{ \quad \text{pointwise ordering, definition of } (F.g).a \quad \} \\
 & \forall\langle a:: a\oplus\langle a:: \mu\langle x:: a{\oplus}x\rangle\rangle.a \sqsubseteq \mu\langle x:: a{\oplus}x\rangle\rangle \\
\equiv & \qquad \{ \quad \mu\langle x:: a{\oplus}x\rangle \text{ is a prefix point of } (\,a\oplus\,) \quad \} \\
 & \mathsf{true} \quad .
\end{array}$$

Second, it is at most any prefix point. Suppose g is a prefix point of F, i.e. $F.g \mathrel{\dot{\sqsubseteq}} g$. Then

$$
\begin{array}{ll}
 & \langle a{::}\ \mu\langle x{::}\ a{\oplus}x\rangle\rangle \mathrel{\dot{\sqsubseteq}} g \\
\equiv & \qquad \{ \qquad \text{pointwise ordering} \quad \} \\
 & \forall\langle a{::}\ \mu\langle x{::}\ a{\oplus}x\rangle \sqsubseteq g.a\rangle \\
\Leftarrow & \qquad \{ \qquad \text{fixed point induction: (70)} \quad \} \\
 & \forall\langle a{::}\ a \oplus g.a \sqsubseteq g.a\rangle \\
\equiv & \qquad \{ \qquad \text{pointwise ordering, definition of } (F.g).a \quad \} \\
 & F.g \mathrel{\dot{\sqsubseteq}} g \\
\equiv & \qquad \{ \qquad g \text{ is a prefix point of } F \quad \} \\
 & \mathsf{true} \quad .
\end{array}
$$

This concludes the proof.

Exercise 103 Suppose the operator $\oplus$ is defined by $x{\oplus}y = f.y$. *What property does one then obtain by instantiating the abstraction rule?*

□

7.4.2 The Beautiful Theorem

A very important application of the abstraction rule in combination with fixed point fusion is a theorem that has been dubbed "beautiful" by Dijkstra and Scholten [DS90, p. 159].

As observed precisely above, a parameterised least-fixed-point equation defines a function in terms of a binary operator. The beautiful theorem is that this function enjoys any kind of supremum-preserving property enjoyed by the binary operator. Thus supremum-preserving properties are preserved in the process of constructing least fixed points.

Dijkstra and Scholten formulate the "beautiful theorem" in the context of the predicate calculus. In terms of poset theory the theorem is stated as follows. Suppose that $\mathcal{A} = (A,\sqsubseteq)$ and $\mathcal{B} = (B,\preceq)$ are ordered sets and suppose $\oplus \in (\mathcal{A}{\leftarrow}\mathcal{A}){\leftarrow}\mathcal{B}$. (Thus $\oplus$ is a monotonic function mapping elements of the set B into monotonic endofunctions on the set A .) As in section 7.4.1 denote application of $\oplus$ to x by $x\oplus$. Assume $\mu(x\oplus)$ exists for each x and consider the function $\langle x{::}\, \mu(x\oplus)\rangle$, which we denote by $\dagger$. The "beautiful theorem" is that $\dagger$ enjoys any type of supremum-preserving property that is enjoyed by the (uncurried binary) function $\oplus$. More precisely,

$$\forall\langle f{::}\ \sqcup(\dagger{\bullet}f) = \dagger(\sqcup f)\rangle \Leftarrow \forall\langle f, g{::}\ \sqcup(f\dot{\oplus}g) = (\sqcup f) \oplus (\sqcup g)\rangle \quad ,$$

where, by definition, $f\dot{\oplus}g$ is the function mapping x to $f.x\oplus g.x$, $\bullet$ denotes function composition and $\sqcup$ denotes the supremum operator on functions of the shape of f and g.

The supremum operator, if it exists for a class of functions of a certain shape, is the lower adjoint in a Galois connection. This is the key to the proof of the beautiful theorem. In order to make this explicit we formulate the theorem yet more abstractly. The theorem is that for arbitrary function G that is the lower adjoint in a Galois connection, and arbitrary function H (G and H having of course appropriate types),

$$\forall\langle f::\ G.(\dagger\bullet f)=\dagger(H.f) \Leftarrow \forall\langle g::\ G.(f\dot{\oplus}g) = H.f\oplus G.g\rangle\rangle \quad .$$

The proof is very short. We first note that $(\dagger\bullet f).x=\dagger(f.x)=\mu((f.x)\oplus)$, by definition of $\dagger$. Thus, by the abstraction rule, $\dagger\bullet f=\mu(f\dot{\oplus})$. This is just what we need in order to apply the fusion theorem.

$$\begin{array}{ll}
 & G.(\dagger\bullet f)=\dagger.(H.f) \\
\equiv & \qquad\{\qquad \text{above} \quad\} \\
 & G.\mu(f\dot{\oplus}) = \mu((H.f)\oplus) \\
\Leftarrow & \qquad\{\qquad \text{by assumption, } G \text{ is a lower adjoint.} \\
 & \qquad\qquad\ \ \text{fusion: theorem 95} \quad\} \\
 & \forall\langle g::\ G.(f\dot{\oplus}g) = H.f\oplus G.g\rangle \quad .
\end{array}$$

7.4.3 Cocontinuity of Iteration

We conclude the discussion of parameterised fixed points by considering the implications of the beautiful theorem for the iteration operator. Specifically, we show that in a regular algebra the iteration operator is *cocontinuous*[5]. Specifically, this is the property that for an ascending sequence $f.0\leq f.1\leq \ldots \leq f.n\leq \ldots$

$$(\Sigma\langle n:\ 0\leq n:\ f.n\rangle)^* \ =\ \Sigma\langle n:\ 0\leq n:\ (f.n)^*\rangle \quad .$$

A property of a regular algebra that we need to complete the proof is that, for all such sequences f and for all z,

$$z\cdot\Sigma\langle n:\ 0\leq n:\ f.n\rangle \ =\ \Sigma\langle n:\ 0\leq n:\ z\cdot f.n\rangle$$

and $\quad \Sigma\langle n:\ 0\leq n:\ f.n\rangle\cdot z \ =\ \Sigma\langle n:\ 0\leq n:\ f.n\cdot z\rangle$

[5]There is a disturbing confusion between the standard terminology in the computing science literature and the literature on category theory in this respect. What is called a "continuity" property in the computing science literature is called a "*co*continuity" property in the category theory literature. We use the category theory terminology here.

The property of iteration is thus that in a Kleene algebra in which the sections $(\cdot z)$ and $(z\cdot)$ are cocontinuous, for all z, the iteration operator * is also cocontinuous. (This proviso is satisfied in a regular algebra and thus in all Kleene algebras of interest to us.)

Recalling the discussion in the introduction to this section, the instantiation of the beautiful theorem we wish to consider is as follows: the function $\dagger$ is the operator * (i.e. the function $\langle a{::}\ a^*\rangle$; correspondingly the binary operator $\oplus$ is defined by $a{\oplus}x = 1+a{\cdot}x$. The functions G and H are both taken to be the supremum operator of ascending functions on the natural numbers to the carrier of a regular algebra, which we denote by Σ.

The instantiation of the beautiful theorem (with $G,H,\dagger\ :=\ \Sigma,\Sigma,^*$) gives us, for all ascending sequences f:

$$\Sigma\langle n{::}\ (f.n)^*\rangle = (\Sigma f)^* \quad \Leftarrow \quad \forall\langle f,g{::}\ \Sigma\langle n{::}\ \ 1+f.n\cdot g.n\rangle \ = \ 1+(\Sigma f){\cdot}(\Sigma g)\rangle$$

where the dummies n range over the natural numbers, and the dummies f and g range over ascending sequences. We try to simplify the antecedent of this theorem. First, we have:

$$\begin{array}{ll}
 & \Sigma\langle n{::}\ \ 1+f.n\cdot g.n\rangle \ \le\ 1+(\Sigma f){\cdot}(\Sigma g) \\
\equiv & \qquad\{ \qquad \text{definition of supremum} \quad \} \\
 & \forall\langle n{::}\ \ 1+f.n\cdot g.n \ \le\ 1+(\Sigma f){\cdot}(\Sigma g)\rangle \\
\Leftarrow & \qquad\{ \qquad + \text{ is the binary supremum operator} \quad \} \\
 & \forall\langle n{::}\ f.n\cdot g.n \ \le\ (\Sigma f){\cdot}(\Sigma g)\rangle \\
\equiv & \qquad\{ \qquad \text{property of suprema} \quad \} \\
 & \mathsf{true} \quad .
\end{array}$$

So it is the opposite inclusion that is crucial. Now,

$$\begin{array}{ll}
 & (\Sigma f){\cdot}(\Sigma g) \\
= & \qquad\{ \qquad \bullet \quad \text{assume, for all } z,\ (\cdot z) \text{ is cocontinuous} \quad \} \\
 & \Sigma\langle n{::}\ \ f.n\cdot(\Sigma g)\rangle \\
= & \qquad\{ \qquad \bullet \quad \text{assume, for all } z,\ (z\cdot) \text{ is cocontinuous} \quad \} \\
 & \Sigma\langle n{::}\ \Sigma\langle m{::}\ \ f.n\cdot g.m\rangle\rangle \quad .
\end{array}$$

