

# Temporal algebra

BURGHARD VON KARGER

*Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel,  
Preusserstraße 1–9, D-24105 Kiel, Germany*

*Received 16 June 1996; revised 13 October 1997*

We develop temporal logic from the theory of complete lattices, Galois connections and fixed points. In particular, we prove that all seventeen axioms of Manna and Pnueli's sound and complete proof system for linear temporal logic can be derived from just two postulates, namely that  $(\oplus, \ominus)$  is a Galois connection and that  $(\ominus, \oplus)$  is a perfect Galois connection. We also obtain a similar result for the branching time logic CTL.

A surprising insight is that most of the theory can be developed without the use of negation. In effect, we are studying intuitionistic temporal logic. Several examples of such structures occurring in computer science are given. Finally, we show temporal algebra at work in the derivation of a simple graph-theoretic algorithm.

This paper is tutorial in style and there are no difficult technical results. To the experts in temporal logics, we hope to convey the simplicity and beauty of algebraic reasoning as opposed to the machine-orientedness of logical deduction. To those familiar with the calculational approach to programming, we want to show that their methods extend easily and smoothly to temporal reasoning. For anybody else, this text may serve as a gentle introduction to both areas.

## 1. Introduction

The enormous success of temporal logic is due to the fact that it combines simplicity and appeal to intuition with practical usefulness. The temporal connectives are easy to grasp, possess nice mathematical properties, and are well suited for specifying properties of concurrent and reactive systems. Powerful decision procedures and model checking algorithms exist and have proved their worth in numerous practical applications.

It is known, although not widely, that temporal logic has an algebraic counterpart. This connection has been neglected, perhaps, because all of the spectacular results (particularly model checking) were obtained using logical, rather than algebraic, methods. We shall argue that the algebraic presentation deserves more attention. The first argument is simplicity. For example, Manna and Pnueli (1991) give a complete proof system for LTL (linear temporal logic) that requires seventeen axioms (not counting Boolean tautologies). In a complete Boolean algebra, all of these can be deduced from just two Galois correspondences.

Our second argument is that in algebra concepts can be introduced and understood at a higher and more user-friendly level. For example, we have the following theorem that relates the 'Next' and 'Previous' operators of temporal logic to the 'Eventually' and 'Once' operators.

**Theorem** If  $(\mathcal{G}, \oplus, \ominus)$  is a Galois algebra, then so is  $(\mathcal{G}, \diamond, \diamond^*)$ , where  $\diamond = \oplus^*$  and  $\diamond^* = \ominus$ .

From this theorem we can infer that every formula valid for  $\oplus$  and  $\ominus$  also holds for  $\diamond$  and  $\diamond^*$ .

Our third argument is that the concise definitions of algebra facilitate the discovery of new models. We have found that the temporal connectives have natural interpretations in a very diverse set of non-standard models, including monotonic predicate transformers, fuzzy relations and prefix-closed sets of traces. None of this can be guessed from reading, say, Manna and Pnueli's book on temporal logic (Manna and Pnueli 1991) or Emerson's handbook article (Emerson 1990).

The limited popularity of the algebraic approach to temporal logic is partly due to the lack of a convincing presentation. Usually, *modal algebras* are proposed as an algebraic counterpart to temporal (or modal) logic (Tarski and Jónsson 1952; Stirling 1992). A modal algebra is a Boolean algebra with one or more unary operators that preserve its bottom element and distribute over finite disjunctions.

In our view, these presentations suffer from excess of generality. In all practical applications, the Boolean algebra is complete (in fact, considerable ingenuity is required to construct a non-complete Boolean algebra), and most finitely disjunctive operators distribute over infinite disjunctions as well. There is much to be gained from strengthening the assumptions, because every universally disjunctive operator on a complete lattice has a unique Galois adjoint. This fact greatly simplifies the theory because it allows us to *define* the 'past' operators as adjoints of the 'future' operators (whereas, in other approaches, the 'past' operators have to be *postulated*). Moreover, calculating with Galois connections is efficient and very enjoyable, as many have noticed before us.

An equally important tool that requires completeness is the fixed point theorem of Knaster and Tarski. With the aid of recursion and Galois connections, the entire zoo of operators that populate the books on linear temporal logic can be defined in terms of the humble 'next' operator. The advantages of completeness and universal disjunctivity are so significant that algebras enjoying them deserve a special name. A complete Boolean algebra with a universally disjunctive operator will be called a *Galois algebra*.

Why, then, is completeness not assumed in most logical treatments of the subject? The most compelling reason against completeness is that it is awkward to state as an axiom or inference rule. In a finitary syntactical framework, completeness can only be dealt with indirectly, for example by postulating the existence of fixed point operators. Even so, it tends to destroy decidability. Another reason for not requiring completeness in the first place is that it can be added *a posteriori* if necessary, using the Perfect Extension Theorem of Jónsson and Tarski (Tarski and Jónsson 1952), which asserts that any Boolean algebra with finitely disjunctive operators can be embedded in a complete Boolean algebra with universally disjunctive operators. For these reasons, logicians rightly reject the power of completeness. In the context of algebra, completeness is not a complicated postulate at all, and it is a necessary requirement for calculating with Galois connections and fixed points.

An intriguing aspect of our algebraic proofs is the fact that we almost never seem to require the negation operator. This suggests dropping negation from the signature of a

Galois algebra, and adding the (very few) laws that cannot be proved without it as axioms. By a number of examples we show that this ‘intuitionistic’ version of Galois algebra is quite commonplace in computing.

We conclude the introduction with an overview of the paper’s structure.

In Section 2 (Galois algebras) we introduce classical and intuitionistic Galois algebras and derive their most basic laws (those not involving fixed points).

In Section 3 (Fixed point calculus) we combine the theorem of Knaster and Tarski with the theory of Galois connections to develop a calculus of least fixed points and especially of iteration.

In Section 4 (Further temporal operators) we introduce, among others, the box and diamond operators.

In Section 5 (Confluence and linearity) we study the effect of adding additional axioms about the underlying transition relation and how to abstract from it.

In Section 6 (Linear temporal logic) we reduce all seventeen axioms of Manna and Pnueli’s logic to the existence of two Galois connections.

In Section 7 (Computation tree logic) we make a similar connection between Galois algebra and CTL.

In Section 8 (Examples) we demonstrate that Galois algebras occur naturally in computing.

In Section 9 (A reachability algorithm) we apply Galois algebra to the development of a simple algorithm for graphs.

In Section 10 (Concluding remarks) we comment on the relative merits of logical deduction and algebraic calculation.

## 2. Galois algebras

In this section we recall the definition and some basic properties of Galois connections and demonstrate their relevance for temporal logic.

### 2.1. Galois connections

For every lattice mentioned in this article we shall use the the symbols  $\sqsubseteq$ ,  $\cup$ ,  $\cap$ ,  $\top$ , and  $\perp$  to denote the lattice ordering, join, meet, top and bottom, respectively.

A partial order is called a *complete lattice* if every subset has a meet (greatest lower bound) and a join (least upper bound). A function between complete lattices is *universally disjunctive* (*conjunctive*) if it preserves all joins (meets). Every universally disjunctive function is monotonic and strict (maps  $\perp$  to  $\perp$ ). Universally conjunctive functions are also monotonic and preserve  $\top$ . We are interested in universally conjunctive or disjunctive functions because they possess a kind of inverse, called an adjoint. A pair of adjoint functions is called a *Galois connection*. Formally,  $(f, g)$  is a Galois connection between the two complete lattices  $\mathcal{F}$  and  $\mathcal{G}$  if  $f$  is a function from  $\mathcal{G}$  to  $\mathcal{F}$  and  $g$  a function from  $\mathcal{F}$  to  $\mathcal{G}$  such that

$$f.x \sqsubseteq y \quad \Leftrightarrow \quad x \sqsubseteq g.y \quad \text{for all } x \in \mathcal{G} \text{ and } y \in \mathcal{F}. \quad (1)$$

The dot in terms like  $f.x$  denotes functional application. It associates to the right, so that  $f.g.x = f.(g.x)$ . If  $(f, g)$  is a Galois connection, we say that  $f$  is a lower adjoint of  $g$  and that  $g$  is an upper adjoint of  $f$ . The following result describes the correspondence between Galois connections and universal disjointness. It is well known and the proof is straightforward.

**Proposition 2.1.1.** For any function  $f$  between complete lattices, the following are equivalent:

1.  $f$  is universally disjoint;
2.  $f$  has an upper adjoint;
3.  $f$  has precisely one upper adjoint.

Similarly, a function has a lower adjoint if and only if it is universally conjunctive, and lower adjoints are also unique. Thus universally disjoint functions are in bijective correspondence with universally conjunctive ones, and we use  $f^\flat$  and  $f^\sharp$  to denote the lower and upper adjoints of  $f$  when they exist. In other words,

$$(f, g) \text{ is a Galois connection} \iff g = f^\sharp \iff f = g^\flat.$$

We note that  $\sharp$  and  $\flat$  are anti-monotonic with respect to pointwise ordering:

$$f \subseteq g \iff g^\sharp \subseteq f^\sharp, \tag{2}$$

for any universally disjoint  $f$  and  $g$ .

There is yet another characterization of Galois connections, which is exploited in many calculations and is given by the following proposition.

**Proposition 2.1.2. (Cancellation Rule)** Suppose  $f$  and  $g$  are two monotonic functions between complete lattices. Then  $(f, g)$  is a Galois connection if and only if

$$f \circ g \subseteq id \quad \text{and} \quad id \subseteq g \circ f.$$

The following rule shows that functional composition can be extended to Galois connections.

**Proposition 2.1.3. (Composition Rule for Galois connections)** Suppose that  $(f_1, g_1)$  is a Galois connection between  $\mathcal{G}$  and  $\mathcal{F}$ , and that  $(f_2, g_2)$  is a Galois connection between  $\mathcal{H}$  and  $\mathcal{G}$ . Then  $(f_1 \circ f_2, g_2 \circ g_1)$  is a Galois connection between  $\mathcal{H}$  and  $\mathcal{F}$ . In other words, if  $f_1^\sharp$  and  $f_2^\sharp$  exist and  $f_1 \circ f_2$  is defined, then

$$(f_1 \circ f_2)^\sharp = f_2^\sharp \circ f_1^\sharp. \tag{3}$$

More detailed accounts of the theory of Galois connections can be found in Ore (1944), Everett (1944), Herrlich (1985), Melton *et al.* (1986) and Aarts (1992).

### 2.2. Classical Galois algebras

Consider a set  $M$  of states and a transition relation  $R \subseteq M \times M$  (you may think of  $(M, R)$  as a directed graph). For any  $p \subseteq M$  the *relational pre-image* of  $p$  under  $R$  is defined by

$$R \triangleleft p \stackrel{def}{=} \{l \in M \mid \exists m : (l, m) \in R \text{ and } m \in p\}. \tag{4}$$

When  $R$  is fixed it need not be mentioned and we can write  $\oplus.p$  (pronounced ‘next  $p$ ’)† instead of  $R\triangleleft p$ . Most of the time, we omit the application dot as well. In graph-theoretical terms, the set  $\oplus p$  consists of all points that have a successor in  $p$ . Now consider the equation

$$\oplus p \cap q = F, \tag{5}$$

which states that there is no arc from any  $q$ -state to any  $p$ -state. This proposition can also be expressed in terms of the converse relation  $R^\cup \stackrel{def}{=} \{(y, x) \mid (x, y) \in R\}$ . With  $\ominus q \stackrel{def}{=} R^\cup \triangleleft q$ , we can replace equation (5) by

$$p \cap \ominus q = F. \tag{6}$$

The equivalence of (5) and (6) is so fundamental that it deserves a name and a number.

**Proposition 2.2.1. (Exchange rule)** With  $\oplus$  and  $\ominus$  defined as above, we have

$$\oplus p \subseteq \neg q \iff \ominus q \subseteq \neg p. \tag{7}$$

The logical dual of  $\oplus$  is defined by  $\widetilde{\oplus} p \stackrel{def}{=} \neg \oplus \neg p$ . In the graph-theoretic interpretation,  $x$  is an element of  $\widetilde{\oplus} p$  if every successor of  $x$  is in  $p$ . Similarly,  $\widetilde{\ominus} p \stackrel{def}{=} \neg \ominus \neg p$ . With the aid of the new operators, we can restate the Exchange Rule as a Galois connection either as

$$\oplus p \subseteq q \iff p \subseteq \widetilde{\ominus} q \tag{8}$$

or as

$$\ominus p \subseteq q \iff p \subseteq \widetilde{\oplus} q. \tag{9}$$

Since each component of a Galois connection  $(f, g)$  uniquely determines its adjoint, nothing more needs saying about the relationship between  $\oplus$  and  $\ominus$ . Therefore, we give the following definition.

**Definition 2.2.2.** A *classical Galois algebra* is a complete Boolean algebra  $\mathcal{G}$  with two additional unary operators  $\oplus$  and  $\ominus$  satisfying the Exchange Rule (7).

We employ the qualifier ‘classical’ because we will consider an intuitionistic version later on. For an explanation of why completeness is required, we refer back to the discussion in the introduction.

Galois algebra inherits the *logical* duality principle of Boolean algebra: given any valid equation or inequation, we obtain another one by replacing every operator with its logical dual (and, in the case of an inequation, reversing the inclusion sign). This follows from the fact that the Exchange Rule (7) is equivalent to its logical dual.

Unlike Boolean algebra, Galois algebra enjoys a second and independent symmetry, the *time-wise* duality, which replaces each occurrence of  $\oplus$  with  $\ominus$  and *vice versa*. Thus, with every theorem we prove, we obtain three more for free.

† A remark on our choice of notation: the operator  $\oplus$  is the algebraic counterpart of the EX operator of CTL. We will later add axioms that make it more like the  $\oplus$  operator of Linear Temporal Logic, so adopting the notation of CTL (which is clumsy in the first place) would be unfortunate here. The plus sign is written inside the ‘next’ operator to stress the symmetry of past and future.

2.3. Algebraic laws

We will now derive some basic laws of Galois algebra. We are doing this for two reasons. First, we wish to introduce the reader to the spirit of proving temporal formulae by algebraic calculation. Second, we can capitalize on the work done in this section when we discover that each Galois algebra gives rise to many new Galois algebras.

All laws proved in this section hold not just for classical Galois algebras, but also for intuitionistic ones (to be defined in the next section).

We obtain our first law by instantiating Proposition 2.1.2 with the Galois connections (8) and (9).