So,

$$\begin{array}{ll}
 & \Sigma\langle n{::}\ \ 1+f.n\cdot g.n\rangle \ \ge\ 1+(\Sigma f){\cdot}(\Sigma g) \\
\equiv & \qquad\{ \qquad + \text{ is the binary supremum operator} \quad \}
\end{array}$$

$$
\begin{array}{ll}
 & \Sigma\langle n{::}\ 1 + f.n \cdot g.n\rangle \geq 1 \ \wedge\ \Sigma\langle n{::}\ 1 + f.n \cdot g.n\rangle \geq (\Sigma f)\cdot(\Sigma g) \\
\Leftarrow & \{ \quad \Sigma\langle n{::}\ 1{+}c\rangle \geq 1 \text{, for all } c \text{, and } 1 + f.n \cdot g.n \ \geq\ f.n \cdot g.n \quad \} \\
 & \Sigma\langle n{::}\ f.n \cdot g.n\rangle \ \geq\ (\Sigma f)\cdot(\Sigma g) \\
\equiv & \{ \quad \text{above} \quad \} \\
 & \Sigma\langle n{::}\ f.n \cdot g.n\rangle \ \geq\ \Sigma\langle n{::}\ \Sigma\langle m{::}\ f.n \cdot g.m\rangle\rangle \\
\equiv & \{ \quad \text{suprema, dummy change} \quad \} \\
 & \forall\langle n, m{::}\ \Sigma\langle p{::}\ f.p \cdot g.p\rangle \ \geq\ f.n \cdot g.m\rangle \\
\Leftarrow & \{ \quad \Sigma\langle p{::}\ f.p \cdot g.p\rangle \ \geq\ f.(n{\uparrow}m) \cdot g.(n{\uparrow}m) \\
 & \quad\ \ \text{where } (\, n{\uparrow}m \,) \text{ denotes the maximum of } m \text{ and } n \quad \} \\
 & \forall\langle n, m{::}\ f.(n{\uparrow}m) \cdot g.(n{\uparrow}m) \ \geq\ f.n \cdot g.m\rangle \\
\equiv & \{ \quad f \text{ and } g \text{ are ascending sequences, product is monotonic} \quad \} \\
 & \text{true} \ \ .
\end{array}
$$

This completes the proof.

7.5 Mutual Recursion

In all but the simplest cases, recursive functions are defined by *mutual recursion*. For example, the BNF definition of the syntax of a programming language uses mutual recursion to simultaneously define the syntax of expressions, statements, declarations etc. A definition by mutual recursion can be seen as a *single* fixed point equation where there is one unknown —a vector of values— and the function defining the unknown maps a vector to a vector. On the other hand, one also wishes to view a definition by mutual recursion as a collection of individual equations, defining several unknown values, and which may be solved on an individual basis. In this section we show that these two views are compatible with each other. We present one theorem to the effect that least prefix points can be calculated by solving individual equations and back-substituting.

Without loss of generality we can confine ourselves to fixed point equations involving two unknowns: a fixed point equation with $n{+}1$ unknowns can always be viewed as an equation on two unknowns, one of which is a vector of length n.

Any fixed point of a function mapping pairs of values to pairs of values is, of course, a pair. We show how to calculate the individual components of the least fixed point of such a function, given that it is possible to calculate least fixed points of functions mapping single elements to single elements. The fixed-point equation to be dealt with is the following.

(104) $x,y{::}\ x = x\odot y \ \wedge\ y = x\otimes y$.

We first consider two special cases in which $\odot$ and $\otimes$ depend on one of their arguments only.

Lemma 105

(a) $(\ \mu f\,,\mu g\ ) = \mu\langle x,y{::}\,(f.x,g.y)\rangle$

(b) $(\ \mu(f\bullet g)\,,\mu(g\bullet f)\ ) = \mu\langle x,y{::}\,(f.y,g.x)\rangle$.

Proof of (a). This is an instance of the abstraction rule. To see this, rename the functions to f_0 and f_1 and their arguments to x_0 and x_1. Then (102) says that

$$\langle i{::}\ \mu\langle x{::}\ f_i.x\rangle\rangle \ = \ \mu\langle g{::}\ \langle i{::}\, f_i.(g.i)\rangle\rangle$$

Identifying the pair $(\,x_0,x_1\,)$ with the function $\langle i{::}\ x.i\rangle$,

$$\mu\langle g{::}\ \langle i{::}\, f_i.(g.i)\rangle\rangle \ = \ \mu\langle x_0,x_1\ {::}\ (f_0.x_0\,,f_1.x_1)\rangle \ .$$

So

$$\langle i{::}\ \mu\langle x{::}\ f_i.x\rangle\rangle \ = \ \mu\langle x_0,x_1\ {::}\ (f_0.x_0\,,f_1.x_1)\rangle$$

as required.
Proof of (b)

$$\begin{array}{ll}
 & (\ \mu(f\bullet g)\,,\mu(g\bullet f)\ ) = \mu\langle x,y{::}\,(f.y,g.x)\rangle \\
\Leftarrow & \qquad \{ \qquad \text{(a) on lhs} \quad \} \\
 & \mu\langle x,y{::}\,(f.g.x\,,g.f.y)\rangle = \mu\langle x,y{::}\ (f.y,g.x)\rangle \\
\equiv & \qquad \{ \qquad \text{define } \phi:\ \phi(x,y) = (f.y\,,g.x) \quad \} \\
 & \mu(\phi\bullet\phi) = \mu\phi \\
\equiv & \qquad \{ \qquad \text{square rule} \quad \} \\
 & \mathsf{true}\ .
\end{array}$$

□

With the aid of lemma (105), we now compute the least solution of (104), viz. we prove

Theorem 106

$$(\ \mu\langle x ::\ x \odot p.x\rangle\ ,\ \mu\langle y ::\ q.y \otimes y\rangle\ ) = \mu\langle x,y :: (x\odot y, x\otimes y)\rangle$$

where $p.x = \mu\langle v :: x\otimes v\rangle$ and $q.y = \mu\langle u :: u\odot y\rangle$, i.e. $p.x$ and $q.y$ are the least fixed points of the individual equations.

Proof

$$\begin{array}{ll}
 & \mu\langle x,y ::\ (x\odot y, x\otimes y)\rangle \\
= & \quad \{ \quad \text{diagonal rule (92)} \quad \} \\
 & \mu\langle x,y ::\ \mu\langle u,v :: (u\odot y, x\otimes v)\rangle\rangle \\
= & \quad \{ \quad \text{lemma (105a) and definition of } p \text{ and } q \quad \} \\
 & \mu\langle x,y ::\ (q.y, p.x)\rangle \\
= & \quad \{ \quad \text{lemma (105b)} \quad \} \\
 & (\ \mu\langle x ::\ q.(p.x)\rangle\ ,\ \mu\langle y ::\ p.(q.y)\rangle\ ) \\
= & \quad \{ \quad \text{definition } q\ ,\ p \quad \} \\
 & (\ \mu\langle x ::\ \mu\langle u ::\ u \odot p.x\rangle\rangle\ ,\ \mu\langle y ::\ \mu\langle v ::\ q.y \otimes v\rangle\rangle\ ) \\
= & \quad \{ \quad \text{diagonal rule (92) twice: } u := x\ ,\ v := y \quad \} \\
 & (\ \mu\langle x ::\ x \odot p.x\rangle\ ,\ \mu\langle y ::\ q.y \otimes y\rangle\ )\ \ .
\end{array}$$

□

This concludes the presentation of the calculus. The corresponding calculus of greatest fixed points is obtained by the interchanges $\mu \leftrightarrow \nu$, $\sqsubseteq \leftrightarrow \sqsupseteq$, and lower adjoint $\leftrightarrow$ upper adjoint.

Exercise 107 Suppose f is a (unary) function and $\otimes$ is a binary operator. Prove the following:

(a) $\mu\langle x,y :: (f.x\ , x\otimes y)\rangle\ =\ (\ \mu f\ ,\ \mu\langle y :: \mu f \otimes y\rangle\ )$

(b) $\mu\langle x,y :: (f.y\ , x\otimes y)\rangle\ =\ \mu\langle x,y :: (f.(x\otimes y)\ , x\otimes y)\rangle$.

Hint: Use theorem 106 for (a). Use the diagonal rule and lemma 105 for (b).

□

7.6 An Illustration — Arithmetic Expressions

The Algol 60 definition of arithmetic expressions, restricted to just multiplication and addition, has the following form. The nonterminals $\langle\mathsf{Term}\rangle$ and $\langle\mathsf{Factor}\rangle$ serve to introduce a precedence of multiplication over addition, and the use of left recursion (for example, "$\langle\mathsf{Expression}\rangle$" is repeated as the leftmost symbol on the right of the production $\langle\mathsf{Expression}\rangle ::= \langle\mathsf{Expression}\rangle + \langle\mathsf{Term}\rangle$) serves to define a left-to-right evaluation order on repeated occurrences of the same operator.

$$\begin{array}{lcl}
\langle\mathsf{Expression}\rangle & ::= & \langle\mathsf{Expression}\rangle + \langle\mathsf{Term}\rangle \;\;|\;\; \langle\mathsf{Term}\rangle \\
\langle\mathsf{Term}\rangle & ::= & \langle\mathsf{Term}\rangle \times \langle\mathsf{Factor}\rangle \;\;|\;\; \langle\mathsf{Factor}\rangle \\
\langle\mathsf{Factor}\rangle & ::= & (\,\langle\mathsf{Expression}\rangle\,) \;\;|\;\; \langle\mathsf{Variable}\rangle
\end{array}$$

A much simpler grammar for such arithmetic expressions takes the following form.

$$\begin{array}{lcl}
\langle\mathsf{Expression}\rangle & ::= & \langle\mathsf{Expression}\rangle + \langle\mathsf{Expression}\rangle \\
& & \;|\;\; \langle\mathsf{Expression}\rangle \times \langle\mathsf{Expression}\rangle \\
& & \;|\;\; (\,\langle\mathsf{Expression}\rangle\,) \\
& & \;|\;\; \langle\mathsf{Variable}\rangle
\end{array}$$

This grammar imposes no precedence of multiplication over addition and is ambivalent about the order in which repeated additions or multiplications are evaluated. (The former is clearly undesirable but the latter is of no consequence because addition and multiplication are associative.)

We want to prove that the languages defined by $\langle\mathsf{Expression}\rangle$ in the two grammars above are equal. The proof is a combination of the fixed point calculus presented in this section and Kleene algebra presented in chapter 6.3.

To avoid confusion, let us rename the nonterminal and terminal symbols in the grammars. (We need to rename the nonterminals in order not to confuse the two instances of $\langle\mathsf{Expression}\rangle$; we need to rename the terminal symbols because of the use of "$+$" and the parentheses "(" and ")" in both regular expressions (the meta language) and in arithmetic expressions (the object language).)

Choosing letters a, b, c, d and e for "$+$", "$\times$", "(", ")" and $\langle\mathsf{Variable}\rangle$, and renaming $\langle\mathsf{Expression}\rangle$, $\langle\mathsf{Term}\rangle$ and $\langle\mathsf{Factor}\rangle$ as E, T and F, respectively, the first grammar is transformed to the following.

$$\begin{array}{lclcl}
E & ::= & E\,a\,T & | & T \\
T & ::= & T\,b\,F & | & F \\
F & ::= & c\,E\,d & | & e
\end{array}$$

Using the same renaming for the terminal symbols but renaming $\langle\mathsf{Expression}\rangle$ as D, we get the following for the second grammar.

$$D \;::=\; D\,a\,D \;\mid\; D\,b\,D \;\mid\; c\,D\,d \;\mid\; e$$

The task is now to prove that the languages generated by E and D are equal. Note that the proof does not rely on a, b, c, d and e being terminal symbols. They may denote arbitrary languages.

We begin with an informal calculation. Then we show how the informal calculation is justified using the rules of the fixed point calculus.

The informal calculation treats the grammars as systems of simultaneous equations. We have four equations:

(108) $E \;=\; E{\cdot}a{\cdot}T + T$,

(109) $T \;=\; T{\cdot}b{\cdot}F + F$,

(110) $F \;=\; c{\cdot}E{\cdot}d + e$,

(111) $D \;=\; D{\cdot}a{\cdot}D + D{\cdot}b{\cdot}D + c{\cdot}D{\cdot}d + e$.

The first step is to "solve" the equations for E and T. That is, we eliminate E and T from, respectively, the right sides of equations (108) and (109). We get:

(112) $E = T{\cdot}(a{\cdot}T)^*$

and

(113) $T = F{\cdot}(b{\cdot}F)^*$.

Now we substitute the right side of the equation for T in the right side of the equation for E. We get:

$$E \;=\; F{\cdot}(b{\cdot}F)^*{\cdot}(a{\cdot}F{\cdot}(b{\cdot}F)^*)^*$$

which can be simplified using star decomposition (exercise 87(e)) to

$$E \;=\; F{\cdot}(b{\cdot}F + a{\cdot}F)^* \quad .$$

This can in turn be simplified, using the "arithmetic" rules of Kleene algebra, to

$$E \;=\; F{\cdot}((a{+}b){\cdot}F)^* \quad .$$

So, applying exercise 93(c),

$$E = \mu\langle X{::}\; F + X{\cdot}(a{+}b){\cdot}X\rangle \quad .$$

Now we substitute the right side of equation (110) into this last equation, in order to eliminate F. We obtain

$$E = \mu\langle X{::}\ c\cdot E\cdot d + e + X\cdot(a+b)\cdot X\rangle \quad .$$

"Solving" this equation by eliminating E on the right side:

$$E = \mu\langle Y{::}\ \mu\langle X{::}\ c\cdot Y\cdot d + e + X\cdot(a+b)\cdot X\rangle\rangle \quad .$$

The last step is to apply the diagonal rule, together with a little "arithmetic":

$$E = \mu\langle X{::}\ X\cdot a\cdot X + X\cdot b\cdot X + c\cdot X\cdot d + e\rangle \quad .$$

That is, E solves the equation (111) for D.

Let us now conduct this calculation formally using the fixed point calculus. The first two steps combine the use of the diagonal rule with exercise 107(a) to justify the derivation of equations (112) and (113). The third step uses exercise 107(b) to justify the substitution of $F\cdot(b\cdot F)^*$ for T in the informal proof.

$$\begin{array}{ll}
 & \mu\langle E,T,F{::}\ (E\cdot a\cdot T + T\ ,\ T\cdot b\cdot F + F\ ,\ c\cdot E\cdot d + e)\rangle \\
= & \{\quad \text{diagonal rule}\quad\} \\
 & \mu\langle E,T,F{::}\ \mu\langle X,Y,Z{::}\ (X\cdot a\cdot T + T\ ,\ Y\cdot b\cdot F + F\ ,\ c\cdot E\cdot d + e)\rangle\rangle \\
= & \{\quad \text{exercise 107(a) and definition of } {}^*\ \text{(twice)}\quad\} \\
 & \mu\langle E,T,F{::}\ (T\cdot(a\cdot T)^*\ ,\ F\cdot(b\cdot F)^*\ ,\ c\cdot E\cdot d + e)\rangle \\
= & \{\quad \text{exercise 107(b)}\quad\} \\
 & \mu\langle E,T,F{::}\ ((F\cdot(b\cdot F)^*\cdot(a\cdot F\cdot(b\cdot F)^*)^*)\ ,\ F\cdot(b\cdot F)^*\ ,\ c\cdot E\cdot d + e)\rangle \quad .
\end{array}$$

The remaining steps in the calculation do not alter the subexpressions "$F\cdot(b\cdot F)^*$" and "$c\cdot E\cdot d + e$". We therefore introduce abbreviations for these subexpressions. Note how formally all elements of the system of equations being solved have to be carried along whereas it is only the equation for E that is being manipulated, just as in the informal calculation.

$$\begin{array}{ll}
 & \mu\langle E,T,F{::}\ ((F\cdot(b\cdot F)^*\cdot(a\cdot F\cdot(b\cdot F)^*)^*)\ ,\ F\cdot(b\cdot F)^*\ ,\ c\cdot E\cdot d + e)\rangle \\
= & \{\quad \text{introduce the abbreviation } \Phi \text{ for } F\cdot(b\cdot F)^* \\
 & \qquad \text{and } \Psi \text{ for } c\cdot E\cdot d + e\quad\} \\
 & \mu\langle E,T,F{::}\ (F\cdot(b\cdot F)^*\cdot(a\cdot F\cdot(b\cdot F)^*)^*\ ,\ \Phi\ ,\ \Psi)\rangle \\
= & \{\quad \text{star decomposition (exercise 87(e)), arithmetic}\quad\} \\
 & \mu\langle E,T,F{::}\ (F\cdot((a+b)\cdot F)^*\ ,\ \Phi\ ,\ \Psi)\rangle
\end{array}$$

$=\quad\{\quad$ exercise 93(c) $\quad\}$

$\mu\langle E,T,F{::}\ (\mu\langle X{::}\ F+X\cdot(a+b)\cdot X\rangle\ ,\Phi\ ,\Psi)\rangle$

$=\quad\{\quad$ exercise 107(b), definition of $\Psi\quad\}$

$\mu\langle E,T,F{::}\ (\mu\langle X{::}\ c\cdot E\cdot d\ +\ e+X\cdot(a+b)\cdot X\rangle\ ,\Phi\ ,\Psi)\rangle\ .$

In the final part of the calculation, we again use exercise 107(a) in combination with the diagonal rule.

$\mu\langle E,T,F{::}\ (\mu\langle X{::}\ c\cdot E\cdot d\ +\ e+X\cdot(a+b)\cdot X\rangle\ ,\Phi\ ,\Psi)\rangle$

$=\quad\{\quad$ exercise 107(a) $\quad\}$

$\mu\langle E,T,F{::}\ \mu\langle X,Y,Z{::}\ (X\cdot a\cdot X+X\cdot b\cdot X+c\cdot E\cdot d+e\ \ ,\ \ \Phi\ \ ,\ \ \Psi)\rangle\rangle$

$=\quad\{\quad$ diagonal rule $\quad\}$

$\mu\langle E,T,F{::}\ (E\cdot a\cdot E+E\cdot b\cdot E+c\cdot E\cdot d+e\ ,\ \Phi\ ,\ \Psi)\rangle$

$=\quad\{\quad$ exercise 107(a)

where $\Theta\ =\ \mu\langle E{::}\ \ E\cdot a\cdot E+E\cdot b\cdot E+c\cdot E\cdot d+e\rangle\quad\}$

$(\mu\langle E{::}\ \ E\cdot a\cdot E+E\cdot b\cdot E+c\cdot E\cdot d+e\rangle\ ,\ \mu\langle T,F{::}\ (F\cdot(b\cdot F)^*\ ,\ c\cdot\Theta\cdot d\ +\ e)\rangle)\ .$

No use has been made of the mutual recursion rule, theorem 106, in this calculation although it is applicable to several steps. This is often the case. Theorem 106 is a combination of the diagonal rule and lemma (105) and exercise 107 is another combination. Often it is easier to use the simpler rules, or some straightforward combination of them, than it is to apply the more complicated rule.

8 Further Reading

In spite of the name, the general concept of a "Galois" connection was not invented by the famous mathematician Évariste Galois, but an instance of a "Galois connection" is *the* "Galois correspondence" between groups and extension fields to which Galois owes his fame. The term "adjoint function" used here (and in category theory texts) is derived from the original terminology used by Galois. (Extending a field involves "adjoining" new values to the field.) Detailed discussion of the Galois correspondence can be found in [Ste89, Fen91].

Feijen [FB90] has written a delightful article on the Galois connection defining the maximum of two numbers. The source of several of the properties of the floor and ceiling functions discussed in section (2) is [GKP89]. The Galois connections defining the floor and ceiling function are mentioned in [GKP89] but never used because the authors "can never remember which inequality to use"! Further elementary examples of Galois connections can be found in [Lam94].