**Proposition 2.3.1. (Cancellation Rule)** In every Galois algebra we have

$$\oplus \circ \widetilde{\ominus} \subseteq id \subseteq \widetilde{\ominus} \circ \oplus \quad \text{and} \quad \ominus \circ \widetilde{\oplus} \subseteq id \subseteq \widetilde{\oplus} \circ \ominus. \tag{10}$$

From Proposition 2.1.1 we know that  $\oplus$  and  $\ominus$  are universally disjunctive. In particular,  $\oplus F = F = \ominus F$ . By duality,  $\widetilde{\oplus}$  and  $\widetilde{\ominus}$  are universally conjunctive and preserve T.

From this it follows that all four operators are monotonic. This observation is important because monotonicity allows us to weaken an expression by weakening some subexpression; without monotonicity, inequational reasoning is impossible. We also require monotonicity for calculating with least and greatest fixed points.

We must be wary of the fact that negation is *not* monotonic. This is one reason why we want to use it as little as possible in our calculus. The following law is a crucial step towards this objective, because it relates the ‘next’ and ‘previous’ operators in a negation-free manner.

**Proposition 2.3.2. (Best-of-Both-Worlds)** If some successor (of the current state) enjoys property  $p$  and furthermore every successor enjoys  $q$ , then there must be a successor in the ‘best of both worlds’.

$$\oplus p \cap \widetilde{\oplus} q \subseteq \oplus(p \cap q) \quad \text{and} \quad \ominus p \cap \widetilde{\ominus} q \subseteq \ominus(p \cap q). \tag{11}$$

*Proof.* In the following chain of equivalences and implications we start from the demonstrandum. This is the recommended style because it is consistent with the proof’s discovery in the sense that the individual steps are natural and in most cases motivated by the desire to simplify the current expression, or in some cases, to prepare for such a simplification in the next step.

$$\begin{aligned} & \oplus p \cap \widetilde{\oplus} q \subseteq \oplus(p \cap q) \\ \Leftrightarrow & \quad \{ \text{Boolean algebra} \} \\ & \oplus p \subseteq \neg \widetilde{\oplus} q \cup \oplus(p \cap q) \\ \Leftrightarrow & \quad \{ \text{definition of } \widetilde{\oplus} \} \\ & \oplus p \subseteq \oplus \neg q \cup \oplus(p \cap q) \\ \Leftarrow & \quad \{ \oplus \text{ distributes over disjunction (and is monotonic)} \} \\ & p \subseteq \neg q \cup (p \cap q) \\ \Leftrightarrow & \quad \{ \text{complement rule of Boolean algebra} \} \\ & \text{true.} \end{aligned}$$

Of course, one can write this chain in the reverse order, deriving the demonstrandum from ‘true’. However, if we do so, then the very first step of the proof (deducing  $p \subseteq \neg q \cup (p \cap q)$  from ‘true’) comes as a surprise, like the proverbial rabbit from a conjurer’s hat, and the reader does not immediately see the reason why this specific deduction is made rather than any one of a hundred other possibilities (van Gasteren 1991).  $\square$

The Dedekind law was first known in group theory: for subsets  $O, P, Q$  of a group, one has  $OP \cap Q \subseteq O(P \cap O^{-1}Q)$  where  $O^{-1} = \{o^{-1} \mid o \in O\}$ . In relation algebra it takes the form  $O;P \cap Q \subseteq O;(P \cap O^U;Q)$ . Since these inequations have the same shape as the defining property of a modular lattice (replace ‘;’ with ‘ $\cup$ ’ and transposition with the identity), Dedekind laws are often called *modular laws*. The Dedekind rules of group theory and relation algebra are special cases of the following result, which we therefore also call a Dedekind rule.

**Proposition 2.3.3. (Dedekind Rule)** In every Galois algebra we have

$$\oplus p \cap q \subseteq \oplus(p \cap \ominus q) \quad \text{and} \quad \ominus p \cap q \subseteq \ominus(p \cap \oplus q). \tag{12}$$

*Proof.*  $\oplus p \cap q$   
 $\subseteq$  { Cancellation Rule (10) }  
 $\oplus p \cap \widetilde{\oplus} \ominus q$   
 $\subseteq$  { Best-of-Both-Worlds }  
 $\oplus(p \cap \ominus q)$ .  $\square$

The Best-of-Both-Worlds and Dedekind Rules are illustrated in the proof of the following law.

**Lemma 2.3.4.** In every Galois algebra we have  $\oplus \circ \widetilde{\ominus} \subseteq \oplus \circ \ominus$ .

*Proof.*  $\oplus \widetilde{\ominus} p$   
 $=$   
 $\oplus \widetilde{\ominus} p \cap \top$   
 $\subseteq$  { Dedekind }  
 $\oplus(\widetilde{\ominus} p \cap \ominus \top)$   
 $\subseteq$  { Best-of-Both-Worlds, monotonicity of  $\oplus$  }  
 $\oplus \ominus p$ .  $\square$

2.4. *Shunting and modus ponens*

One thesis of this article is that negation is not necessary for temporal reasoning and we avoid it wherever possible. In our experience, the intuitionistic discipline does not make proofs harder: on the contrary, it improves clarity and elegance. In keeping with this principle, we introduce implication not by its usual definition  $q \rightarrow r = r \cup \neg q$ , but by the following Galois correspondence, the so-called Shunting Rule:

$$q \cap p \subseteq r \quad \Leftrightarrow \quad p \subseteq q \rightarrow r. \tag{13}$$

Applying the Shunting Rule with  $q \rightarrow r$  in place of  $p$  yields the *Modus Ponens*:

$$q \cap (q \rightarrow r) \subseteq r.$$

To exploit the calculational properties of Galois connections involving binary operators, it is frequently necessary to keep one of the arguments fixed. Therefore we define, for any binary infix operator  $\bullet$ , the functions  $(x \bullet)$  and  $(\bullet x)$  by

$$(x \bullet).y \stackrel{def}{=} x \bullet y \quad \text{and} \quad (\bullet x).y \stackrel{def}{=} y \bullet x.$$

This process of constructing unary operators from binary ones is known as *sectioning*. For example, the Shunting Rule can be rendered as

$$(q \cap)^{\#} = (q \rightarrow).$$

A lattice that has an implication operator satisfying the Shunting Rule is called a *Heyting algebra*. Heyting algebras are to intuitionistic logic as Boolean algebras are to classical logic<sup>†</sup> (Rasiowa and Sikorski 1963; Vickers 1988). A complete lattice  $L$  is a Heyting algebra if and only if  $(p \cap)^{\#}$  exists for every  $p \in L$ , in other words if conjunction is universally disjunctive.

The proof of the following proposition shows the Shunting Rule in action.

**Proposition 2.4.1. (Distributivity over implication)** The following inequations hold in every Galois algebra.

$$\widetilde{\oplus}(p \rightarrow q) \subseteq \widetilde{\oplus}p \rightarrow \widetilde{\oplus}q \quad \text{and} \quad \oplus(p \rightarrow q) \subseteq \widetilde{\oplus}p \rightarrow \oplus q.$$

*Proof.* Shunting  $\widetilde{\oplus}p$  to the left-hand side, we rewrite the first claim to

$$\widetilde{\oplus}(p \rightarrow q) \cap \widetilde{\oplus}p \subseteq \widetilde{\oplus}q,$$

which follows from conjunctivity of  $\widetilde{\oplus}$  and *modus ponens*. To prove the second inequation, we shunt  $\widetilde{\oplus}p$  to the left-hand side and obtain

$$\oplus(p \rightarrow q) \cap \widetilde{\oplus}p \subseteq \oplus q,$$

which follows from Best-of-Both-Worlds and *modus ponens*. □

### 2.5. Intuitionistic Galois algebras

The only places so far where we have needed negation were in rewriting the Exchange Rule (7) to the Galois correspondences (8) and (9) and in proving the Best-of-Both-Worlds law. We shall now eliminate the need for negation entirely by adopting these laws as axioms.

**Definition 2.5.1.** A structure  $(\mathcal{G}, \oplus, \ominus)$  is an *intuitionistic Galois algebra* if

1.  $\mathcal{G}$  is a complete Heyting algebra;
2.  $\oplus$  and  $\ominus$  are universally disjunctive operators on  $\mathcal{G}$ ;
3. the following Best-of-Both-Worlds laws hold:

$$\oplus p \cap \ominus^{\#} q \subseteq \oplus(p \cap q) \quad \text{and} \quad \ominus p \cap \oplus^{\#} q \subseteq \ominus(p \cap q).$$

<sup>†</sup> Heyting algebras are also known as duals of *Brouwer lattices* in the sense that  $(\mathcal{G}, \subseteq)$  is a Heyting algebra if and only if  $(\mathcal{G}, \supseteq)$  is a Brouwer lattice.



As before, we will write  $\widetilde{\oplus}$  in place of  $\ominus^\#$  and  $\widetilde{\ominus}$  instead of  $\oplus^\#$ . From the results in the previous section it follows that every classical Galois algebra is also an intuitionistic Galois algebra. Therefore, we shall omit the qualifier ‘intuitionistic’ unless we want to stress that the underlying lattice need not be a Boolean algebra.

The following lemma is useful because it allows us to prove that a structure is a Galois algebra without actually computing the  $\widetilde{\oplus}$  and  $\widetilde{\ominus}$  operators.

**Lemma 2.5.2.** Suppose  $\oplus$  and  $\ominus$  are universally disjunctive operators on some Heyting algebra  $\mathcal{G}$  that satisfy the Dedekind laws (12). Then  $(\mathcal{G}, \oplus, \ominus)$  is a Galois algebra.

*Proof.* We have to prove the Best-of-Both-Worlds laws. By symmetry, it is enough to check one of them.

$$\begin{aligned} & \oplus p \cap \ominus^\# q \\ \subseteq & \quad \{ \text{Dedekind} \} \\ & \oplus(p \cap \ominus \ominus^\# q) \\ \subseteq & \quad \{ \text{Cancellation Rule 2.1.2} \} \\ & \oplus(p \cap q). \end{aligned} \quad \square$$

The following table lists the laws we have established (or postulated) for either kind of Galois algebra.

$\oplus \widetilde{\ominus} p \subseteq p$	$\subseteq$	$p$	$\subseteq$	$\widetilde{\oplus} \ominus p$
$\oplus F = F$	$=$	$F$	$=$	$\ominus F$
$\oplus(p \cup q) = \oplus p \cup \oplus q$	$=$	$\oplus p \cup \oplus q$		
$\oplus p \cap q \subseteq \oplus(p \cap \ominus q)$	$\subseteq$	$\oplus(p \cap \ominus q)$		
$\widetilde{\oplus} p \cap \widetilde{\oplus} q \subseteq \widetilde{\oplus}(p \cap q)$	$\subseteq$	$\widetilde{\oplus}(p \cap q)$		
$\widetilde{\oplus}(p \rightarrow q) \subseteq \widetilde{\oplus} p \rightarrow \widetilde{\oplus} q$	$\subseteq$	$\widetilde{\oplus} p \rightarrow \widetilde{\oplus} q$		
$\oplus(p \rightarrow q) \subseteq \oplus p \rightarrow \oplus q$	$\subseteq$	$\oplus p \rightarrow \oplus q$		

**Table 1.**

Adding the time-wise and (in the classical case) logical duals of these laws we obtain a wealth of useful laws – all of which are derived from the single Exchange Rule.

One advantage of algebra over logic is the ease with which new algebras can be defined from old, for example by forming direct products and function spaces. The following proposition shows how functional composition can be extended to an operator on Galois algebras. Note that the ‘previous’ operators are composed in reverse order.

**Proposition 2.5.3. (Composition Rule for Galois algebras)** If  $G_1 = (\mathcal{G}, \oplus_1, \ominus_1)$  and  $G_2 = (\mathcal{G}, \oplus_2, \ominus_2)$  are Galois algebras, then so is  $G_1 \circ G_2 \stackrel{def}{=} (\mathcal{G}, \oplus_1 \circ \oplus_2, \ominus_2 \circ \ominus_1)$ .

*Proof.* Let  $\oplus \stackrel{def}{=} \oplus_1 \circ \oplus_2$  and  $\ominus \stackrel{def}{=} \ominus_2 \circ \ominus_1$ . By the Composition Rule 2.1.3, the functions  $\oplus$  and  $\ominus$  have upper adjoints. It remains to check the Best-of-Both-Worlds laws. By symmetry, we need only look at one of them.

$$\begin{aligned} & \oplus p \cap \ominus^\# q \\ = & \quad \{ \text{definitions, } (\ominus_2 \circ \ominus_1)^\# = \ominus_1^\# \circ \ominus_2^\# \} \\ & \oplus_1 \oplus_2 p \cap \ominus_1^\# \ominus_2^\# q \end{aligned}$$

$$\begin{aligned}
 &\subseteq \{ \text{Best-of-Both-Worlds in } (\mathcal{G}, \oplus_1, \ominus_1) \} \\
 &\quad \oplus_1(\oplus_2 p \cap \ominus_2 q) \\
 &\subseteq \{ \text{Best-of-Both-Worlds in } (\mathcal{G}, \oplus_2, \ominus_2) \} \\
 &\quad \oplus_1 \oplus_2(p \cap q) \\
 &= \{ \text{definition of } \oplus \} \\
 &\quad \oplus(p \cap q). \quad \square
 \end{aligned}$$

2.6. Examples

We have proved that every classical Galois algebra satisfies the axioms of intuitionistic Galois algebra. The following examples show that the converse is not true. A number of examples that are more specifically relevant to computing will be given in Section 8.

*Sections of Conjunction.* The following proposition shows a simple way to construct Galois algebras from any complete Heyting algebra. Trivial though they are, these are handy building blocks and we will use them for defining the ‘since’ and ‘until’ operators of temporal logic.

**Proposition 2.6.1.** Assume that  $\mathcal{G}$  is a complete Heyting algebra and  $x \in \mathcal{G}$ . Then  $(\mathcal{G}, (x \cap), (x \cap))$  is an intuitionistic Galois algebra.