The idea of extracting a Galois connection from a relation would appear to be due to Hartmanis and Stearns [HS64, HS66]. The same idea appears in [Lam81]. In that paper Lambek gives several mathematical examples of polarities. Further examples in the areas of geometry and the theory of rings and groups can be found in Birkhoff's classic text [Bir67]. The notion of *concept analysis* is discussed in detail in [GW99]. Conditional correctness assertions were studied by Hoare in [Hoa69] and are called *Hoare triples*.

The conclusion of section 5 appears in [MSS92]; the solution presented here is based on Aarts [Aar92]. The use of the prefix lemma has been strongly influenced by Lambek's article on so-called *subequalizers* [Lam70]. Lambek [Lam81] summarises the philosophy of dialectical contradictions as "the view [due to Heraclitus] of the universe as a sort of divine debating society in which opposing ideas are forever struggling to produce a synthesis". He attributes the phrase "unity of opposites" to Mao Tse-tung, and the idea that dialectical contradictions are illustrated by adjoint functions[6] to discussions he had with Bill Lawvere in 1965-6. Lambek and Scott [LS86] describe the unity of opposites theorem as "the most interesting consequence of a Galois correspondence". Our own use of the name "unity-of-opposites" theorem in this text is not intended to suggest some deep philosophical significance but simply to help make the theorem stick in one's memory.

[6]Actually funct*ors*, not funct*ions*. Monotonic functions are a particular case of the notion in category theory of a functor, and a Galois connection is a particular case of the notion of an adjunction. Many of the examples in Lambek's paper are however Galois connections and can be understood without a knowledge of category theory.

References

[Aar92] C.J. Aarts. Galois connections presented calculationally. Afstudeer verslag (Graduating Dissertation), Department of Computing Science, Eindhoven University of Technology, 1992.

[BC75] 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.

[Bir67] Garrett Birkhoff. *Lattice Theory*, volume 25 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, Rhode Island, 3rd edition, 1967.

[DP90] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge Mathematical Textbooks. Cambridge University Press, first edition, 1990.

[DS90] E.W. Dijkstra and C.S. Scholten. *Predicate Calculus and Program Semantics*. Springer-Verlag, Berlin, 1990.

[FB90] W.H.J. Feijen and Lex Bijlsma. Exercises in formula manipulation. In E.W. Dijkstra, editor, *Formal Development of Programs and Proofs*, pages 139–158. Addison-Wesley Publ. Co., 1990.

[Fen91] Maureen H. Fenrick. *Introduction to the Galois Correspondence*. Birkhaüser Boston, 1991.

[Gen69] G. Gentzen. Investigations into logical deduction. In M.E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–213. North-Holland, Amsterdam, 1969.

[GHK$^+$80] G. Gierz, K. H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D. S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, 1980.

[GKP89] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics : a Foundation for Computer Science*. Addison-Wesley Publishing Company, 1989.

[GW99] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis. Mathematical Foundations*. Springer-Verlag Berlin Heidelberg, 1999. Translated from the German by Cornelia Franzke. Title of the original German edition: Formale Begriffsanalyse – Mathematische Grundlagen.

[Hoa69] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.

[HS64] J. Hartmanis and R.E. Stearns. Pair algebras and their application to automata theory. *Information and Control*, 7(4):485–507, 1964.

[HS66] J. Hartmanis and R.E. Stearns. *Algebraic Structure Theory of Sequential Machines*. Prentice-Hall, 1966.

[Kle56] 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.

[Koz94] Dexter Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994.

[Lam70] Joachim Lambek. Subequalizers. *Canadian Mathematical Bulletin*, 13(3):337–349, 1970.

[Lam81] J. Lambek. The influence of Heraclitus on modern mathematics. In J. Agassi and R.S.Cohen, editors, *Scientific Philosophy Today*, pages 111–121. D. Reidel Publishing Co., 1981.

[Lam94] J. Lambek. Some Galois connections in elementary number theory. *J. of Number Theory*, 47(3):371–377, June 1994.

[LS86] J. Lambek and P.J. Scott. *Introduction to Higher Order Categorical Logic*, volume 7 of *Studies in Advanced Mathematics*. Cambridge University Press, 1986.

[MSS92] 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.

[Nau63] P. (Ed.) Naur. Revised report on the algorithmic language ALGOL 60. *Comm. ACM*, 6:1–20, Also in *The Computer Journal*, 5: 349–67 (1963); *Numerische Mathematik*, 4: 420–52 (1963) 1963.

[Ore44] Oystein Ore. Galois connexions. *Transactions of the American Mathematical Society*, 55:493–513, 1944.

[Sch53] J. Schmidt. Beitrage für filtertheorie. II. *Math. Nachr.*, 10:197–232, 1953.

[Ste89] Ian Stewart. *Galois Theory*. Chapman and Hall, 2nd edition, 1989.

9 Solutions to Exercises

7 (a) We have, for all n,

$$
\begin{array}{ll}
 & n \geq -\lfloor x \rfloor \\
= & \qquad \{ \quad \text{negation} \quad \} \\
 & -n \leq \lfloor x \rfloor \\
= & \qquad \{ \quad \text{definition of floor} \quad \} \\
 & -n \leq x \\
= & \qquad \{ \quad \text{negation} \quad \} \\
 & n \geq -x \\
= & \qquad \{ \quad \text{definition of ceiling} \quad \} \\
 & n \geq \lceil -x \rceil \quad .
\end{array}
$$

Thus, by indirect equality, the function f is the ceiling function.
(b) We have, for all n,

$$
\begin{array}{ll}
 & n \leq \lfloor x+m \rfloor \\
= & \qquad \{ \quad \text{definition of floor} \quad \} \\
 & n \leq x+m \\
= & \qquad \{ \quad \text{arithmetic} \quad \} \\
 & n-m \leq x \\
= & \qquad \{ \quad \text{definition of floor} \quad \} \\
 & n-m \leq \lfloor x \rfloor \\
= & \qquad \{ \quad \text{arithmetic} \quad \} \\
 & n \leq \lfloor x \rfloor + m \quad .
\end{array}
$$

The result follows by indirect equality.
(c) We have, for all n,

$$
\begin{array}{ll}
 & n \leq \lfloor x/m \rfloor \\
= & \qquad \{ \quad \text{definition of floor} \quad \} \\
 & n \leq x/m \\
= & \qquad \{ \quad \text{arithmetic, } m \text{ is positive} \quad \}
\end{array}
$$

$$n\times m\leq x$$
$= \quad \{ \quad$ definition of floor $\quad \}$
$$n\times m\leq \lfloor x\rfloor$$
$= \quad \{ \quad$ arithmetic, m is positive $\quad \}$
$$n\leq \lfloor x\rfloor /m$$
$= \quad \{ \quad$ definition of floor $\quad \}$
$$n\leq \lfloor\lfloor x\rfloor /m\rfloor \quad .$$

The result follows by indirect equality.
(d) We have, for all n,

$$n\leq \left\lfloor\sqrt{\lfloor x\rfloor}\right\rfloor$$
$= \quad \{ \quad$ n is an integer, definition 3 $\quad \}$
$$n\leq \sqrt{\lfloor x\rfloor}$$
$= \quad \{ \quad$ arithmetic $\quad \}$
$$n^2\leq \lfloor x\rfloor \ \vee n<0$$
$= \quad \{ \quad$ n^2 is an integer, definition 3 $\quad \}$
$$n^2\leq x \vee n<0$$
$= \quad \{ \quad$ arithmetic $\quad \}$
$$n\leq \sqrt{x}$$
$= \quad \{ \quad$ n is an integer, definition 3 $\quad \}$
$$n\leq \left\lfloor\sqrt{x}\right\rfloor \quad .$$

The result follows by indirect equality.
□

22 Suppose first that F and G are inverse poset monomorphisms. Then,

$$F.x\sqsubseteq y$$
$\Leftarrow \quad \{ \quad$ F is the inverse of G. Thus $F.(G.y)=y$ $\quad \}$
$$F.x\sqsubseteq F.(G.y)$$
$\equiv \quad \{ \quad$ F is a poset isomorphism $\quad \}$
$$x\sqsubseteq G.y \quad .$$

Thus, $F.x \sqsubseteq y \Leftarrow x \sqsubseteq G.y$. Symmetrically, $x \sqsubseteq G.y \Leftarrow F.x \sqsubseteq y$. Combining these two properties we get that , (F, G) is a Galois connection between $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$ and $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$. Replacing F by G and G by F, we also get that (G, F) is a Galois connection between $(\mathcal{B}, \sqsubseteq_{\mathcal{B}})$ and $(\mathcal{A}, \sqsubseteq_{\mathcal{A}})$.

Suppose now that (F, G) and (G, F) are both Galois connections. Then, by theorem 14, $F.(G.y) \sqsubseteq y$, because (F, G) is a Galois connection, and $y \sqsubseteq F.(G.y)$, because (G, F) is a Galois connection. Thus $F.(G.y) = y$. Symmetrically, $G.(F.x) = x$. That is, F and G are inverses (and thus surjective). Moreover, both F and G are monotonic. Thus,

$$
\begin{array}{ll}
 & x \sqsubseteq y \\
\Rightarrow & \{ \quad F \text{ is monotonic} \quad \} \\
 & F.x \sqsubseteq F.y \\
\Rightarrow & \{ \quad G \text{ is monotonic} \quad \} \\
 & G.(F.x) \sqsubseteq G.(F.y) \\
\Rightarrow & \{ \quad F \text{ and } G \text{ are inverses} \quad \} \\
 & x \sqsubseteq y \quad .
\end{array}
$$

That is, F is a poset monomorphism. Symmetrically, G is a poset monomorphism.
□

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

$$
\begin{array}{ll}
 & F_0 \bullet h \mathrel{\dot{\sqsubseteq}} k \bullet F_1 \\
\equiv & \{ \quad (16) \quad \} \\
 & h \mathrel{\dot{\sqsubseteq}} G_0 \bullet k \bullet F_1 \\
\Rightarrow & \{ \quad \text{monotonicity} \quad \} \\
 & h \bullet G_1 \mathrel{\dot{\sqsubseteq}} G_0 \bullet k \bullet F_1 \bullet G_1 \\
\Rightarrow & \{ \quad \text{cancellation: (15),} \\
 & \quad\quad \text{monotonicity and transitivity of } \dot{\sqsubseteq} \quad \} \\
 & h \bullet G_1 \mathrel{\dot{\sqsubseteq}} G_0 \bullet k
\end{array}
$$

(The hints “monotonicity” refer to the fact that the function $\bullet f$ is monotonic for all functions f .)

□

31 Take $m=2$ and $n=4$ Let G_0, G_2 and G_5 be the floor function; also let G_1 and G_3 be the function f and G_4 to be the function, U, embedding the integers in the reals. Let F_0, F_2 and F_5 be U; also let F_1 and F_3 be the lower adjoint of the function f and F_4 be the ceiling function. The right side of (27) can then be read as the requirement that the lower adjoint of f maps integers to integers.
□

49 The relevant Galois connections take the form:

$$F_x.b \subseteq P \equiv b \Rightarrow x \in P$$

and

$$x \in P \Rightarrow b \equiv P \subseteq G_x.b$$

for some functions F and G. In each case the construction of the closed formula involves a case analysis on b. The simplest cases are $F_x.\mathsf{false} = \phi$ and $G_x.\mathsf{true} = \mathcal{U}$. The other two cases are $F_x.\mathsf{true} = \{x\}$ and $G_x.\mathsf{false} = \mathcal{U}/\{x\}$ (where $\mathcal{U}/\{x\}$ denotes the set of all elements of $\mathcal{U}$ different from x). Thus,

$$F_x.b = \textbf{if } b \textbf{ then } \{x\} \textbf{ else } \phi \quad ,$$

and

$$G_x.b = \textbf{if } b \textbf{ then } \mathcal{U} \textbf{ else } \mathcal{U}/\{x\} \quad .$$

□

50 The negation of $(\neq \phi)$ is the predicate $(= \phi)$ which is equivalent to the predicate $(\subseteq \phi)$. By the dual of 46, the latter is universally $\cup$-junctive, assuming booleans are ordered by follows-from. That is, for all sets, P, of subsets of U,

$$\cup P = \phi \equiv \forall \langle S : S \in P : S = \phi \rangle \quad .$$

Thus, since $\neg(\forall p) \equiv \exists(\neg p)$,

$$\cup P \neq \phi \equiv \exists \langle S : S \in P : S \neq \phi \rangle \quad .$$

This is the sense in which $(\neq \phi)$ preserves set union: the predicate is a function from subsets of $\mathcal{U}$ ordered by set inclusion to Bool ordered by implication. The upper adjoint is the function mapping boolean b to **if** b **then** $\mathcal{U}$ **else** ϕ. That is, for all sets S and all Booleans b,

$$(S \neq \phi \Rightarrow b) \ \equiv\ S \subseteq (\textbf{if } b \textbf{ then } \mathcal{U} \textbf{ else } \phi) \quad .$$

□

51 Applying (23), we have

$$\mathsf{F} \mathrel{\dot{\sqsubseteq}} \mathsf{G}^{\flat} \equiv \mathsf{G} \mathrel{\dot{\sqsubseteq}} \mathsf{F}^{\sharp} \quad .$$

and

$$\mathsf{G}^{\flat} \mathrel{\dot{\sqsubseteq}} \mathsf{F} \equiv \mathsf{F}^{\sharp} \mathrel{\dot{\sqsubseteq}} \mathsf{G} \quad .$$

Thus, by exercise 22, $^{\sharp}$ and $^{\flat}$ are inverse poset isomorphisms between the set of inf-preserving functions ordered by the usual pointwise ordering $\dot{\sqsubseteq}$ and the set of sup-preserving functions ordered by $\dot{\sqsupseteq}$, the converse of the pointwise ordering $\dot{\sqsubseteq}$.
□

52 Suppose $\mathcal{A}$ is a complete poset. Let $\sqcap$ denote the infimum operator for $\mathcal{A}$. We use theorem 47 to show that $\mathcal{A}$ is cocomplete.

Consider the relation R between the set of functions with range $\mathcal{A}$ and $\mathcal{A}$ defined by

$$(f,a) \in R \;\equiv\; \forall\langle b{::}\; f.b \sqsubseteq a\rangle \quad .$$

Then, applying theorem 47, there is a function $\sqcup$ satisfying (35) if and only if the predicate $\langle a{::}\; (f,a) \in R\rangle$ is inf-preserving. We verify that this is indeed the case as follows:

$$\begin{array}{ll}
 & \langle a{::}\; (f,a) \in R\rangle \quad \text{is inf-preserving} \\
\equiv & \{ \quad \text{definition of inf-preserving: (43)} \\
 & \quad\ \ \text{with } p.x \equiv \forall\langle b{::}\; f.b \sqsubseteq x\rangle \quad \} \\
 & \forall\langle g{::}\; \forall\langle b{::}\; f.b \sqsubseteq \sqcap g\rangle \equiv \forall\langle x{::}\; \forall\langle b{::}\; f.b \sqsubseteq g.x\rangle\rangle\rangle \\
\equiv & \{ \quad \text{definition of infimum: (33)} \quad \} \\
 & \forall\langle g{::}\; \forall\langle b{::}\forall\langle x{::}\; f.b \sqsubseteq g.x\rangle\rangle \equiv \forall\langle x{::}\forall\langle b{::}\; f.b \sqsubseteq g.x\rangle\rangle\rangle \\
\equiv & \{ \quad \text{predicate caculus} \quad \} \\
 & \mathsf{true} \quad .
\end{array}$$

□

56

$$\begin{array}{ll}
 & \sqcup_{\mathcal{A}}.(f\bullet h) \sqsubseteq f.(\sqcup_{\mathcal{C}}.h) \\
\equiv & \{ \quad \text{definition} \quad \} \\
 & f\bullet h \mathrel{\dot{\sqsubseteq}} K_{\mathcal{A}}.(f.(\sqcup_{\mathcal{C}}.h))
\end{array}$$

$\equiv \qquad \{ \quad (54) \quad \}$

$\quad f \bullet h \stackrel{.}{\sqsubseteq} f \bullet K_{\mathcal{A}}.(\sqcup_{\mathcal{C}}.h)$

$\Leftarrow \qquad \{ \quad f \text{ is monotonic} \quad \}$

$\quad h \stackrel{.}{\sqsubseteq} K_{\mathcal{A}}.(\sqcup_{\mathcal{C}}.h)$

$\equiv \qquad \{ \quad \text{cancellation} \quad \}$

$\quad \mathsf{true}$.

□

63 (a) Suppose first that f is reflexive, idempotent and monotonic. Then

$\quad x \sqsubseteq f.y$

$\Rightarrow \qquad \{ \quad f \text{ is monotonic} \quad \}$

$\quad f.x \sqsubseteq f.(f.y)$

$\equiv \qquad \{ \quad f \text{ is idempotent} \quad \}$

$\quad f.x \sqsubseteq f.y$

$\Rightarrow \qquad \{ \quad f \text{ is reflexive} \quad \}$

$\quad x \sqsubseteq f.y$.

Now suppose that $x \sqsubseteq f.y \equiv f.x \sqsubseteq f.y$ for all x and y. Instantiating y to x we get that f is reflexive. Now monotonicity follows:

$\quad f.x \sqsubseteq f.y$

$\equiv \qquad \{ \quad \text{assumption} \quad \}$

$\quad x \sqsubseteq f.y$

$\Leftarrow \qquad \{ \quad f \text{ is reflexive, transitivity} \quad \}$

$\quad x \sqsubseteq y$.

Finally, idempotency is proved by a mutual inclusion argument. The inclusion $f \stackrel{.}{\sqsubseteq} f \bullet f$ is immediate from reflexivity. The other inclusion is the instance of the assumption obtained by instantiating x to $f.x$ and y to x.

(b)

$\quad H \bullet f \bullet G \bullet g \stackrel{.}{\sqsubseteq} H \bullet f \bullet G \bullet h$

$\Leftarrow \qquad \{ \quad \text{monotonicity of } H\bullet \quad \}$

$\quad f \bullet G \bullet g \stackrel{.}{\sqsubseteq} f \bullet G \bullet h$

$\equiv \quad \{ \quad f$ is a closure operator $\}$

$G{\bullet}g \mathrel{\dot{\sqsubseteq}} f{\bullet}G{\bullet}h$

$\equiv \quad \{$ Galois connection $\}$

$g \mathrel{\dot{\sqsubseteq}} H{\bullet}f{\bullet}G{\bullet}h$

$\Leftarrow \quad \{ \quad I \mathrel{\dot{\sqsubseteq}} H{\bullet}f{\bullet}G$

$\quad\quad \Leftarrow \quad \{ \quad f$ is reflexive, monotonicity $\}$

$\quad\quad I \mathrel{\dot{\sqsubseteq}} H{\bullet}G$

$\quad\quad \equiv \quad \{$ cancellation $\}$

$\quad\quad \mathsf{true}$.

$\quad\quad$ transitivity of $\dot{\sqsubseteq}$ and monotonicity $\}$

$H{\bullet}f{\bullet}G{\bullet}g \mathrel{\dot{\sqsubseteq}} H{\bullet}f{\bullet}G{\bullet}h$.