*Proof.* By the Shunting Rule,  $(x \cap)$  has an upper adjoint, namely  $(x \rightarrow)$ . The Best-of-Both-Worlds Rule

$$(x \cap).p \cap (x \rightarrow).q \subseteq (x \cap).(p \cap q)$$

follows from *modus ponens*. □

*Predecessor and Successor.* Let  $\mathcal{G} \stackrel{\text{def}}{=} \mathbb{Z} \cup \{-\infty, +\infty\}$  with the natural ordering. Then  $\mathcal{G}$  is a complete Heyting algebra, with operations defined by  $p \cup q \stackrel{\text{def}}{=} (p \max q)$  and  $p \cap q \stackrel{\text{def}}{=} (p \min q)$  and

$$p \rightarrow q \stackrel{\text{def}}{=} \begin{cases} q & \text{if } q < p \\ +\infty & \text{if } q \geq p. \end{cases}$$

Now choose  $m, n \in \mathbb{Z}$  with  $m \leq n$  (note the asymmetry) and define

$$\oplus p = p + n \quad \ominus p = p - m \quad \widetilde{\oplus} p = p + m \quad \widetilde{\ominus} p = p - n, \tag{14}$$

for all  $p \in \mathcal{G}$  (with the convention that adding or subtracting a finite number has no effect on an infinity). Then we have the following proposition, the verification of which we leave to the reader.

**Proposition 2.6.2.** With  $\oplus$  and  $\ominus$  defined by (14),  $(\mathcal{G}, \oplus, \ominus)$  is a Galois algebra.

We conclude that the ‘next’ and ‘previous’ operators do not, in general, determine each other (though they do in *classical* Galois algebras).

*Greatest common divisor and least common multiple.* The following example is based on the multiplicative structure of the natural numbers. For nonnegative integers  $n$  and  $m$  define

$$n \cup m = \text{gcd}(n, m) \quad \text{and} \quad n \cap m = \text{lcm}(n, m), \tag{15}$$

where it is understood that  $\text{lcm}(0,0) = \text{gcd}(0,0) = 0$ . With these operations, the set of all nonnegative numbers forms a complete Heyting algebra (with bottom element 0 and top element 1). Let

$$\widetilde{\oplus}n = \widetilde{\ominus}n = n^2 \quad \text{and} \quad \oplus n = \ominus n = \sqrt{n}, \tag{16}$$

where  $\sqrt{0} = 0$  and  $\sqrt{n} = m$  if  $n > 0$ , and  $m$  is the largest number such that  $m^2$  divides  $n$ .

**Proposition 2.6.3.** With the operators defined by (15) and (16), the set of nonnegative integers forms an intuitionistic Galois algebra.

### 3. Fixed point calculus

Recursion is the most powerful tool for defining new operators in Galois algebras, and to determine their properties we must calculate with fixed points. Naturally, the scope of fixed point calculus is much broader than just Galois algebra. This alone might not justify our treating it in full generality here, but as is often the case in mathematics, the general approach is actually simpler and more elegant than one tailored to the specific theory at hand. Therefore, we introduce the notion of one lattice operating on another, a concept that enables us to treat various incarnations of calculation rules for fixed points in a uniform manner.

Several of the results in this section are well known, although often in less general form. However, most of the proofs are new and shorter than those appearing in the literature.

#### 3.1. Preliminaries

Suppose  $f$  is a monotonic function on a complete lattice  $G$ . Then the famous theorem of Knaster and Tarski provides us with two crucial bits of information. First, it assures us that  $f$  has a unique least fixed point, denoted  $\mu f$ , and second, it provides a rule for establishing upper bounds of  $\mu f$ : for every  $r \in \mathcal{G}$  we have

$$\mu f \subseteq r \quad \Leftrightarrow \quad \exists q : f.q \subseteq q \subseteq r. \tag{17}$$

This rule is better known in the following form, which states that  $\mu f$  is the least solution of the inequation  $f.r \subseteq r$ .

$$\mu f \subseteq r \quad \Leftarrow \quad f.r \subseteq r. \tag{18}$$

It is easy to see that (17) and (18) are equivalent. The latter formula, which we will refer to as ‘Induction Rule’, has the advantage of simplicity, but in certain calculations (17) works better, because it is an equivalence. A notable example is the proof of the Star Adjunction Theorem in Section 3.5. By symmetry,  $f$  also has a unique greatest fixed point  $\nu f$  satisfying

$$p \subseteq \nu f \quad \Leftrightarrow \quad \exists q : p \subseteq q \subseteq f.q \tag{19}$$

and

$$p \subseteq \nu f \quad \Leftarrow \quad p \subseteq f.p. \tag{20}$$

Because it is frequently inconvenient to assign names to the functions occurring in fixed point expressions we allow the notations  $\mu_x(f.x)$  and  $\nu_x(f.x)$  instead of  $\mu f$  and  $\nu f$ . For

example,  $\mu_x(p \cup \bigoplus x)$  denotes the least fixed point of the (anonymous) function that maps  $x$  to  $p \cup \bigoplus x$ .

The single most useful theorem of the fixed point calculus is the  $\mu$ -Fusion Rule, also known as Transfer Lemma, which provides a way to compute the image of a least fixed point under the lower adjoint of a Galois connection.

**Theorem 3.1.1. ( $\mu$ -Fusion)** Suppose  $E$  and  $F$  are complete lattices,  $e : E \mapsto E$  and  $f : F \mapsto F$  are monotonic and  $\sigma : E \mapsto F$  is universally disjunctive. Then we have the implications

$$\begin{aligned} \mu f \subseteq \sigma.\mu e &\Leftarrow f \circ \sigma \subseteq \sigma \circ e \\ \sigma.\mu e \subseteq \mu f &\Leftarrow \sigma \circ e \subseteq f \circ \sigma, \end{aligned}$$

and hence

$$\sigma.\mu e = \mu f \quad \Leftarrow \quad \sigma \circ e = f \circ \sigma.$$

*Remark.* The above is not the strongest possible version of the  $\mu$ -Fusion Theorem but it has the merit of being provable by almost syntactical rewriting. The following proof, which we owe to Roland Backhouse, is a nice exercise on Galois connections.

*Proof.* The first implication is a straightforward consequence of the Induction Rule (18) and we leave it as an exercise. To prove the second inequation, we assume  $\sigma.e.x \subseteq f.\sigma.x$  holds for all  $x$  (recall that the application dot associates to the right). Then we have

$$\begin{aligned} &\sigma.\mu e \subseteq \mu f \\ \Leftrightarrow &\quad \{ \text{Galois connection} \} \\ &\mu e \subseteq \sigma^\#.\mu f \\ \Leftarrow &\quad \{ \text{Induction Rule (18)} \} \\ &e.\sigma^\#.\mu f \subseteq \sigma^\#.\mu f \\ \Leftrightarrow &\quad \{ \text{Galois connection} \} \\ &\sigma.e.\sigma^\#.\mu f \subseteq \mu f \\ \Leftrightarrow &\quad \{ \text{assumption} \} \\ &f.\sigma.\sigma^\#.\mu f \subseteq \mu f \\ \Leftarrow &\quad \{ \mu f = f.\mu f, \text{ monotonicity of } f \} \\ &\sigma.\sigma^\#.\mu f \subseteq \mu f \\ \Leftrightarrow &\quad \{ \text{Galois connection} \} \\ &\sigma^\#.\mu f \subseteq \sigma^\#.\mu f. \quad \square \end{aligned}$$

The following theorem, which does not seem to have been noticed before, is a nifty application of the  $\mu$ -Fusion Rule. Its importance for this paper is its close relation to the Best-of-Both-Worlds Rule (take  $h = f$  in the antecedent). Because of its relevance to the convergence (termination) of recursive programs (to be explained in a separate paper), we name it the Convergence Rule.

**Theorem 3.1.2. (Convergence Rule)** If  $f$ ,  $g$ , and  $h$  are monotonic functions on a complete Heyting algebra, we have

$$\nu f \cap \mu g \subseteq \mu h \quad \Leftarrow \quad \forall x, y : f.x \cap g.y \subseteq h.(x \cap y). \tag{21}$$

*Proof.*  $vf \cap \mu g \subseteq \mu h$   
 $\Leftarrow \{ \mu\text{-Fusion, } (vf \cap) \text{ is universally disjunctive} \}$   
 $\forall y \in \mathcal{G} : vf \cap g.y \subseteq h.(vf \cap y)$   
 $\Leftrightarrow \{ vf = f.vf \}$   
 $\forall y \in \mathcal{G} : f.vf \cap g.y \subseteq h.(vf \cap y)$   
 $\Leftarrow \{ \text{take } x = vf \}$   
 $\forall x, y \in \mathcal{G} : f.x \cap g.y \subseteq h.(x \cap y) . \quad \square$

3.2. Operator lattices – definition

Consider the set  $Mon(\mathcal{G})$  of all monotonic functions on a complete lattice  $\mathcal{G}$ . Then  $Mon(\mathcal{G})$  is itself a complete lattice and every element of  $Mon(\mathcal{G})$  is a unary operator on  $\mathcal{G}$ . In this sense, the lattice  $Mon(\mathcal{G})$  operates on the lattice  $\mathcal{G}$ . This situation is so fundamental to the study of fixed points that it deserves an axiomatization.

To minimize the temptation of reasoning from the concrete model, rather than the proposed axioms, it is advisable to choose new symbols for the axiomatic theory. Therefore we rename the base set and operators of the structure  $Mon(\mathcal{G})$  as follows:

- $\mathcal{F} \stackrel{\text{def}}{=} Mon(\mathcal{G})$
- Application: define  $\cdot : \mathcal{F} \times \mathcal{G} \rightarrow \mathcal{G}$  by  $f \cdot g = f.g$
- Composition: define  $\diamond : \mathcal{F} \times \mathcal{F} \rightarrow \mathcal{F}$  by  $f_1 \diamond f_2 = f_1 \circ f_2 .$

Here are some properties of this structure:

1.  $\mathcal{F}$  is a complete lattice.
2.  $(\mathcal{F}, \diamond, id)$  is a monoid.
3. The composition operator  $\diamond$  is monotonic in its right and universally disjunctive in its left argument.
4. Composition and application associate:

$$(f_1 \diamond f_2) \cdot g = f_1 \cdot (f_2 \cdot g).$$

5.  $id \cdot g = g .$
6. The application operator  $\cdot$  is monotonic in its right and universally disjunctive in its left argument.

We abstract now from the specific model  $(\mathcal{F}, \diamond, \cdot) = (Mon(\mathcal{G}), \circ, \cdot)$  and propose the following definition.

**Definition 3.2.1.** Suppose that  $\mathcal{G}$  is a complete lattice. The structure  $(\mathcal{F}, \diamond, \cdot)$  operates on  $\mathcal{G}$  if Axioms 1–6 hold. A structure  $(\mathcal{F}, \diamond)$  that satisfies Axioms 1–3 is called an *operator lattice*.

Monoids operating on sets (Axioms 2, 4, and 5) play a very prominent role in algebra (transformation groups). Our definition simply extends this concept by taking the underlying lattice structure into account.

In Backhouse *et al.* (1992), operator lattices are called *semi-regular algebras*. Our choice of name is justified by the following lemma.

**Lemma 3.2.2.** If  $(\mathcal{F}, \diamond)$  is an operator lattice, then  $(\mathcal{F}, \diamond, \diamond)$  operates on  $\mathcal{F}$ .

3.3. Operator lattices – models

We have already encountered two models for the axiom system 1–6 above, namely

- $\mathcal{G}$  is a complete lattice and  $(\mathcal{F}, \diamond, \cdot) = (\text{Mon}(\mathcal{G}), \circ, \cdot)$
- $(\mathcal{G}, \diamond)$  is an operator lattice and  $(\mathcal{F}, \diamond, \cdot) = (\mathcal{G}, \diamond, \diamond)$ .

The second example covers the cases where  $(\mathcal{G}, \diamond)$  is any regular algebra, such as the set of all binary relations on a given set, but also where  $(\mathcal{G}, \diamond) = (\text{Mon}(\mathcal{H}), \circ)$  for some complete lattice  $\mathcal{H}$ . Note that this is different from the first example.

The relational calculus provides yet another model. Let  $M$  be a set,  $\mathcal{G}$  the powerset of  $M$  and  $\mathcal{F}$  the set of all binary relations on  $M$ . We use a semicolon to denote relational composition:

$$f_1; f_2 = \{(k, m) \mid \exists l : (k, l) \in f_1 \text{ and } (l, m) \in f_2\}.$$

The other operator is the relational image operator  $\triangleleft$ , which we have defined by

$$f \triangleleft g = \{l \in M \mid \exists m : (l, m) \in f \text{ and } m \in g\}.$$

Then  $(\mathcal{F}, ;, \triangleleft)$  operates on  $\mathcal{G}$ .

3.4. Iteration and tail recursion

Suppose that  $(\mathcal{F}, \diamond)$  is an operator lattice. Then we define an iteration operator on  $\mathcal{F}$  by

$$f^* \stackrel{\text{def}}{=} \mu_x(\text{id} \cup f \diamond x). \tag{22}$$

If  $\mathcal{F}$  is the set of all regular languages over a given alphabet, iteration is just the Kleene star operator. Or, if  $(\mathcal{F}, \diamond)$  is the set of all binary relations on a given set and  $\diamond$  is relational composition, then  $f^*$  is the reflexive and transitive closure of  $f$ .

A basic, yet powerful, calculation rule for iteration is the Tail Recursion Rule. It is so named because it is often used to replace procedures in so-called tail-recursive form by while loops.

**Theorem 3.4.1. (Tail Recursion Rule – general form)** Suppose  $\mathcal{G}$  is a complete lattice and  $(\mathcal{F}, \diamond, \cdot)$  operates on  $\mathcal{G}$ . Then we have, for all  $f \in \mathcal{F}$  and  $g \in \mathcal{G}$ ,

$$f^* \cdot g = \mu_x(g \cup f \cdot x). \tag{23}$$

*Proof.*  $f^* \cdot g = \mu_x(g \cup f \cdot x)$

- $\Leftrightarrow$  { sectioning, definition of  $f^*$  }
- $(\cdot g) \cdot \mu_x(\text{id} \cup f \diamond x) = \mu_x(g \cup f \cdot x)$
- $\Leftarrow$  {  $\mu$ -fusion,  $(\cdot g)$  being disjunctive by Axiom 6 }
- $\forall x \in \mathcal{F} : (\text{id} \cup f \diamond x) \cdot g = g \cup f \cdot (x \cdot g)$
- $\Leftrightarrow$  { distributivity (Axiom 6) and  $\text{id} \cdot g = g$  (Axiom 5) }
- $\forall x \in \mathcal{F} : g \cup (f \diamond x) \cdot g = g \cup f \cdot (x \cdot g)$
- $\Leftrightarrow$  { associativity of  $\diamond$  with  $\cdot$  (Axiom 4) }

true. □

Applying the Tail Recursion Rule to an operator lattice operating on itself – the situation described in Lemma 3.2.2 – yields the following corollary.