Comparing the first, seventh and last lines, we have thus shown that

$$g \mathrel{\dot{\sqsubseteq}} H{\bullet}f{\bullet}G{\bullet}h \;\equiv\; H{\bullet}f{\bullet}G{\bullet}g \mathrel{\dot{\sqsubseteq}} H{\bullet}f{\bullet}G{\bullet}h$$

as required.
□

64 Assume x and y are elements of $F.\mathcal{B}$. Note that this is equivalent to $x = F.(G.x)$ and $y = F.(G.y)$.

$x \sqcup_{F.\mathcal{B}} y \;=\; y$

$\equiv \quad \{ \quad x \sqcup_{F.\mathcal{B}} y \;=\; F.(G.x \sqcup_{\mathcal{B}} G.y)$,

$\quad\quad y = F.(G.y) \quad \}$

$F.(G.x \sqcup_{\mathcal{B}} G.y) \;=\; F.(G.y)$

$\equiv \quad \{$ unity-of-opposites theorem

$\quad\quad$ (F is an isomorphism) $\}$

$G.x \sqcup_{\mathcal{B}} G.y = G.y$

$\equiv \quad \{$ suprema $\}$

$G.x \sqsubseteq_{\mathcal{B}} G.y$

$\equiv \quad \{$ unity-of-opposites theorem

$\quad\quad$ (G is an isomorphism) $\}$

$x \sqsubseteq_{\mathcal{A}} y$.

□

66 The "at-most" relation is the least prefix point of the function f where, for any relation R,

$$f.R \;=\; \{n\colon n{\in}\mathbb{N}\colon (0,n)\} \cup \{m,n\colon (m,n){\in}R\colon (m{+}1\,,\, n{+}1)\} \;\; .$$

□

76

$\mu f \sqsubseteq \nu f$

$\Leftarrow$ { induction }

$f.\nu f \sqsubseteq \nu f$

$\equiv$ { computation: $f.\nu f = \nu f$ }

true .

□

78

$\leq$ is reflexive

$\equiv$ { definition of reflexive }

$\forall\langle x{::}\, x \leq x\rangle$

$\equiv$ { definition of $\leq$ }

$\forall\langle x{::}\; x{+}x = x\rangle$

$\equiv$ { definition of idempotent }

$+$ is idempotent.

$\leq$ is transitive

$\equiv$ { definition of transitive }

$\forall\langle x,y,z{::}\;\; x \leq y \wedge y \leq z \Rightarrow x \leq z\rangle$

$\equiv$ { definition of $\leq$ }

$\forall\langle x,y,z{::}\;\; x{+}y = y \,\wedge\, y{+}z = z \;\Rightarrow\; x{+}z = z\rangle$

Now suppose $x{+}y = y$ and $y{+}z = z$. Then,

$$\begin{array}{ll}
& x+z \\
= & \{ \quad y+z = z \quad \} \\
& x+(y+z) \\
= & \{ \quad + \text{ is associative} \quad \} \\
& (x+y)+z \\
= & \{ \quad x+y = y \quad \} \\
& y+z \\
= & \{ \quad y+z = z \quad \} \\
& z \quad .
\end{array}$$

Thus $\leq$ is transitive.
Now for the antisymmetry of $\leq$.

$$\begin{array}{ll}
& x \leq y \wedge y \leq x \\
\equiv & \{ \quad \text{definition of } \leq \quad \} \\
& x+y = y \wedge y+x = x \\
\equiv & \{ \quad + \text{ is symmetric} \quad \} \\
& x+y = y \wedge x+y = x \\
\Rightarrow & \{ \quad \text{substitution} \quad \} \\
& x = y \quad .
\end{array}$$

We now show that $+$ is monotonic.

$$\begin{array}{ll}
& x+y \leq x+z \\
\equiv & \{ \quad \text{definition of } \leq \quad \} \\
& (x+y)+(x+z) = x+z \\
\equiv & \{ \quad \text{arithmetic, } + \text{ is idempotent} \quad \} \\
& x+y+z = x+z \\
\Leftarrow & \{ \quad \text{Leibniz} \quad \} \\
& y+z = z \\
\equiv & \{ \quad \text{definition of } \leq \quad \} \\
& y \leq z \quad .
\end{array}$$

To show that $\cdot$ is monotonic we have to show that it is monotonic in both its arguments. We only show that it is monotonic in its right argument. The other case is symmetrical.

$$\begin{array}{ll}
 & x\cdot y \le x\cdot z \\
\equiv & \quad \{ \quad \text{definition of } \le \quad \} \\
 & x\cdot y + x\cdot z = x\cdot z \\
\equiv & \quad \{ \quad \text{distributivity} \quad \} \\
 & x\cdot(y+z) = x\cdot z \\
\Leftarrow & \quad \{ \quad \text{Leibniz} \quad \} \\
 & y+z = z \\
\equiv & \quad \{ \quad \text{definition of } \le \quad \} \\
 & y \le z \quad .
\end{array}$$

To show that $+$ is the binary supremum operator we have to show that

$$x+y \le z \equiv x \le z \wedge y \le z \quad .$$

This we do as follows.

$$\begin{array}{ll}
 & x+y \le z \\
\equiv & \quad \{ \quad \text{definition of } \le \quad \} \\
 & (x+y)+z = z \\
\equiv & \quad \{ \quad \text{substitution} \quad \} \\
 & (x+y)+z = z \wedge x+z = x+((x+y)+z) \\
\equiv & \quad \{ \quad + \text{ is associative and idempotent} \quad \} \\
 & (x+y)+z = z \wedge x+z = (x+y)+z \\
\Rightarrow & \quad \{ \quad \text{substitution} \quad \} \\
 & x+z = z \\
\equiv & \quad \{ \quad \text{definition of } \le \quad \} \\
 & x \le z \quad .
\end{array}$$

Symmetrically, $x+y \le z \Rightarrow y \le z$. Finally, $x \le z \wedge y \le z \Rightarrow x+y \le z$ by the monotonicity of $+$.

$\square$

87 (a)

$$\begin{array}{ll} & a \cdot b^* \leq c^* \cdot a \\ \Leftarrow & \quad \{ \quad (83) \quad \} \\ & a + c^* \cdot a \cdot b \;\leq\; c^* \cdot a \\ \Leftarrow & \quad \{ \quad c^* \cdot a \;=\; a + c^* \cdot c \cdot a \text{, monotonicity} \quad \} \\ & a{\cdot}b \leq c{\cdot}a \quad . \end{array}$$

(b)

$$\begin{array}{ll} & c^* \cdot a \leq a \cdot b^* \\ \Leftarrow & \quad \{ \quad (80) \quad \} \\ & a + c \cdot a \cdot b^* \;\leq\; a \cdot b^* \\ \Leftarrow & \quad \{ \quad a \cdot b^* \;=\; a + a \cdot b \cdot b^* \text{, monotonicity} \quad \} \\ & c{\cdot}a \leq a{\cdot}b \quad . \end{array}$$

(c) By combining (a) and (b) using the anti-symmetry of the ordering relation we obtain

$$a \cdot b^* = c^* \cdot a \Leftarrow a{\cdot}b = c{\cdot}a \quad .$$

Thus, with $b,c \;:=\; b{\cdot}a\,, a{\cdot}b$, we have:

$$a \cdot (b{\cdot}a)^* = (a{\cdot}b)^* \cdot a \Leftarrow a{\cdot}(b{\cdot}a) = (a{\cdot}b){\cdot}a \quad .$$

The property now follows from the associativity of product.
(d) First we use induction:

$$\begin{array}{ll} & (a{+}b)^* \leq b^* \cdot (a \cdot b^*)^* \\ \Leftarrow & \quad \{ \quad (80) \quad \} \\ & 1 + (a{+}b) \cdot b^* \cdot (a \cdot b^*)^* \;\leq\; b^* \cdot (a \cdot b^*)^* \end{array}$$

Now we show that $1 + (a{+}b) \cdot b^* \cdot (a \cdot b^*)^* \;=\; b^* \cdot (a \cdot b^*)^*$.

$$\begin{array}{ll} & 1 + (a{+}b) \cdot b^* \cdot (a \cdot b^*)^* \\ = & \quad \{ \quad \text{arithmetic} \quad \} \\ & 1 + a \cdot b^* \cdot (a \cdot b^*)^* + b \cdot b^* \cdot (a \cdot b^*)^* \\ = & \quad \{ \quad 1 + x \cdot x^* \;=\; x^* \text{ with } x \;:=\; a \cdot b^* \quad \} \\ & (a \cdot b^*)^* + b \cdot b^* \cdot (a \cdot b^*)^* \end{array}$$

$=$ { arithmetic }

$(1 + b \cdot b^*) \cdot (a \cdot b^*)^*$

$=$ { $1 + x \cdot x^* = x^*$ with $x := b$ }

$b^* \cdot (a \cdot b^*)^*$.

The opposite inequality is proved as follows:

$b^* \cdot (a \cdot b^*)^*$

$\leq$ { monotonicity }

$(a{+}b)^* \cdot ((a{+}b) \cdot (a{+}b)^*)^*$

$\leq$ { $x \cdot x^* \leq x^*$ with $x := a{+}b$, monotonicity }

$(a{+}b)^* \cdot ((a{+}b)^*)^*$

$=$ { $x^* = x^* \cdot x^* = (x^*)^*$ with $x := a{+}b$ }

$(a{+}b)^*$.

Finally, $b^* \cdot (a \cdot b^*)^* = (b^* \cdot a)^* \cdot b^*$ is an instance of (c).
(e)

$(a^*)^* = a^*$

$\equiv$ { anti-symmetry }

$(a^*)^* \leq a^* \wedge a^* \leq (a^*)^*$

$\Leftarrow$ { * is a closure operator and * is monotonic }

$a^* \leq a^* \wedge a \leq a^*$

$\equiv$ { reflexivity and $a^* = 1 + a{\cdot}a^* = 1 + a + a{\cdot}a{\cdot}a^*$ }

true .

□

88 (a) $a^+ = \{\text{definition}\}\ a \cdot a^* \geq \{a^* \geq 1\}\ a{\cdot}1 = a$.
(b) This is an application of the mirror rule, 87(c), with b instantiated to 1 .
(c)

$a^+ \cdot a^+$

$=$ { definition, (b) }

$a \cdot a^* \cdot a^* \cdot a$

$=$ { $a^* = a^* \cdot a^*$ }

$$a \cdot a^* \cdot a$$
$\leq$ { $a^* = 1 + a^* \cdot a$ }
$$a \cdot a^*$$
$=$ { definition }
$$a^+ \ .$$

(d)

$$a + x \cdot x \leq x$$
$\Rightarrow$ { induction (83), with $a,b,x := x,a,x$ }
$$a \cdot x^* \leq x$$
$\equiv$ { $x^* = 1 + x \cdot x^*$, arithmetic }
$$a + a \cdot x \cdot x^* \leq x$$
$\Rightarrow$ { $1 \leq x^*$, addition and multiplication are monotonic }
$$a + a \cdot x \leq x$$
$\Rightarrow$ { induction (80), with $a,b,x := a,a,x$ }
$$a^* \cdot a \leq x$$
$\equiv$ { (a) }
$$a^+ \leq x \ .$$

(e) First, we have that a^* is a prefix point of the function $\langle x{::}\ 1 + a + x \cdot x \rangle$ since $1 \leq a^*$, $a \leq a^*$ and $a^* \cdot a^* \leq a^*$. We now show that it is the least prefix point:

$$1 + a + x \cdot x \leq x$$
$\equiv$ { definition of suprema }
$$1 \leq x \ \wedge \ a + x \cdot x \leq x$$
$\Rightarrow$ { (d) }
$$1 \leq x \wedge a^+ \leq x$$
$\equiv$ { definition of suprema }
$$1 + a^+ \leq x$$
$\equiv$ { $a^* = 1 + a \cdot a^* = 1 + a^+$ }
$$a^* \leq x \ .$$

□

93 (a)

$$\begin{array}{ll}
 & \mu\langle X{::}\ a\cdot X^*\rangle \\
= & \{\quad \text{definition of } a\cdot X^* \quad\} \\
 & \mu\langle X{::}\ \mu\langle Y{::}\ a+Y\cdot X\rangle\rangle \\
= & \{\quad \text{diagonal rule} \quad\} \\
 & \mu\langle X{::}\ a+X\cdot X\rangle \\
= & \{\quad \text{exercise 88} \quad\} \\
 & a^+ \quad .
\end{array}$$

(b)

$$\begin{array}{ll}
 & \mu\langle X{::}\ (a{+}X)^*\rangle \\
= & \{\quad \text{definition of } {}^* \quad\} \\
 & \mu\langle X{::}\ \mu\langle Y{::}\ 1+(a{+}X)\cdot Y\rangle\rangle \\
= & \{\quad \text{diagonal rule} \quad\} \\
 & \mu\langle X{::}\ 1+(a{+}X)\cdot X\rangle \\
= & \{\quad \text{distributivity of concatenation over sum} \quad\} \\
 & \mu\langle X{::}\ 1+a\cdot X+X\cdot X\rangle \\
= & \{\quad \text{diagonal rule} \quad\} \\
 & \mu\langle X{::}\ \mu\langle Y{::}\ 1+a\cdot X+Y\cdot Y\rangle\rangle \\
= & \{\quad \text{exercise 88(e)} \quad\} \\
 & \mu\langle X{::}\ (a\cdot X)^*\rangle \\
= & \{\quad \text{rolling rule} \quad\} \\
 & (\mu\langle X{::}\ a\cdot X^*\rangle)^* \\
= & \{\quad \text{(a)} \quad\} \\
 & (a^+)^* \\
= & \{\quad \text{exercise 87(e)} \quad\} \\
 & a^* \quad .
\end{array}$$

From this calculation we conclude that

$$\mu\langle X{::}\ (a{+}X)^*\rangle \;=\; \mu\langle X{::}\ (a\cdot X)^*\rangle \;=\; a^* \quad .$$

The final piece is then

$\mu\langle X{::}\ a+X^*\rangle$

$=\quad\{$ rolling rule $\}$

$a+\mu\langle X{::}\ (a+X)^*\rangle$

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

$a+a^*$

$=\quad\{\ a\leq a^*\ \}$

a^* .

(c)

$\mu\langle X{::}\ a+X{\cdot}b{\cdot}X\rangle$

$=\quad\{$ diagonal rule $\}$

$\mu\langle X{::}\ \mu\langle Y{::}\ a+Y{\cdot}b{\cdot}X\rangle\rangle$

$=\quad\{$ definition of * $\}$

$\mu\langle X{::}\ a{\cdot}(b{\cdot}X)^*\rangle$

$=\quad\{$ rolling rule $\}$

$a\cdot\mu\langle X{::}\ (b{\cdot}a{\cdot}X)^*\rangle$

$=\quad\{$ (b) $\}$

$a{\cdot}(b{\cdot}a)^*$.

(d)

$\mu\langle X{::}\ 1+a{\cdot}X{\cdot}b{\cdot}X+b{\cdot}X{\cdot}a{\cdot}X\rangle$

$=\quad\{$ diagonal rule (92) $\}$

$\mu\langle X{::}\ \mu\langle Y{::}\ 1+a{\cdot}X{\cdot}b{\cdot}Y+b{\cdot}X{\cdot}a{\cdot}Y\rangle\rangle$

$=\quad\{$ distributivity $\}$

$\mu\langle X{::}\ \mu\langle Y{::}\ 1+(a{\cdot}X{\cdot}b+b{\cdot}X{\cdot}a){\cdot}Y\rangle\rangle$

$=\quad\{$ definition of * $\}$

$\mu\langle X{::}\ (a{\cdot}X{\cdot}b+b{\cdot}X{\cdot}a)^*\rangle$

$=\quad\{$ exercise 88(e) $\}$

$\mu\langle X{::}\ \mu\langle Y{::}\ 1+a{\cdot}X{\cdot}b+b{\cdot}X{\cdot}a+Y{\cdot}Y\rangle\rangle$

$=$ { diagonal rule (92) }

$\mu\langle X{::}\ 1+a\cdot X\cdot b+b\cdot X\cdot a+X\cdot X\rangle$.

□

94 Rolling rule: First,

$\mu(g\bullet f) \preceq g.\mu(f\bullet g)$

$\Leftarrow$ { induction }

$(g\bullet f).(g.\mu(f\bullet g)) \preceq g.\mu(f\bullet g)$

$\equiv$ { definition of $(g\bullet f)$ }

$g.(f.(g.\mu(f\bullet g))) \preceq g.\mu(f\bullet g)$

$\Leftarrow$ { g is monotonic }

$f.(g.\mu(f\bullet g)) \sqsubseteq \mu(f\bullet g)$

$\equiv$ { definition of $(f\bullet g)$ }

$(f\bullet g)\,.\,\mu(f\bullet g) \sqsubseteq \mu(f\bullet g)$

$\equiv$ { computation rule }

true .

Second, we suppose that y is an arbitrary prefix point of $g\bullet f$ and calculate as follows:

$g.\mu(f\bullet g) \preceq y$

$\Leftarrow$ { • $(g\bullet f).y \preceq y$, transitivity }

$g.\mu(f\bullet g) \preceq (g\bullet f).y$

$\Leftarrow$ { definition of $g\bullet f$, monotonicity of g }

$\mu(f\bullet g) \sqsubseteq f.y$

$\Leftarrow$ { induction }

$(f\bullet g).(f.y) \sqsubseteq f.y$

$\Leftarrow$ { $(f\bullet g).(f.y) = f.((g\bullet f).y)$, monotonicity of f }

$(g\bullet f).y \preceq y$

$\equiv$ { y is a prefix point of $g\bullet f$ }

true .

Square rule:

$$\begin{array}{ll}
& \mu f = \mu(f^2) \\
\equiv & \{ \quad \text{antisymmetry} \quad \} \\
& \mu f \sqsubseteq \mu(f^2) \wedge \mu(f^2) \sqsubseteq \mu f \\
\Leftarrow & \{ \quad \text{induction} \quad \} \\
& f.\mu(f^2) \sqsubseteq \mu(f^2) \ \wedge \ f^2.\mu f \sqsubseteq \mu f \\
\equiv & \{ \quad \text{rolling rule and computation rule (applied twice)} \quad \} \\
& \mathsf{true} \quad .
\end{array}$$

Diagonal rule: First suppose that $\mu\langle y{::}\,x{\oplus}y\rangle$ exists, for all x, as does $\mu\langle x{::}\,\mu\langle y{::}\,x{\oplus}y\rangle\rangle$. We prove that the latter satisfies the definition of the least prefix point of the function $\langle x{::}\,x{\oplus}x\rangle$.