**Corollary 3.4.2. (Tail Recursion Rule – operator version)** Assume that  $(\mathcal{F}, \diamond)$  is an operator lattice. Then we have, for all  $e, f \in \mathcal{F}$ ,

$$e^* \diamond f = \mu_x(f \cup e \diamond x).$$

We obtain yet another version of the Tail Recursion Rule by instantiating  $(\mathcal{F}, \diamond, \cdot)$  to  $(Mon(\mathcal{G}), \circ, \cdot)$ .

**Corollary 3.4.3. (Tail Recursion Rule – functional version)** Assume that  $\mathcal{G}$  is a complete lattice and  $f$  is a monotonic function on  $\mathcal{G}$ . Then we have

$$f^*.y = \mu_x(y \cup f.x) \quad \text{for every } y \in \mathcal{G}.$$

All three versions of the Tail Recursion Rule will be applied in the sequel.

### 3.5. The Star Adjunction Theorem

Suppose that  $f$  is a universally disjunctive function on a complete lattice. Then we may ask ourselves if the iterated function  $f^*$  is also universally disjunctive. The best way to answer this question is by calculating the upper adjoint of  $f^*$ .

$$\begin{aligned} f^*.p \subseteq r & \\ \Leftrightarrow & \quad \{ \text{Tail Recursion, functional version} \} \\ & \mu_x(p \cup f.x) \subseteq r \\ \Leftrightarrow & \quad \{ \text{Knaster–Tarski (17)} \} \\ \exists q : p \cup f.q \subseteq q \subseteq r & \\ \Leftrightarrow & \\ \exists q : (p \subseteq q \subseteq r) \wedge (f.q \subseteq q) & \\ \Leftrightarrow & \quad \{ \text{Galois connection} \} \\ \exists q : (p \subseteq q \subseteq r) \wedge (q \subseteq f^\#.q) & \\ \Leftrightarrow & \\ \exists q : p \subseteq q \subseteq r \cap f^\#.q & \\ \Leftrightarrow & \quad \{ \text{Knaster–Tarski (19)} \} \\ p \subseteq v_x(r \cap f^\#.x). & \end{aligned}$$

Thus we have proved the following result.

**Theorem 3.5.1. (Star Adjunction)** Assume that  $f$  is a universally disjunctive function on a complete lattice. Then  $f^*$  is universally disjunctive and its upper adjoint is given by

$$f^{*\#}.r = v_x(r \cap f^\#.x). \tag{24}$$

The Star Adjunction can be rendered even more nicely by introducing a name for the Kleene star’s dual. Define the interior  $f^\circ$  of  $f$  as

$$f^\circ = v_x(id \cap f \circ x).$$

Then the upper adjoint of  $f^*$  equals the interior of  $f^\#$  (provided that  $f^\#$  exists).

Although it never seems to have appeared in print, the Star Adjunction Theorem is known in the community. There exist handwritten proofs by Dijkstra (who attributes it to

Scholten), by Backhouse and by van der Woude. However, both the name and the above derivation of this result are new.

### 3.6. The Star Fusion Theorem

Suppose that  $(\mathcal{F}, \diamond, \cdot)$  operates on the complete lattice  $\mathcal{G}$ . Associated with every  $f \in \mathcal{F}$  is the function  $apply.f \stackrel{def}{=} (f \cdot)$  from  $\mathcal{G}$  to  $\mathcal{G}$ . Both  $f$  and  $apply.f$  may be iterated. The Star Fusion Theorem states that application and iteration commute.

$$* \circ apply = apply \circ *. \tag{25}$$

In other words, we have, for every  $f \in \mathcal{F}$ ,

$$(f \cdot)^* = (f^* \cdot) \tag{26}$$

*Proof.* For every  $g \in \mathcal{G}$  we have

$$\begin{aligned} & (f \cdot)^*.g \\ = & \quad \{ \text{Tail Recursion, functional version} \} \\ & \mu_x(g \cup (f \cdot).x) \\ = & \quad \{ \text{sectioning} \} \\ & \mu_x(g \cup f \cdot x) \\ = & \quad \{ \text{Tail Recursion, general form} \} \\ & f^*.g \\ = & \quad \{ \text{sectioning} \} \\ & (f^* \cdot).g \end{aligned} \quad \square$$

The Star Fusion Rule appears in Backhouse *et al.* (1992) for the special case where  $(\mathcal{G}, \diamond)$  is an operator lattice (called a semi-regular algebra there) and  $(\mathcal{F}, \diamond, \cdot) = (\mathcal{G}, \diamond, \diamond)$ , in which case the conclusion is  $(g \diamond)^* = (g^* \diamond)$  for every  $g \in \mathcal{G}$ .

### 3.7. Further fixed point rules

If  $d$  commutes with  $f$ , then  $d$  commutes with every power  $f^i$  as well. Assuming that composition with  $d$  is universally disjunctive, it is therefore to be expected that  $d$  should commute with  $f^*$  (which is something like the disjunction of all powers of  $f$ ). The following result is a useful generalization of this simple ‘leapfrog’ rule.

**Theorem 3.7.1. (Leapfrog Rule)** Suppose that  $(\mathcal{F}, \diamond)$  is an operator lattice and  $d, e, f \in \mathcal{F}$ . Assume that  $(d \diamond)$  is universally disjunctive. Then we have

$$d \diamond f^* = e^* \diamond d \quad \Leftarrow \quad d \diamond f = e \diamond d. \tag{27}$$

*Proof.*  $d \diamond f^* = e^* \diamond d$

$$\begin{aligned} \Leftrightarrow & \quad \{ \text{definition of } f^*, \text{ Tail Recursion (operator version)} \} \\ & d \diamond \mu_x(id \cup f \diamond x) = \mu_x(d \cup e \diamond x) \\ \Leftarrow & \quad \{ \mu\text{-Fusion, using the fact that } (d \diamond) \text{ is universally disjunctive} \} \\ & \forall x \in \mathcal{F} : d \diamond (id \cup f \diamond x) = d \cup e \diamond d \diamond x \end{aligned}$$



$$\begin{aligned} &\Leftrightarrow \{ (d \diamond) \text{ is disjunctive, } d \diamond id = d \} \\ &\quad \forall x \in \mathcal{F} : d \cup d \diamond f \diamond x = d \cup e \diamond d \diamond x \\ &\Leftarrow \\ &\quad d \diamond f = e \diamond d. \quad \square \end{aligned}$$

By definition, the reflexive and transitive closure  $f^*$  of a relation  $f$  is transitive, that is,  $f^*; f^* \subseteq f^*$ . Since  $f^*$  contains the identity relation, the converse inclusion is also true. The following proposition generalizes this fact to arbitrary operator lattices.

**Proposition 3.7.2. (Transitivity Rule)** For any element  $f$  of an arbitrary operator lattice  $(\mathcal{F}, \diamond)$  we have  $f^* \diamond f^* = f^*$ .

$$\begin{aligned} &\textit{Proof. } f^* \diamond f^* = f^* \\ &\Leftrightarrow \{ id \subseteq f^* \} \\ &\quad f^* \diamond f^* \subseteq f^* \\ &\Leftrightarrow \{ \text{Tail Recursion, operator version} \} \\ &\quad \mu_x(f^* \cup f \diamond x) \subseteq f^* \\ &\Leftarrow \{ \text{Induction Rule (18)} \} \\ &\quad f^* \cup f \diamond f^* \subseteq f^* \\ &\Leftarrow \\ &\quad f^* \text{ solves the equation } x = id \cup f \diamond x. \quad \square \end{aligned}$$

Suppose the operators  $e$  and  $f$  absorb each other, in the sense that  $e \diamond f \subseteq id$ . Then repeated absorptions should reduce everything of type  $a^i \diamond b^j$  to a composition of only  $a$ 's or only  $b$ 's, depending on which kind there were more of to start with. Therefore the following theorem is to be expected.

**Theorem 3.7.3. (Star Absorption)** Suppose  $(\mathcal{F}, \diamond)$  is an operator lattice and  $e, f \in \mathcal{F}$ . Assume, furthermore, that  $(e \diamond)$  is disjunctive. Then we have

$$e^* \diamond f^* = e^* \cup f^* \quad \Leftarrow \quad e \diamond f \subseteq id. \quad (28)$$

$$\begin{aligned} &\textit{Proof. } e^* \diamond f^* = e^* \cup f^* \\ &\Leftrightarrow \{ id \subseteq e^* \text{ and } id \subseteq f^* \} \\ &\quad e^* \diamond f^* \subseteq e^* \cup f^* \\ &\Leftrightarrow \{ \text{Tail Recursion, operator version} \} \\ &\quad \mu_x(f^* \cup e \diamond x) \subseteq e^* \cup f^* \\ &\Leftarrow \{ \text{Induction Rule (18), dropping the disjunct } f^* \text{ from the lhs} \} \\ &\quad e \diamond (e^* \cup f^*) \subseteq e^* \cup f^* \\ &\Leftrightarrow \{ (e \diamond) \text{ is disjunctive and } e \diamond e^* \subseteq e^* \} \\ &\quad e \diamond f^* \subseteq e^* \cup f^* \\ &\Leftrightarrow \{ f^* = id \cup f \diamond f^* \text{ and } (e \diamond) \text{ is disjunctive} \} \\ &\quad e \cup e \diamond f \diamond f^* \subseteq e^* \cup f^* \\ &\Leftarrow \\ &\quad e \diamond f \subseteq id. \quad \square \end{aligned}$$

4. Further temporal operators

The expressions  $\oplus p$  and  $\widetilde{\oplus} p$  depend only on states that are just one moment removed from the present; for this reason  $\oplus$ ,  $\widetilde{\oplus}$ , and their past counterparts are sometimes called *immediate* operators. In contrast, *non-immediate* operators, such as  $\diamond$ ,  $\boxplus$  and *until* construct expressions that refer to arbitrarily distant points in time. With the aid of iteration, the non-immediate operators can be *defined* in terms of the immediate ones (whereas in most logical treatments their existence is *postulated*). As a consequence, we can study their properties within the fixed point calculus.

4.1. Diamonds and boxes

We have seen in Section 2.2 that every binary relation  $R$  on a set  $M$  gives rise to a ‘next’ operator on the powerset of  $M$ , defined by

$$\oplus p \stackrel{def}{=} R \triangleleft p. \tag{29}$$

In this model, the set  $\oplus p$  consists of all elements of  $M$  that have a one-step successor in  $p$ . A related object of interest is the set of all points from which an element in  $p$  can be reached in some *arbitrary* number of steps. This set is denoted  $\diamond p$  (pronounced ‘eventually  $p$ ’) and can be defined in terms of the reflexive and transitive closure of  $R$ :

$$\diamond p \stackrel{def}{=} R^* \triangleleft p. \tag{30}$$

Definitions (29) and (30) can be rewritten as  $\oplus = (R \triangleleft)$  and  $\diamond = (R^* \triangleleft)$ , respectively. By the Star Fusion Theorem 3.6, we have  $(R^* \triangleleft) = (R \triangleleft)^*$ . Therefore, (29) and (30) imply

$$\diamond = \oplus^*. \tag{31}$$

The last equation may be used for defining an ‘eventually’ operator in an arbitrary Galois algebra. In this sense, (31) is more general than (30).

We notice that (30) may be seen as an instance of (29). Since a  $\oplus$  operator defined by (29) satisfies the axioms of Galois algebra, the same must be true for a  $\diamond$  operator defined by (30). The following theorem shows that the generalization from (30) to (31) preserves the analogy between  $\oplus$  and  $\diamond$ .

**Theorem 4.1.1. (Iteration theorem of Galois algebra)** Assume that  $(\mathcal{G}, \oplus, \ominus)$  is a Galois algebra. Then so is  $(\mathcal{G}, \oplus^*, \ominus^*)$ .

*Proof.* We know from the Star Adjunction Theorem 3.5 that  $\oplus^*$  and  $\ominus^*$  have upper adjoints. Thus we need only verify the Best-of-Both-Worlds laws. By symmetry, it is sufficient to prove just one of them, say

$$\oplus^*.p \cap \ominus^{*\#}.q \subseteq \oplus^*. (p \cap q).$$

The proof is as follows:

$$\begin{aligned} & \oplus^*.p \cap \ominus^{*\#}.q \subseteq \oplus^*. (p \cap q) \\ \Leftrightarrow & \quad \{ \text{Tail Recursion 3.4.3 and Star Adjunction 3.5} \} \\ & \mu_x(p \cup \oplus x) \cap \nu_y(q \cap \ominus^{*\#}.y) \subseteq \mu_z((p \cap q) \cup \oplus z) \end{aligned}$$

$$\begin{aligned} &\Leftarrow \{ \text{Convergence Rule 3.1.2} \} \\ &\forall x, y \in \mathcal{G} : (p \cup \oplus x) \cap (q \cap \ominus^\# y) \subseteq (p \cap q) \cup \oplus(x \cap y) \\ &\Leftarrow \{ \text{Boolean algebra} \} \\ &\forall x, y \in \mathcal{G} : \oplus x \cap \ominus^\# y \subseteq \oplus(x \cap y) . \end{aligned}$$

The last line in this calculation is precisely the Best-of-Both-Worlds law of the Galois algebra  $(\mathcal{G}, \oplus, \ominus)$ . □

Throughout the remainder of this paper we shall use the abbreviations

$$\diamond = \oplus^* \quad \diamond = \ominus^* \quad \boxplus = \diamond^\# \quad \boxminus = \diamond^\# . \tag{32}$$

The Iteration Theorem allows us to turn all the calculation rules for the round operators into analogous rules for the angular ones. Thus the following laws hold in every Galois algebra  $(\mathcal{G}, \oplus, \ominus)$ .

$\diamond \boxminus p \subseteq p \subseteq \boxplus \diamond p$
$\diamond F = F = \diamond F$
$\diamond(p \cup q) = \diamond p \cup \diamond q$
$\diamond p \cap q \subseteq \diamond(p \cap \diamond q)$
$\diamond p \cap \boxplus q \subseteq \diamond(p \cap q)$
$\boxplus(p \rightarrow q) \subseteq \boxplus p \rightarrow \boxplus q$
$\diamond(p \rightarrow q) \subseteq \boxplus p \rightarrow \diamond q$

**Table 2.**

In the special case where the underlying lattice  $\mathcal{G}$  is Boolean, we also obtain that  $\boxplus p = \neg \diamond \neg p$  and  $\boxminus p = \neg \diamond \neg p$ . In standard treatments of temporal logic these equations serve as *definitions* of the box operators. Departing from this tradition we have introduced  $\boxplus$  and  $\boxminus$  as upper adjoints of  $\diamond$  and  $\diamond$ , which also works in the absence of a negation operator.

Unlike the round operators,  $\diamond$  and  $\boxplus$  are idempotent,

$$\diamond \circ \diamond = \diamond \quad \text{and} \quad \boxplus \circ \boxplus = \boxplus . \tag{33}$$

The first of these equations is a special case of the Transitivity Rule 3.7.2. With the Composition Rule for Galois Connections 2.1.3, the second one is then proved as follows:  $\boxplus \circ \boxplus = \diamond^\# \circ \diamond^\# = (\diamond \circ \diamond)^\# = \diamond^\# = \boxplus$ . Moreover,

$$\diamond \circ \oplus = \oplus \circ \diamond \quad \text{and} \quad \boxplus \circ \widetilde{\oplus} = \widetilde{\oplus} \circ \boxplus . \tag{34}$$

The first equation follows from the Leapfrog Rule 3.7.1, and the second one from the first and the Composition Rule. Analogous rules apply to the past operators.

4.2. Induction

In the calculational approach we perform induction by appealing to the Knaster–Tarski Theorem or to one of its corollaries, such as the Tail Recursion Rule or the Star Adjunction Theorem. For example, the functional version 3.4.3 of the Tail Recursion Rule yields

$$\diamond p = \mu_x(p \cup \oplus x) \quad \text{and} \quad \diamond p = \mu_x(p \cup \ominus x) , \tag{35}$$

and by the Star Adjunction Theorem we have

$$\boxplus p = v_x(p \cap \widetilde{\oplus} x) \quad \text{and} \quad \boxminus p = v_x(p \cap \widetilde{\ominus} x). \tag{36}$$

Temporal logic, which lacks these theorems, employs different induction rules. The following result establishes a temporal induction rule as a theorem of Galois algebra.

**Theorem 4.2.1. (Temporal Induction)** In every Galois algebra,

$$p \cap \boxplus(p \rightarrow \widetilde{\oplus} p) \subseteq \boxplus p. \tag{37}$$

*Proof.*  $p \cap \boxplus(p \rightarrow \widetilde{\oplus} p) \subseteq \boxplus p$   
 $\Leftarrow$  { Induction Rule (20) and (36) }  
 $p \cap \boxplus(p \rightarrow \widetilde{\oplus} p) \subseteq p \cap \widetilde{\oplus}(p \cap \boxplus(p \rightarrow \widetilde{\oplus} p))$   
 $\Leftarrow$  { drop conjunct  $p$  on the rhs,  $\widetilde{\oplus}$  is conjunctive }  
 $p \cap \boxplus(p \rightarrow \widetilde{\oplus} p) \subseteq \widetilde{\oplus} p \cap \widetilde{\oplus} \boxplus(p \rightarrow \widetilde{\oplus} p)$   
 $\Leftarrow$  {  $\boxplus \subseteq \widetilde{\oplus} \circ \boxplus$  }  
 $p \cap \boxplus(p \rightarrow \widetilde{\oplus} p) \subseteq \widetilde{\oplus} p$   
 $\Leftarrow$  {  $\boxplus \subseteq id, \textit{modus ponens}$  }  
 true. □

In Manna and Pnueli’s proof system, the Temporal Induction Rule appears as

$$\boxplus(p \rightarrow \widetilde{\oplus} p) \subseteq \boxplus(p \rightarrow \boxplus p), \tag{38}$$

which is obtained from (37) by shunting the conjunct  $p$  to the right-hand side and then applying  $\boxplus$  to both sides.

### 4.3. Strict operators

Some authors prefer the so-called strict versions of  $\boxplus$  and  $\boxminus$ , which may be defined by

$$\widehat{\boxplus} p = \oplus \boxplus p \quad \text{and} \quad \widehat{\boxminus} p = \widetilde{\oplus} \boxplus p.$$

The corresponding past operators are defined similarly. We could easily derive the calculational laws for the strict operators from those for the non-strict ones, but the following theorem saves us from going through such a tedious procedure.

**Theorem 4.3.1.** If  $(\mathcal{G}, \oplus, \ominus)$  is a Galois algebra, then so is  $(\mathcal{G}, \widehat{\boxplus}, \widehat{\boxminus})$ .

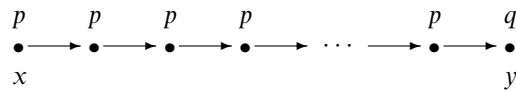
*Proof.* The proof of this theorem highlights the difference between algebraic reasoning and logical deduction. Rather than tediously checking that the proposed structure satisfies all the axioms, we simply remark that it is the sequential composition of two previously known Galois algebras:

$(\mathcal{G}, \oplus \circ \boxplus, \ominus \circ \boxplus)$  is a Galois algebra  
 $\Leftarrow$  { commutativity of  $\oplus$  with  $\boxplus$  (34) }  
 $(\mathcal{G}, \boxplus \circ \oplus, \ominus \circ \boxplus)$  is a Galois algebra  
 $\Leftarrow$  { composition of Galois algebras 2.5.3 }  
 $(\mathcal{G}, \widehat{\boxplus}, \widehat{\boxminus})$  and  $(\mathcal{G}, \oplus, \ominus)$  are Galois algebras. □

As a consequence, the laws in Table 2 remain valid when all the temporal operators are decorated with hats.

4.4. *Until and since*

Another well-known and useful connective of temporal logic is the ‘until’ operator. As with the diamond and box operators, we shall first explain it in terms of a transition relation  $R \subseteq M \times M$  and then generalize it to arbitrary Galois algebras. Assume  $p$  and  $q$  are subsets of  $M$ . Then  $x \in M$  is an element of  $p$  until  $q$  if and only if there is a path from  $x$  to some  $y \in q$  such that every edge  $(a, b)$  in this path satisfies  $(a, b) \in R$  and  $a \in p$ .



If we represent  $p$  by the diagonal relation

$$\Delta_p \stackrel{def}{=} \{(y, y) \mid y \in p\},$$

then the two requirements on edges  $(a, b)$  can be condensed into the single condition  $(a, b) \in \Delta_p ; R$ . Thus we have

$$p \text{ until } q = (\Delta_p ; R)^* \triangleleft q.$$

Now we can calculate an algebraic expression for  $(p \text{ until})$  as follows

$$\begin{aligned} & (p \text{ until}) \\ = & \quad \{ \text{by definition} \} \\ & ((\Delta_p ; R)^* \triangleleft) \\ = & \quad \{ \text{Star Fusion} \} \\ & ((\Delta_p ; R) \triangleleft)^* \\ = & \quad \{ \triangleleft \text{ distributes over composition} \} \\ & ((\Delta_p \triangleleft) \circ (R \triangleleft))^* \\ = & \quad \{ \Delta_p \triangleleft q = p \cap q, \text{ definition of } \oplus \} \\ & ((p \cap) \circ \oplus)^*. \end{aligned}$$

In view of this result, we define, in every Galois algebra,

$$(p \text{ until}) = ((p \cap) \circ \oplus)^*. \tag{39}$$

Using Tail Recursion, we may rewrite (40) to

$$p \text{ until } q = \mu_r(q \cup (p \cap \oplus r)). \tag{40}$$

However, we prefer equation (39) because it reveals that the operator  $(p \text{ until})$  can be seen as the ‘next’ operator of a suitably defined Galois algebra.

**Theorem 4.4.1.** If  $(\mathcal{G}, \oplus, \ominus)$  is a Galois algebra, then so is  $(\mathcal{G}, ((p \cap) \circ \oplus)^*, (\ominus \circ (p \cap))^*)$ .

*Proof.* From the fact that  $(\mathcal{G}, (p \cap), (p \cap))$  is a Galois algebra (by Proposition 2.6.1), the Composition Theorem 2.5.3 and the Iteration Theorem 4.1.1.  $\square$

As a consequence, every law we have established for  $\diamond$  yields an analogous law for  $(p \text{ until})$ . This includes laws mentioning  $\oplus$  if we replace  $\oplus$  by  $(p \cap) \circ \oplus$ . For example,  $\oplus \diamond q = \diamond \oplus q$  translates to the following rolling rule

$$p \cap \oplus(p \text{ until } q) = p \text{ until } (p \cap \oplus q). \tag{41}$$

Unfortunately, Theorem 4.4.1 is tainted by asymmetry in the sense that the two operators  $((p \cap) \circ \oplus)^*$  and  $(\ominus \circ (p \cap))^*$  are not each other’s time-wise duals. In other words, the latter operator is different from the usual ‘since’ operator of temporal logic, which is defined by

$$(p \text{ since}) = ((p \cap) \circ \ominus)^*. \tag{42}$$

In terms of a transition relation, we can trace this flaw to the following asymmetry. The predicate  $p \text{ until } q$  holds at  $x_1$  if there is a path  $x_1 \rightarrow \dots \rightarrow x_n$  such that  $x_n$  satisfies  $q$  and  $x_1, \dots, x_{n-1}$  satisfy  $p$ . Requiring  $p$  at  $x_1$ , but not at  $x_n$ , destroys symmetry. Let’s change the definition by asking that  $p$  should also hold at  $x_n$ . We call the modified operator ‘strong until’ and denote it by  $\text{until}^+$ . It may be defined in terms of the normal ‘until’ operator,

$$p \text{ until}^+ q \stackrel{\text{def}}{=} p \text{ until } (p \cap q). \tag{43}$$

A ‘strong since’ operator is defined similarly:

$$p \text{ since}^+ q \stackrel{\text{def}}{=} p \text{ since } (p \cap q). \tag{44}$$

The Tail Recursion Rule allows us to rewrite these definitions in recursive form

$$p \text{ until}^+ q = \mu_x(p \cap (q \cup \oplus x)) \tag{45}$$

$$p \text{ since}^+ q = \mu_x(p \cap (q \cup \ominus x)). \tag{46}$$

The following theorem shows that the strong operators do not suffer from temporal asymmetry.

**Theorem 4.4.2.**  $(\mathcal{G}, (p \text{ until}^+), (p \text{ since}^+))$  is a Galois algebra.

*Proof.* As before, we need not check the defining axioms, because we can show that the proposed algebra is composed from known Galois algebras. From the definitions (43) and (39) we know that

$$(p \text{ until}^+) = ((p \cap) \circ \oplus)^* \circ (p \cap). \tag{47}$$

An analogous equation holds for the ‘strong since’ operator:

$$(p \text{ since}^+) = ((p \cap) \circ \ominus)^* \circ (p \cap). \tag{48}$$

By virtue of the Leapfrog Rule 3.7.1, we may rewrite (48) to

$$(p \text{ since}^+) = (p \cap) \circ (\ominus \circ (p \cap))^*. \tag{49}$$

Since  $(\mathcal{G}, (p \cap), (p \cap))$  and  $(\mathcal{G}, \oplus, \ominus)$  are Galois algebras, the Composition Theorem 2.5.3 and the Iteration Theorem 4.1.1 imply that

$$(\mathcal{G}, ((p \cap) \circ \oplus)^* \circ (p \cap), (p \cap) \circ (\ominus \circ (p \cap))^*)$$

is a Galois algebra. According to (47) and (49) this is precisely the desired result. □

From a logician’s point of view there is little to choose between the weak and strong versions of ‘since’ and ‘until’, because they are interdefinable, for example by

$$p \text{ until}^+ q = p \text{ until } (p \cap q) \quad \text{and} \quad p \text{ until } q = (p \cup q) \text{ until}^+ q,$$

and so we think it is by accident, rather than design, that most authors base their presentation of temporal logic on the weak operators. In contrast, on the algebraic side Theorem 4.4.2 makes a world of difference, because it says, in a single sentence, everything we need to know about the algebraic properties of  $\text{until}^+$  and  $\text{since}^+$ . Since we are already familiar with the useful theorems for  $\oplus$  and  $\ominus$  (or  $\diamond$  and  $\diamondleftarrow$ ), we need not burden our memory with another bag of algebraic laws.

Theorem 4.4.2 invites another instantiation of Table 1. In order to do this we calculate the upper adjoint of  $(p \text{ since}^+)$ .

$$\begin{aligned} & (p \text{ since}^+)^{\#}.q \\ = & \quad \{ \text{by (49)} \} \\ & ((p \cap) \circ (\ominus \circ (p \cap))^*)^{\#}.q \\ = & \quad \{ \text{Composition Rule for Galois connections 2.1.3} \} \\ & (\ominus \circ (p \cap))^{\#}.(p \cap)^{\#}.q \\ = & \quad \{ (p \cap)^{\#} = (p \rightarrow) \} \\ & (\ominus \circ (p \cap))^{\#}.(p \rightarrow q) \\ = & \quad \{ \text{Star Adjunction Theorem} \} \\ & v_x((p \rightarrow q) \cap (\ominus \circ (p \cap))^{\#}.x) \\ = & \quad \{ (\ominus \circ (p \cap))^{\#} = (p \rightarrow) \circ \widetilde{\oplus} \text{ by 2.1.3} \} \\ & v_x((p \rightarrow q) \cap (p \rightarrow \widetilde{\oplus}x)) \\ = & \quad \{ (p \rightarrow) \text{ is conjunctive} \} \\ & v_x(p \rightarrow (q \cap \widetilde{\oplus}x)). \end{aligned}$$

In terms of a concrete transition system, the expression we just derived for  $(p \text{ since}^+)^{\#}.q$  describes the set of all  $x$  with the following property: If  $s$  is a path that starts at  $x$  and if all nodes of  $s$  satisfy  $p$ , then all nodes of  $s$  satisfy  $q$ . In other words, if we start at  $x$ , then  $q$  cannot become false while  $p$  is true. So let us define

$$\text{while } p \text{ hold } q \stackrel{\text{def}}{=} (p \text{ since}^+)^{\#}.q = v_x(p \rightarrow (q \cap \widetilde{\oplus}x)).$$

Since we have proved that  $(\mathcal{G}, (p \text{ until}^+), (p \text{ since}^+))$  is a Galois algebra, we inherit the following set of laws:

$\begin{aligned} r \text{ since}^+ (\text{while } r \text{ hold } p) &\subseteq p \subseteq \text{while } r \text{ hold } (r \text{ since}^+ p) \\ r \text{ until}^+ \mathbf{F} &= \mathbf{F} = r \text{ since}^+ \mathbf{F} \\ r \text{ until}^+ (p \cup q) &= (r \text{ until}^+ p) \cup (r \text{ until}^+ q) \\ (r \text{ until}^+ p) \cap q &\subseteq r \text{ until}^+ (p \cap (r \text{ since}^+ q)) \\ (r \text{ until}^+ p) \cap (\text{while } r \text{ hold } q) &\subseteq r \text{ until}^+ (p \cap q) \\ \text{while } r \text{ hold } (p \rightarrow q) &\subseteq (\text{while } r \text{ hold } p) \rightarrow (\text{while } r \text{ hold } q) \\ r \text{ until}^+ (p \rightarrow q) &\subseteq (\text{while } r \text{ hold } p) \rightarrow (r \text{ until}^+ q) \end{aligned}$
---

**Table 3.**

**5. Confluence and linearity**

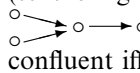
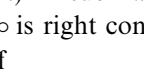
The algebra we have investigated so far works for any pair of operators defined by

$$\oplus p = R \triangleleft p \quad \text{and} \quad \ominus p = R^U \triangleleft p, \tag{50}$$

where  $R \subseteq M \times M$  is an arbitrary relation,  $R^U$  is its transposition and  $p$  ranges over subsets of  $M$ . In Manna and Pnueli’s logic, the relation  $R$  is required to have certain linearity properties. We propose to translate these properties into postulates that can be imposed on Galois algebras so that we can employ the calculus to explore their consequences.