For brevity let us denote $\mu\langle y{::}\,x{\oplus}y\rangle$ by $m.x$ and $\mu\langle x{::}\,\mu\langle y{::}\,x{\oplus}y\rangle\rangle$ by M.
First M is a fixed point of the function:

$$\begin{array}{ll}
& M = M{\oplus}M \\
\equiv & \{ \quad \text{fixed point:} \ \ M = m.M = M \oplus m.M \quad \} \\
& M \oplus m.M = M{\oplus}M \\
\equiv & \{ \quad \text{fixed point:} \ \ M = m.M \quad \} \\
& \mathsf{true} \quad .
\end{array}$$

Second, M is at most every prefix point of the function:

$$\begin{array}{ll}
& M \sqsubseteq x \\
\Leftarrow & \{ \quad \text{induction:} \ \ M = \mu m \quad \} \\
& m.x \sqsubseteq x \\
\Leftarrow & \{ \quad \text{induction:} \ \ m.x = \mu\langle y{::}\,x{\oplus}y\rangle \quad \} \\
& x{\oplus}x \sqsubseteq x \quad .
\end{array}$$

Now, with the same abbreviations as above, we assume that $m.x$ exists for all x. We also introduce the abbreviation N for $\mu\langle x{::}\,x{\oplus}x\rangle$ which we assume to exist. We have to show that N is the least prefix point of the function m.
First it is a prefix point of m.

$$\begin{array}{ll}
& m.N \sqsubseteq N \\
\Leftarrow & \{ \quad \text{induction:} \ \ m.N = \mu\langle y{::}\,N{\oplus}y\rangle \quad \} \\
& N{\oplus}N \sqsubseteq N
\end{array}$$

$\equiv$ { prefix point: $N = \mu\langle x{::}\, x{\oplus}x\rangle$ }

true .

Second, N is at most every prefix point of m:

$N \sqsubseteq x$

$\Leftarrow$ { induction: $N = \mu\langle x{::}\, x{\oplus}x\rangle$ }

$x{\oplus}x \sqsubseteq x$

$\Leftarrow$ { transitivity }

$x{\oplus}x \sqsubseteq x \oplus m.x \sqsubseteq m.x \sqsubseteq x$

$\equiv$ { fixed point: $m.x = x \oplus m.x$ }

$x{\oplus}x \sqsubseteq x \oplus m.x \wedge m.x \sqsubseteq x$

$\equiv$ { $\oplus$ is monotonic }

$m.x \sqsubseteq x$.

□

96

$k.x$ is a prefix point of g

$\equiv$ { definition }

$g.(k.x) \preceq k.x$

$\Leftarrow$ { • $h.x \sqsubseteq x$ (i.e. x is a prefix point of h)
k is monotonic. Thus, $k.(h.x) \preceq k.x$. }

$g.(k.x) \preceq k.(h.x)$.

Abstracting from the variable x and denoting the set of prefix points of h by $\mathsf{Pre}.h$ we have thus established the lemma:

(114) $\forall\langle x{:}\; x{\in}\mathsf{Pre}.h{:}\; k.x \in \mathsf{Pre}.g\rangle \Leftarrow g{\bullet}k \mathrel{\dot{\preceq}} k{\bullet}h$.

In particular, assuming the existence of μh, $k.\mu h$ is a prefix point of g. Thus, since μg is the least prefix point of g,

(115) $\mu g \preceq k.\mu h \quad\Leftarrow\quad g{\bullet}k \mathrel{\dot{\preceq}} k{\bullet}h$.

Now, if k is an upper adjoint in a Galois connection with lower adjoint f then

$$g{\bullet}k \mathrel{\dot{\preceq}} k{\bullet}h \equiv f{\bullet}g \mathrel{\dot{\sqsubseteq}} h{\bullet}f \quad .$$

So, in the case that k is an upper adjoint with lower adjoint f, (114) is equivalent to

(116) $\forall\langle x\colon x{\in}\mathsf{Pre}.h\colon k.x \in \mathsf{Pre}.g\rangle \;\Leftarrow\; f{\bullet}g \mathrel{\dot{\sqsubseteq}} h{\bullet}f$.

Also, (115) is equivalent to:

$$f.\mu g \sqsubseteq \mu h \;\Leftarrow\; f{\bullet}g \mathrel{\dot{\sqsubseteq}} h{\bullet}f \quad .$$

But also, by making the substitutions $k,g,h := f,h,g$ in (114) we obtain immediately:

(117) $\forall\langle y\colon y{\in}\mathsf{Pre}.g\colon f.y \in \mathsf{Pre}.h\rangle \;\Leftarrow\; h{\bullet}f \mathrel{\dot{\sqsubseteq}} f{\bullet}g$.

It is thus evident that, if $h{\bullet}f$ and $f{\bullet}g$ are *equal*, we have both

$$\forall\langle x\colon x{\in}\mathsf{Pre}.h\colon k.x \in \mathsf{Pre}.g\rangle$$

and

$$\forall\langle y\colon y{\in}\mathsf{Pre}.g\colon f.y \in \mathsf{Pre}.h\rangle \quad .$$

Thus the pair of functions f and k forms a Galois connection between the set of prefix points of h and the set of prefix points of g In particular, since lower adjoints map least elements to least elements, we have derived:

$$f.\mu g = \mu h \;\Leftarrow\; f{\bullet}g = h{\bullet}f \quad .$$

□

98 (a) We have:

$$\begin{array}{ll}
 & \mu(g{\bullet}f) \sqsubseteq \mu(h{\bullet}f) \\
\Leftarrow & \{ \quad \text{induction} \quad \} \\
 & (g{\bullet}f)\,.\,\mu(h{\bullet}f) \sqsubseteq \mu(h{\bullet}f) \\
\equiv & \{ \quad \text{composition and rolling rule} \quad \} \\
 & g\,.\,\mu(f{\bullet}h) \sqsubseteq \mu(h{\bullet}f) \\
\Leftarrow & \{ \quad \mu\text{-fusion, } g \text{ is a lower adjoint} \quad \} \\
 & g{\bullet}f{\bullet}h \mathrel{\dot{\sqsubseteq}} h{\bullet}f{\bullet}g \quad .
\end{array}$$

Further,

$$\begin{array}{ll}
 & \mu(f{\bullet}g) \sqsubseteq \mu(f{\bullet}h) \\
\equiv & \{ \quad \text{rolling rule} \quad \} \\
 & f\,.\,\mu(g{\bullet}f) \sqsubseteq f\,.\,\mu(h{\bullet}f)
\end{array}$$

$\Leftarrow \qquad \{ \quad \text{monotonicity of } f \quad \}$

$\mu(g{\bullet}f) \sqsubseteq \mu(h{\bullet}f)$

$\Leftarrow \qquad \{ \quad \text{above} \quad \}$

$g{\bullet}f{\bullet}h \mathrel{\dot{\sqsubseteq}} h{\bullet}f{\bullet}g \quad .$

(b) This part is a straightforward corollary of (a) and the antisymmetry of the ordering relations.

As remarked above, the exchange rule says that g and h can be exchanged within a μ term in which they are composed after or before an arbitrary function f. The generalisation we seek is that they can be exchanged anywhere in a μ term. That is, we seek a condition such that

$$\mu(f_0{\bullet}g{\bullet}f_1) = \mu(f_0{\bullet}h{\bullet}f_1) \quad .$$

We derive such a condition as follows:

$\mu(f_0{\bullet}g{\bullet}f_1) = \mu(f_0{\bullet}h{\bullet}f_1)$

$\equiv \qquad \{ \quad \text{rolling rule} \quad \}$

$f_0\,.\,\mu(g{\bullet}f_1{\bullet}f_0) = f_0\,.\,\mu(h{\bullet}f_1{\bullet}f_0)$

$\Leftarrow \qquad \{ \quad \text{exchange rule} \quad \}$

$g{\bullet}f_1{\bullet}f_0{\bullet}h = h{\bullet}f_1{\bullet}f_0{\bullet}g \quad .$

We don't advise you to remember this rule. It is overly complicated and is so easily recovered from the basic rule.
□

101 In the case of reachability, graph b is acyclic. In the case of shortest paths, the sum of the edge lengths on any cycle of edges is greater than zero. In the case of bottlenecks, there is no cycle of edges each of weight ∞. That is, there is no subset of nodes connected by a cycle of edges with each edge offering no bottleneck.
□

103

$\mu\langle g{::}\ \langle a{::}\ a \oplus g.a\rangle\rangle$

$= \qquad \{ \quad \text{definition of } \oplus \quad \}$

$\mu\langle g{::}\ \langle a{::}\ f.(g.a)\rangle\rangle$

$= \qquad \{ \quad \text{definition of } (\,f{\bullet}\,) \quad \}$

$\mu(f{\bullet})$

$=$ { abstraction rule }

$\langle a{::}\ \mu\langle x{::}\ a\oplus x\rangle\rangle$

$=$ { definition of $\oplus$ }

$\langle a{::}\ \mu\langle x{::}\ f.x\rangle\rangle$

$=$ { η-reduction }

$\langle a{::}\ \mu f\rangle$

Thus the rule is: $\mu(f\bullet) = K.\mu f$ where K is the constant combinator.

□

107 (a)

$\mu\langle x, y{::}\ (f.x\,, x\otimes y)\rangle$

$=$ { theorem 106 with $x\odot y := f.x$ (thus $q.y := \mu\langle u{::}\, f.u\rangle$) }

$(\ \mu\langle x{::}\, f.x\rangle\ ,\ \mu\langle y{::}\ \mu\langle u{::}\, f.u\rangle\otimes y\rangle\ )$

$=$ { $\langle u{::}\, f.u\rangle = f$ }

$(\ \mu f\ ,\ \mu\langle y{::}\ \mu f\otimes y\rangle\ )$.

(b)

$\mu\langle x, y{::}\ (f.y\,, x\otimes y)\rangle$

$=$ { diagonal rule }

$\mu\langle x, y{::}\ \mu\langle u, v{::}\ (g.v\,, x\otimes y)\rangle\rangle$

$=$ { lemma 105(b) with $f := g$ and $g.u := x\otimes y$ }

$\mu\langle x, y{::}\ (\ \mu\langle u{::}\, g.(x\otimes y)\rangle\ ,\ \mu\langle v{::}\, x\otimes y\rangle\ )\rangle$

$=$ { $\mu\langle u{::}\, c\rangle = c$ for any c independent of u (as $\mu f = f.\mu f$) }

$\mu\langle x, y{::}\ (\ g.(x\otimes y)\ ,\ x\otimes y\ )\rangle$.

□