It turns out that quite a number of these can be proved from a much weaker assumption, namely that the transition relation  $R$  be *confluent*. We will therefore look at confluent Galois algebras first.

5.1. *Confluent Galois algebras*

A relation  $R \subseteq M \times M$  is called *right confluent* if its associated directed graph  $(M, R)$  has the following property: any two paths that have the same initial point can be extended (to the right) in such a way that they end up in a common final point. For example,  is right confluent, whereas  is not. In relational terms,  $R$  is right confluent iff

$$(R^U)^*; R^* \subseteq R^*; (R^U)^*. \tag{51}$$

If  $R$  has this property and the Galois operators are defined by (50) and (32), we may calculate

$$\begin{aligned} & \diamond \circ \diamond \\ = & \quad \{ \text{definitions} \} \\ & \ominus^* \circ \oplus^* \\ = & \quad \{ \text{definitions} \} \\ & (R^U \triangleleft)^* \circ (R \triangleleft)^* \\ = & \quad \{ \text{Star Fusion} \} \\ & ((R^U)^* \triangleleft) \circ (R^* \triangleleft) \\ = & \quad \{ \text{distributivity of } \triangleleft \text{ over composition} \} \\ & ((R^U)^*; R^* \triangleleft) \\ \subseteq & \quad \{ \text{right confluence} \} \\ & ((R^*; R^U)^* \triangleleft) \\ = & \quad \{ \text{unwind first four steps} \} \\ & \diamond \circ \diamond. \end{aligned}$$

This result motivates the following definition.

**Definition 5.1.1.** A Galois algebra is called *right confluent* if it satisfies the inequation  $\diamond \circ \diamond \subseteq \diamond \circ \diamond$ . A Galois algebra in which the dual inequation is *left confluent*.

As expected, the (semi-)commutativity of the diamond operators induces a similar relation between the box operators.



**Proposition 5.1.2.** In every right confluent Galois algebra we have

$$\boxplus \circ \boxminus \subseteq \boxminus \circ \boxplus. \tag{52}$$

*Proof.*  $\boxplus \circ \boxminus$   
 = { definitions }  
 $\diamond\# \circ \diamond\#$   
 = { Composition Rule 2.1.3 }  
 $(\diamond \circ \diamond)\#$   
 $\subseteq$  {  $\diamond \circ \diamond \subseteq \diamond \circ \diamond$ , anti-monotonicity of # (2) }  
 $(\diamond \circ \diamond)\#$   
 = { unwind first two steps }  
 $\boxminus \circ \boxplus.$  □

The following result shows that confluence reduces the number of distinct operators that can be formed from composing boxes and diamonds.

**Proposition 5.1.3.** In every right confluent Galois algebra we have

$$\boxplus \circ \diamond \circ \boxplus = \diamond \circ \boxplus \quad \text{and} \quad \diamond \circ \boxplus \circ \diamond = \boxplus \circ \diamond. \tag{53}$$

*Proof.* Since  $\boxplus \subseteq id$ , we certainly have  $\boxplus \circ \diamond \circ \boxplus \subseteq \diamond \circ \boxplus$ . For the other inclusion, we calculate

$\diamond \circ \boxplus$   
 $\subseteq$  { Cancellation Rule 2.1.2:  $id \subseteq \boxplus \circ \diamond$  }  
 $\boxplus \circ \diamond \circ \diamond \circ \boxplus$   
 $\subseteq$  {  $\diamond \circ \diamond \subseteq \diamond \circ \diamond$ , idempotence of  $\boxplus$  }  
 $\boxplus \circ \diamond \circ \diamond \circ \boxplus \circ \boxplus$   
 $\subseteq$  { Cancellation Rule 2.1.2:  $\diamond \circ \boxplus \subseteq id$  }  
 $\boxplus \circ \diamond \circ \boxplus.$

The proof of the second equation is left as an exercise. □

5.2. Reverse induction

Temporal induction rules are suitable for establishing safety properties (expressions that start with a box). The rule we have seen in Section 4.2 starts from the present and works its way, step by step, into the future (or the past). In the presence of confluence it is also possible to do it the other way round, establishing first the truth of the desired property at the very first (or last) moment in time and working from there towards the present. Informally, a state is initial if all its predecessors satisfy F. Therefore we define

$$\text{first} \stackrel{\text{def}}{=} \widetilde{\ominus} F. \tag{54}$$

The following lemma formalizes the induction start.

**Lemma 5.2.1.** In every Galois algebra we have  $p \cap \text{first} \subseteq \boxplus p$ .

*Proof.* The claim is easily proved by appealing to Knaster–Tarski (20) but, as a matter of style, we use this theorem only where it cannot be avoided. Instead, we calculate

$$\begin{aligned}
 & p \cap \text{first} \subseteq \boxminus p \\
 \Leftrightarrow & \quad \{ \text{Galois connection} \} \\
 & \boxplus(p \cap \text{first}) \subseteq p \\
 \Leftrightarrow & \quad \{ \boxplus = id \cup \boxplus \circ \oplus \} \\
 & \boxplus \oplus(p \cap \text{first}) \subseteq p \\
 \Leftarrow & \quad \{ \oplus(p \cap \text{first}) \subseteq \oplus p \cap \oplus \text{first, by monotonicity} \} \\
 & \boxplus(\oplus p \cap \oplus \text{first}) \subseteq p \\
 \Leftrightarrow & \quad \{ \text{first} = \widetilde{\ominus} F, \text{ Cancellation Rule (10)} \} \\
 & \boxplus(\oplus p \cap F) \subseteq p \\
 \Leftrightarrow & \quad \{ \boxplus F = F \} \\
 & \text{true.} \quad \square
 \end{aligned}$$

With the aid of the preceding lemma we are now able to establish the Reverse Induction Rule (which is the algebraic version of one of Manna and Pnueli’s axioms, see (73)).

**Proposition 5.2.2. (Reverse Induction)** In every left-confluent Galois algebra we have

$$\text{first} \cap \boxplus(p \rightarrow \widetilde{\ominus} p) \subseteq \boxplus(p \rightarrow \boxminus p). \tag{55}$$

$$\begin{aligned}
 & \textit{Proof.} \text{ first} \cap \boxplus(p \rightarrow \widetilde{\ominus} p) \\
 \subseteq & \quad \{ \text{Lemma 5.2.1} \} \\
 & \boxminus \boxplus(p \rightarrow \widetilde{\ominus} p) \\
 \subseteq & \quad \{ \text{Proposition 5.1.2 (dual version)} \} \\
 & \boxplus \boxminus(p \rightarrow \widetilde{\ominus} p) \\
 \subseteq & \quad \{ \text{Temporal Induction Rule (38) (dual version)} \} \\
 & \boxplus \boxminus(p \rightarrow \boxminus p) \\
 \subseteq & \quad \{ \boxminus \subseteq id \} \\
 & \boxplus(p \rightarrow \boxminus p). \quad \square
 \end{aligned}$$

We include one more related law, on the grounds that it is also one of Manna and Pnueli’s axioms.

**Proposition 5.2.3.** In every left-confluent Galois algebra we have

$$\text{first} \cap \boxplus p \subseteq \boxplus \widetilde{\ominus} p. \tag{56}$$

$$\begin{aligned}
 & \textit{Proof.} \text{ first} \cap \boxplus p \\
 \subseteq & \quad \{ \text{Lemma 5.2.1} \} \\
 & \boxminus \boxplus p \\
 \subseteq & \quad \{ \text{Proposition 5.1.2 (dual version)} \} \\
 & \boxplus \boxminus p \\
 \subseteq & \quad \{ \boxminus \subseteq \widetilde{\ominus} \} \\
 & \boxplus \widetilde{\ominus} p. \quad \square
 \end{aligned}$$

### 5.3. Linear Galois algebras

A relation  $R \subseteq M \times M$  is said to be *right linear* if no element of  $M$  has two distinct successors ( $R$  does not branch towards the right). With  $R^U$  denoting the transposition of

$R$  this property may be expressed as

$$R^{\cup} \circ R \subseteq id.$$

In view of (50) and the associativity  $(R^{\cup} \circ R) \triangleleft p = R^{\cup} \triangleleft (R \triangleleft p)$ , this postulate may be rewritten to

$$\forall p \subseteq M : \ominus \oplus p \subseteq p.$$

Left linearity may be described in a similar fashion. Therefore, a Galois algebra is said to be *linear* if it satisfies

$$\oplus \circ \ominus \subseteq id \quad \text{and} \quad \ominus \circ \oplus \subseteq id. \tag{57}$$

By virtue of the Galois connections (8) and (9), the above inequations are equivalent to

$$\ominus \subseteq \widetilde{\ominus} \quad \text{and} \quad \oplus \subseteq \widetilde{\oplus}. \tag{58}$$

It is evident that a linear transition relation is confluent. The following result confirms our expectation that the same goes for Galois algebras.

**Theorem 5.3.1.** Every linear Galois algebra is both left and right confluent, in the sense that

$$\diamond \circ \diamond = \diamond \circ \diamond. \tag{59}$$

*Proof.* By the Star Absorption Theorem 3.7.3, both sides are equal to  $\diamond \cup \diamond$ .  $\square$

In Section 2.4 we proved one inclusion (from left to right) of the following distribution law. With the aid of linearity we can show the other inclusion as well.

**Proposition 5.3.2. (Distributivity over implication)** In every linear Galois algebra we have

$$\widetilde{\oplus}(p \rightarrow q) = \widetilde{\oplus}p \rightarrow \widetilde{\oplus}q. \tag{60}$$

*Proof.*  $\widetilde{\oplus}p \rightarrow \widetilde{\oplus}q = \widetilde{\oplus}(p \rightarrow q)$   
 $\Leftrightarrow$  { Proposition 2.4.1 }  
 $\widetilde{\oplus}p \rightarrow \widetilde{\oplus}q \subseteq \widetilde{\oplus}(p \rightarrow q)$   
 $\Leftrightarrow$  { Galois connection }  
 $\ominus(\widetilde{\oplus}p \rightarrow \widetilde{\oplus}q) \subseteq p \rightarrow q$   
 $\Leftrightarrow$  { shunting }  
 $p \cap \ominus(\widetilde{\oplus}p \rightarrow \widetilde{\oplus}q) \subseteq q$   
 $\Leftrightarrow$  { Dedekind }  
 $\ominus(\oplus p \cap (\widetilde{\oplus}p \rightarrow \widetilde{\oplus}q)) \subseteq q$   
 $\Leftrightarrow$  { Galois connection }  
 $\oplus p \cap (\widetilde{\oplus}p \rightarrow \widetilde{\oplus}q) \subseteq \widetilde{\oplus}q$   
 $\Leftrightarrow$  {  $\oplus \subseteq \widetilde{\oplus}$  (by (58)) and *modus ponens* }  
 true.  $\square$

We conclude this section with an analogy to Dijkstra’s predicate transformers. Everybody who has seen Dijkstra’s work on semantics knows the following relation between weakest and weakest liberal preconditions (associated with a fixed program  $\pi$ ).

$$wp_{\pi}.p = wlp_{\pi}.p \cap wp_{\pi}.T \quad \text{for every postcondition } p.$$

Compare this equation to the following result.

**Proposition 5.3.3.** In every linear Galois algebra we have  $\oplus p = \widetilde{\oplus} p \cap \oplus T$ .

$$\begin{aligned}
 & \text{Proof. } \widetilde{\oplus} p \cap \oplus T \\
 \subseteq & \quad \{ \text{Best-of-Both-Worlds} \} \\
 & \oplus p \\
 \subseteq & \quad \{ \oplus \subseteq \widetilde{\oplus}, \text{ monotonicity of } \oplus \} \\
 & \widetilde{\oplus} p \cap \oplus T. \quad \square
 \end{aligned}$$

5.4. *Lagois correspondences*

The purpose of this section is to link Galois algebra to the recently introduced theory of *Lagois connections* (Melton *et al.* 1994). This material will not be needed for the sequel, so some readers may wish to skip directly to Section 5.5.

In a linear Galois algebra, the relation between  $\oplus$  and  $\ominus$  is almost, but not quite, a Galois correspondence (compare (57) with the Cancellation Rule 2.1.2). In fact, it is a *Lagois correspondence*. In order to explain both the difference and the analogy let us present a Galois correspondence on  $\mathcal{G}$  as a triple of functions (*up*, *h*, *down*) where

- *up* is an upwards closure operator, that is, *up* is monotonic, idempotent and  $id \subseteq up$ ;
- *down* is a downwards closure operator, that is, *down* is monotonic, idempotent and  $down \subseteq id$ ;
- *h* is an order isomorphism from  $up.\mathcal{G} \stackrel{def}{=} \{ up.p \mid p \in \mathcal{G} \}$  to  $down.\mathcal{G} \stackrel{def}{=} \{ down.p \mid p \in \mathcal{G} \}$ .

Galois connections are in bijective correspondence with such triples; given a Galois connection (*f*, *g*) the corresponding triple is ( $g \circ f$ , *h*,  $f \circ g$ ), where *h* is the restriction of *f* to  $g.f.\mathcal{G}$ . Given a Galois triple (*up*, *h*, *down*), the associated Galois connection (*f*, *g*) can be retrieved by taking  $f = h \circ up$  and  $g = h^{-1} \circ down$ .

Like a Galois connection, a Lagois connection consists of two closure operators and an isomorphism between the two sets of closed elements. But for Lagois connections, both closures must be of the downward type.

Given a Lagois triple ( $down_1, h, down_2$ ), the associated Lagois pair (*f*, *g*) is defined by  $f = h \circ down_1$  and  $g = h^{-1} \circ down_2$ . The following characterization is proved in Melton *et al.* (1994).

**Proposition 5.4.1.** (*f*, *g*) is a Lagois Connection if and only if the following conditions hold:

- [1] *f* and *g* are monotonic;
- [2]  $f \circ g \circ f = f$  and  $g \circ f \circ g = g$ ;
- [3]  $f \circ g \subseteq id$  and  $g \circ f \subseteq id$ .

These conditions could be used for defining a *Galois* connection, except that the second inclusion sign in [3] would have to be reversed (in which case [2] could be omitted because it would follow from [1] and [3]). Melton *et al.* (1994) shows that Lagois connections are in many ways similar to Galois connections, but also in several ways different. Using elaborate examples involving type coercion and verification of interpreters, they argue that Lagois connections occur naturally in computer science. The following proposition provides a much simpler example.

**Theorem 5.4.2.** A Galois algebra  $(\mathcal{G}, \oplus, \ominus)$  is linear if and only if  $(\oplus, \ominus)$  is a Lagois connection.

*Proof.* If  $(\oplus, \ominus)$  is a Lagois connection then  $\mathcal{G}$  is linear by [3] of the previous proposition. Conversely, assume that  $(\mathcal{G}, \oplus, \ominus)$  is linear. Then [1] and [3] hold, and by time-wise duality we need only show one of the equations in [2], say

$$\oplus \circ \ominus \circ \oplus = \oplus.$$

The inclusion from left to right follows from linearity (57). To prove the other inclusion we appeal to the Cancellation Law (10) and then to Lemma 2.3.4:

$$\oplus \subseteq \oplus \circ \widetilde{\ominus} \circ \oplus \subseteq \oplus \circ \ominus \circ \oplus. \quad \square$$

5.5. Perfect Galois connections

In a linear Galois algebra we have only  $\oplus \subseteq \widetilde{\oplus}$ , whereas in linear temporal logic these two operators coincide. In terms of a transition relation, the identification occurs when every state has precisely one successor, whereas the linearity axiom requires only that there be *at most* one. The following proposition explains the situation in terms of Galois connections.

In Ore (1944), a Galois connection  $(f, g)$  is called *perfect* if  $g \circ f = id$ . A Galois connection is perfect if and only if its lower adjoint is injective; for this reason, perfect Galois connections are sometimes called *Galois insertions* (Melton *et al.* 1986; Melton *et al.* 1994).

**Proposition 5.5.1.** Assume that  $(\mathcal{G}, \oplus, \ominus)$  is a Galois algebra. Then the following conditions are mutually equivalent.

1.  $\ominus \subseteq \widetilde{\ominus}$  and  $\widetilde{\oplus} = \oplus$ ;
2.  $(\ominus, \oplus)$  is a perfect Galois connection;
3.  $(\ominus, \oplus)$  is both a Lagois and a Galois connection;
4.  $\ominus \circ \oplus \subseteq id$  and  $\oplus \circ \ominus = id$ ;
5.  $\ominus \circ \oplus \subseteq id$  and  $\oplus \circ \ominus \subseteq id$  and  $\oplus T = T$ .

We omit the proof because it is completely straightforward. A Galois algebra with Properties 1–5 will be called *strongly linear*. Some authors call a pair of functions satisfying these conditions an injection-projection pair.

Comparing the last condition with the linearity axiom, we see that the additional constraint is

$$\oplus T = T. \tag{61}$$

There are two reasons why we did not introduce this postulate earlier. Firstly, we wished to show that all the hard work is done by the linearity property (57), whereas the addition of (61) has very little effect. In the next section we will prove that a strongly linear Galois algebra satisfies all seventeen of Manna and Pnueli’s axioms, but sixteen of them can be derived without postulate (61).

Another and even more serious objection to postulating (61) is the violation of time-wise symmetry. On the other hand, there is no reason for disliking (61) if we can also have its time-wise dual,  $\ominus T = T$ . Then there is an even simpler way to describe the entire situation, namely by

$$\oplus \text{ is an order isomorphism, and } \ominus \text{ is its inverse.} \tag{62}$$

In terms of a transition relation, this means that every state also has precisely one predecessor. This postulate is usually rejected, and rightly so, because every program and every system starts life in some *initial* state, but some do not terminate.

### 6. Linear temporal logic

Algebraic reasoning is quite different in style from logical deduction, and it is now time for a closer look at the relationship between the two paradigms. First, we recall the basic definitions of linear temporal logic and the seventeen axioms of the sound and complete proof system given in Manna and Pnueli (1991). Then we translate logical notions, such as validity, into the language of algebra. The Completeness Theorem at the end of this section states a condition under which the above-mentioned seventeen axioms can be derived as theorems of Galois algebra.

#### 6.1. Syntax

Formulae are built according to the following grammar:

$$\phi ::= P \mid \phi \cup \psi \mid \neg \phi \mid \oplus \phi \mid \ominus \phi \mid \phi \text{ until } \psi \mid \phi \text{ since } \psi,$$

where  $P$  ranges over some set of propositional variables. In addition we use the following abbreviations:

$$\begin{array}{ll} \phi \cap \psi = \neg(\neg \phi \cup \neg \psi) & \phi \rightarrow \psi = \neg \phi \cup \psi \\ T = \phi \cup \neg \phi & F = \neg T \\ \widetilde{\oplus} \phi = \neg \oplus \neg \phi & \widetilde{\ominus} \phi = \neg \ominus \neg \phi \\ \diamond \phi = T \text{ until } \phi & \diamond \phi = T \text{ since } \phi \\ \boxplus \phi = \neg \diamond \neg \phi & \boxminus \phi = \neg \diamond \neg \phi. \end{array}$$

Temporal logic also uses the following stronger forms of implication and equivalence.

$$\phi \implies \psi = \boxplus(\phi \rightarrow \psi) \qquad \phi \iff \psi = (\phi \implies \psi) \cap (\psi \implies \phi).$$

#### 6.2. Semantics

Let  $\Sigma$  denote some set of *states* and  $\Sigma^\omega$  the set of all infinite sequences over  $\Sigma$ . An *interpretation*  $\mathcal{I}$  maps every propositional variable  $P$  to a subset of  $\Sigma^\omega \times \mathbb{N}$ , namely the set of all  $(\sigma, i)$  where  $P$  is deemed to hold. The semantics of a formula (with respect to a fixed interpretation  $\mathcal{I}$ ) is given as a *validity relation*  $\models$ , which is defined recursively as

follows:

$$\begin{aligned}
 \sigma, i \models P & \quad \text{iff } (\sigma, i) \in \mathcal{I}(P) \\
 \sigma, i \models \phi \cup \psi & \quad \text{iff } \sigma, i \models \phi \text{ or } \sigma, i \models \psi \\
 \sigma, i \models \neg\phi & \quad \text{iff it is not true that } \sigma, i \models \phi \\
 \sigma, i \models \boxplus\phi & \quad \text{iff } \sigma, (i + 1) \models \phi \\
 \sigma, i \models \ominus\phi & \quad \text{iff } i > 0 \text{ and } \sigma, (i - 1) \models \phi \\
 \sigma, i \models \phi \text{ until } \psi & \quad \text{iff there is some } j \text{ with } i \leq j \text{ such that} \\
 & \quad \sigma, j \models \psi \\
 & \quad \text{and } \sigma, k \models \phi \text{ for all } k \text{ with } i \leq k < j \\
 \sigma, i \models \phi \text{ since } \psi & \quad \text{iff there is some } j \text{ with } 0 \leq j \leq i \text{ such that} \\
 & \quad \sigma, j \models \psi \\
 & \quad \text{and } \sigma, k \models \phi \text{ for all } k \text{ with } j < k \leq i .
 \end{aligned}$$

A formula  $\phi$  is called *valid* if  $\sigma, 0 \models \phi$  holds in every interpretation  $\mathcal{I}$ .

If  $\phi \implies \psi$  is valid,  $\phi$  is said to *entail*  $\psi$ . Two formulae that entail each other are called *equivalent*.

### 6.3. Axioms

According to Manna and Pnueli (1991), all valid formulae can be derived<sup>†</sup> from the following set of *axioms*:

$$\phi \text{ until } \psi \iff \psi \cup (\phi \cap \boxplus(\phi \text{ until } \psi)) \tag{63}$$

$$\phi \text{ since } \psi \iff \psi \cup (\phi \cap \ominus(\phi \text{ since } \psi)) \tag{64}$$

$$\phi \text{ until } F \implies F \tag{65}$$

$$\ominus F \tag{66}$$

$$\boxplus\phi \implies \phi \tag{67}$$

$$\boxplus\phi \implies \boxplus\boxplus\phi \tag{68}$$

$$\boxplus(\phi \rightarrow \psi) \implies (\boxplus\phi \rightarrow \boxplus\psi) \tag{69}$$

$$\boxminus(\phi \rightarrow \psi) \implies (\boxminus\phi \rightarrow \boxminus\psi) \tag{70}$$

$$\phi \implies \boxplus\ominus\phi \tag{71}$$

$$\phi \implies \boxminus\boxplus\phi \tag{72}$$

$$\boxplus(\phi \rightarrow \boxplus\phi) \implies \boxplus(\phi \rightarrow \boxplus\phi) \tag{73}$$

$$\boxplus(\phi \rightarrow \boxminus\phi) \rightarrow \boxplus(\phi \rightarrow \boxminus\phi) \tag{74}$$

$$\boxplus\phi \rightarrow \boxplus\boxminus\phi \tag{75}$$

$$\ominus\phi \implies \boxminus\phi \tag{76}$$

<sup>†</sup> To make this precise we should formalize the notion of a proof. We refer the reader to Manna and Pnueli (1991) or Stirling (1992).

$$\widetilde{\ominus}(\phi \rightarrow \psi) \iff (\widetilde{\ominus}\phi \rightarrow \widetilde{\ominus}\psi) \tag{77}$$

$$\widetilde{\oplus}(\phi \rightarrow \psi) \iff (\widetilde{\oplus}\phi \rightarrow \widetilde{\oplus}\psi) \tag{78}$$

$$\widetilde{\oplus}\phi \iff \oplus\phi. \tag{79}$$

The above list is not copied literally from Manna and Pnueli (1991) – we have made some changes to fit our presentation. From a logical point of view, these changes are trivial: modulo propositional tautologies our axiom set is equivalent to the one of Manna and Pnueli. Specifically:

- We have avoided the use of negation, so that the axioms also make sense in an intuitionistic Galois algebra. The only axiom affected by this change is the last one, which appears as  $\oplus\neg\phi \iff \neg\oplus\phi$  in Manna and Pnueli (1991).
- In view of Axiom (79), we have allowed ourselves to replace  $\oplus$  by  $\widetilde{\oplus}$  in a number of axioms. For example, (71) and (72) are time-wise duals, but in Manna and Pnueli (1991) the symmetry is destroyed by using  $\oplus$  in place of  $\widetilde{\oplus}$ . Another benefit of this change is that the modified axioms are valid in many Galois algebras that do not satisfy Axiom (79).
- We have cast the first three axioms in terms of *since* and *until*, whereas Manna and Pnueli use two different operators called *wait-for* and *back-to*. These two pairs of operators are inter-definable via

$$p \text{ until } q = p \text{ wait-for } q \cap \diamond q \quad \text{and} \quad p \text{ wait-for } q = p \text{ until } q \cup \boxplus p,$$

and, symmetrically,

$$p \text{ since } q = p \text{ back-to } q \cap \diamond q \quad \text{and} \quad p \text{ back-to } q = p \text{ since } q \cup \boxminus p.$$

We have preferred *until* and *since* because they can be expressed in terms of iteration, rather than general recursion, and because we have studied them so thoroughly in Section 4.4.

#### 6.4. Translation from logic to algebra

To relate logical concepts to algebra we need an algebraic counterpart of validity. In logic, validity is defined in terms of initial states. A state is initial if it has no predecessors or, equivalently, if all of its predecessors satisfy F. Thus the proposition *first*, which is defined by

$$\text{first} = \widetilde{\ominus}F \tag{80}$$

and thus holds in all initial states, is the *strongest* valid proposition. In other words, an arbitrary proposition  $p$  is valid if and only if it is weaker than *first*. Therefore, we *define* an element  $p$  of a Galois algebra to be valid if

$$\text{first} \subseteq p. \tag{81}$$

By virtue of this definition, the notions of logical entailment and equivalence, which are defined in terms of validity, carry over to elements of a Galois algebra.



Ideally, logical equivalence should coincide with algebraic equality. The next theorem shows that this is indeed the case, provided the Galois algebra under consideration has the *reachability* property. A Galois algebra is said to be *reachable* if it satisfies  $\diamond\text{first} = \text{T}$ . In terms of a transition relation, this is true if and only if every state is reachable in finitely many steps from some initial state.

**Theorem 6.4.1. (Translation from LTL to algebra)** Let  $(\mathcal{G}, \oplus, \ominus)$  be a Galois algebra, and  $p, q \in \mathcal{G}$ . Then we have

1. If  $p \subseteq q$ , then  $p$  entails  $q$ .
2. If  $\mathcal{G}$  is reachable and  $p$  entails  $q$ , then  $p \subseteq q$ .

*Proof.* Both claims follow from the following calculation:

$$\begin{aligned}
 & p \text{ entails } q \\
 \Leftrightarrow & \quad \{ \text{definition of entailment} \} \\
 & p \implies q \text{ is valid} \\
 \Leftrightarrow & \quad \{ \text{definition of } \implies \text{ and of validity} \} \\
 & \text{first} \subseteq \boxplus(p \rightarrow q) \\
 \Leftrightarrow & \quad \{ \text{Galois correspondence} \} \\
 & \diamond\text{first} \subseteq p \rightarrow q \\
 \Leftrightarrow & \quad \{ \text{shunting} \} \\
 & p \cap \diamond\text{first} \subseteq q. \quad \square
 \end{aligned}$$

The preceding theorem is useful for establishing that terms of the shape  $p \implies q$  are valid. Most axioms of linear temporal logic do have this shape, but (74) and (75) are of the form  $p \rightarrow q$ . Their validity can be proved with the aid of the following lemma.

**Lemma 6.4.2.** Let  $(\mathcal{G}, \oplus, \ominus)$  be a Galois algebra, and  $p, q \in \mathcal{G}$ . Then we have

$$p \rightarrow q \text{ is valid} \quad \Leftrightarrow \quad \text{first} \cap p \subseteq q.$$

*Proof.*  $p \rightarrow q$  is valid

$$\begin{aligned}
 \Leftrightarrow & \quad \{ \text{definition of validity} \} \\
 & \text{first} \subseteq p \rightarrow q \\
 \Leftrightarrow & \quad \{ \text{shunting} \} \\
 & \text{first} \cap p \subseteq q. \quad \square
 \end{aligned}$$

### 6.5. Completeness

The following theorem summarizes our knowledge about the relation between Galois algebra and linear temporal logic.

**Theorem 6.5.1. (Completeness)** For any Galois algebra  $G = (\mathcal{G}, \oplus, \ominus)$  we have the following:

1. Axioms (63–73) are valid in  $G$ .
2. If  $G$  is confluent (that is,  $\diamond \circ \diamond = \diamond \circ \diamond$ ), then Axioms (63–75) are valid in  $G$ .
3. If  $G$  is linear (that is,  $\oplus \circ \ominus \subseteq id$  and  $\ominus \circ \oplus \subseteq id$ ), then axioms (63–78) are valid in  $G$ .

4. If  $G$  is strongly linear (that is,  $\ominus \circ \oplus \subseteq \oplus \circ \ominus = id$ ), then all seventeen of the axioms (63–79) are valid in  $G$ .

*Proof.* The validity of each axiom is no more than a restatement of some theorem about Galois algebras that we have established somewhere in this paper. For convenience we provide a list of pointers to the relevant places. Part 1 of the Translation Theorem 6.4.1 is used tacitly.

(63): From (40).

(64): By symmetry.

(65): By (39) and Proposition 4.4.1, ( $p$  until) is a lower adjoint, and therefore strict.

(66): By definition of first (80) and of validity (81).

(67): From the recursive equation (36).

(68): From the recursive equation (36) and the commutativity of  $\boxplus$  with  $\widetilde{\oplus}$  (34).

(69): From the sixth line of Table 2.

(70): By symmetry with (69).

(71) and (72): From the Cancellation Rule (10).

(73): From the Temporal Induction Rule (38).

(74): From the Reverse Induction Rule 5.2.2 and Proposition 6.4.2.

(75): From Proposition 5.2.3 and Proposition 6.4.2.

(76): From (58).

(77) and (78): By 5.3.2 and its time-wise dual.

(79): By Proposition 5.5.1 we have  $\oplus = \widetilde{\oplus}$  in every strongly linear Galois algebra.  $\square$

### 7. Computation tree logic

In this section we show that the operations of CTL (computation tree logic) (Clarke and Emerson 1981) may also be defined in Galois algebra, and that all of its axioms come out as theorems, with one exception: in CTL it is assumed (as in Linear Temporal Logic) that every node in a computation tree has at least one successor, *i.e.*, that  $\oplus T = T$ . If we add this postulate to the axioms of Galois algebra, we can translate to and then prove within Galois algebra all axioms and rules of a complete proof system for CTL.

#### 7.1. Syntax

CTL formulae are built according to the following grammar:

$$\phi ::= P \mid \phi \cup \phi \mid \neg\phi \mid EX\phi \mid AX\phi \mid E\phi U\phi \mid A\phi U\phi,$$

where  $P$  ranges over some set of propositional variables. In addition we use the following abbreviations:

$$\begin{array}{ll} T = \phi \cup \neg\phi & F = \neg T \\ \phi \cap \psi = \neg(\neg\phi \cup \neg\psi) & \phi \rightarrow \psi = \neg\phi \cup \psi \\ EF\phi = E T U \phi & AF\phi = A T U \phi \\ EG\phi = \neg AF \neg\phi & AG\phi = \neg EF \neg\phi. \end{array}$$

## 7.2. Semantics

Let  $\Sigma$  denote some set of *states* and let  $R$  be a relation on  $S$ . It is required that  $R$  be total, that is for every  $\sigma \in \Sigma$  there is a  $\sigma' \in \Sigma$  with  $(\sigma, \sigma') \in R$ . A path is an infinite sequence  $s_0, s_1, \dots$  such that  $(s_i, s_{i+1}) \in R$  for every  $i \geq 0$ . An *interpretation*  $\mathcal{I}$  maps every propositional variable  $P$  to a subset of  $\Sigma$ , namely the set of all  $\sigma$  where  $P$  is deemed to hold. The semantics of a formula (with respect to a fixed structure  $(\Sigma, R, \mathcal{I})$ ) is given as a *validity relation*  $\models$ , which is defined recursively as follows:

$$\begin{aligned}
\sigma_0 \models \phi & \quad \text{iff } \sigma_0 \in \mathcal{I}(P) \\
\sigma_0 \models \phi \cup \psi & \quad \text{iff } \sigma_0 \models \phi \text{ or } \sigma_0 \models \psi \\
\sigma_0 \models \neg\phi & \quad \text{iff it is not true that } \sigma_0 \models \phi \\
\sigma_0 \models \text{EX } \phi & \quad \text{iff there is a } \sigma_1 \text{ with } (\sigma_0, \sigma_1) \in R \text{ and } \sigma_1 \models \phi \\
\sigma_0 \models \text{AX } \phi & \quad \text{iff } \sigma_1 \models \phi \text{ for all } \sigma_1 \text{ with } (\sigma_0, \sigma_1) \in R \\
\sigma_0 \models \text{E } \phi \text{ U } \psi & \quad \text{iff there is a path } \sigma_0, \sigma_1, \dots \text{ and some } i \geq 0 \text{ such that} \\
& \quad \sigma_i \models \psi \\
& \quad \text{and } \sigma_j \models \phi \text{ for all } j < i \\
\sigma_0 \models \text{A } \phi \text{ U } \psi & \quad \text{iff for every path } \sigma_0, \sigma_1, \dots \text{ there is some } i \geq 0 \text{ such that} \\
& \quad \sigma_i \models \psi \\
& \quad \text{and } \sigma_j \models \phi \text{ for all } j < i
\end{aligned}$$

A CTL formula  $\phi$  is called *valid* if  $\sigma \models \phi$  holds for every state  $\sigma$  of every possible structure. In Emerson (1990) we find the following complete proof system for CTL

- (Ax1) All validities of propositional logic;
- (Ax2)  $\text{EF } \phi \equiv \text{E T U } \phi$ ;
- (Ax2b)  $\text{AG } \phi \equiv \neg \text{EF } \neg \phi$ ;
- (Ax3)  $\text{AF } \phi \equiv \text{A T U } \phi$ ;
- (Ax3b)  $\text{EG } \phi \equiv \neg \text{AF } \neg \phi$ ;
- (Ax4)  $\text{EX } (\pi \cup \phi) \equiv \text{EX } \pi \cup \text{EX } \phi$ ;
- (Ax5)  $\text{AX } \pi \equiv \neg \text{EX } \neg \pi$ ;
- (Ax6)  $\text{E } \phi \text{ U } \psi \equiv \psi \cup (\phi \cap \text{EX } (\text{E } \pi \text{ U } \psi))$ ;
- (Ax7)  $\text{A } \phi \text{ U } \psi \equiv \psi \cup (\phi \cap \text{AX } (\text{A } \pi \text{ U } \psi))$ ;
- (Ax8) EXT;
- (Ax8b) AX T;
- (Ax9)  $\text{AG } (\psi \rightarrow (\neg \phi \cap \text{EX } \psi)) \rightarrow (\psi \rightarrow \neg \text{A } \pi \text{ U } \phi)$ ;
- (Ax9b)  $\text{AG } (\psi \rightarrow (\neg \phi \cap \text{EX } \psi)) \rightarrow (\psi \rightarrow \neg \text{AF } \phi)$ ;
- (Ax10)  $\text{AG } (\psi \rightarrow (\neg \phi \cap (\pi \rightarrow \text{AX } \psi))) \rightarrow (\psi \rightarrow \neg \text{E } \pi \text{ U } \phi)$ ;
- (Ax10b)  $\text{AG } (\psi \rightarrow (\neg \phi \cap \text{EX } \psi)) \rightarrow (\psi \rightarrow \neg \text{EF } \phi)$ ;
- (Ax11)  $\text{AX } (\pi \rightarrow \phi) \rightarrow (\text{EX } \pi \rightarrow \text{EX } \phi)$ ;
- (Rule1) If  $\phi$  is valid then so is  $\text{AG } \phi$  (Generalization);
- (Rule2) If  $\phi$  and  $\phi \rightarrow \psi$  are valid then so is  $\psi$  (*modus ponens*).

7.3. Translation of CTL formulae to Galois algebra

The CTL operators may be defined in a Galois algebra as follows:

$$\begin{aligned}
 EX p &\stackrel{def}{=} \oplus p \\
 AX p &\stackrel{def}{=} \widetilde{\oplus} p \\
 EF p &\stackrel{def}{=} \diamond p \\
 AF p &\stackrel{def}{=} \mu_r(p \cup \widetilde{\oplus} r) \\
 AG p &\stackrel{def}{=} \boxplus p \\
 EG p &\stackrel{def}{=} \nu_r(p \cap \oplus r) \\
 E p U q &\stackrel{def}{=} p \text{ until } q \\
 A p U q &\stackrel{def}{=} \mu_r(q \cup (p \cap \widetilde{\oplus} r)).
 \end{aligned}$$

Since validity in CTL means validity in *all* states, we call an element of a Galois algebra *valid* if and only if it is equal to T. In other words, T is the only valid element. With this interpretation, all the above-listed axioms and rules for CTL become theorems of classical Galois algebra. We leave it to the reader to check the details.

The above axiomatization makes heavy use of negation. It would be nice to find an alternative axiomatization that does not require negation, except as in (Ax1) (all validities of propositional logic). This would provide further evidence for our conjecture that much of temporal logic can be generalized to the intuitionistic case. This is left as future work.

8. Examples

This section provides a collection of examples of Galois algebras that are relevant to computer science. The first subsection is about classical Galois algebras, but all subsequent examples are truly intuitionistic.

8.1. Timing diagrams

Consider a discrete time domain, say  $Time = \mathbb{Z}$ , and some arbitrary set  $\Sigma$  of possible states. A *timing diagram* is a function from some finite interval of  $\mathbb{Z}$  to  $\Sigma$ . Let  $M$  be the set of all such timing diagrams and let  $\mathcal{G}$  be the powerset of  $M$ .

We say that timing diagram  $\eta$  can evolve into timing diagram  $\zeta$  if  $\zeta$  can be obtained from  $\eta$  by deleting its leftmost value and appending an arbitrary value to the right. In other words,  $\eta$  and  $\zeta$  agree on every point where both of them are defined, but the domain of  $\zeta$  is shifted one place to the right.

The ‘next’ and ‘previous’ operators are defined with respect to this transition relation on timing diagrams, that is,

$$\begin{aligned}
 \oplus p &= \{\zeta \mid \zeta \text{ can evolve to some } \eta \in p\} \\
 \ominus p &= \{\eta \mid \eta \text{ can evolve from some } \zeta \in p\}.
 \end{aligned}$$

Clearly,  $(\mathcal{G}, \oplus, \ominus)$  is a classical Galois algebra. This algebra is attractive because it also has a natural sequential composition operator, as explained below.

If  $\zeta_1$  and  $\zeta_2$  are adjacent timing diagrams (with  $\zeta_1$  on the left and  $\zeta_2$  on the right) that have the same value on the unique point in the intersection of their domains, then their union  $\zeta_1; \zeta_2 \stackrel{def}{=} \zeta_1 \cup \zeta_2$  is again a timing diagram. This composition lifts to a total operation on  $\mathcal{G}$ , defined by

$$p; q \stackrel{def}{=} \{ \zeta_1; \zeta_2 \mid \zeta_1 \in p \wedge \zeta_2 \in q \wedge \zeta_1; \zeta_2 \text{ is defined} \}.$$

We have shown in von Karger and Berghammer (1997) that  $(\mathcal{G}, ;)$  is a *sequential algebra* (for a treatment of sequential algebra and its calculus see von Karger and Hoare (1995) and von Karger (1997)), and that the temporal operators can be defined in terms of sequential composition. Thus both temporal and sequential reasoning can be performed in the same model. In von Karger and Berghammer (1997) we show that Galois algebra applies equally well to continuous time domains, and we also show how to construct algebras of timing diagrams that give rise to *linear* Galois algebras.

8.2. Relational semantics for total correctness

There are many ways to define semantics for sequential programs, but perhaps the simplest idea is to represent programs as relations between inputs and outputs drawn from a common set  $\Sigma$  of possible states. Following Hoare and He (Hoare and He 1985), we add one irregular state  $\perp$  to represent nontermination.

Assume that we are interested exclusively in *total correctness*. Then, Hoare and He argue, there is no point in distinguishing between a program that may possibly diverge and one that will surely diverge (for a given input), because both are equally unsatisfactory. Instead of working with classes of equivalent programs, it is convenient to restrict attention to a set of representatives, subsequently called *programs*. The natural candidate to pick from each class is its maximal element. Thus a relation  $R$  on  $\Sigma \cup \{\perp\}$  is a *program* iff

$$\forall \sigma, \tau \in \Sigma \cup \{\perp\} : (\sigma, \perp) \in R \Rightarrow (\sigma, \tau) \in R.$$

This condition is sometimes referred to as demonic (or chaotic) closure. It is possible to impose further healthiness conditions on programs, such as the ‘absence of miracles’, but we shall not do so here, because we do not wish to destroy the lattice structure. The set  $\mathcal{P}$  of all programs is closed under unions and intersections but not under complements; it forms a complete Heyting algebra. If  $R$  is a program, its converse

$$R^\cup \stackrel{def}{=} \{ (\tau, \sigma) \mid (\sigma, \tau) \in R \}$$

need not, in general, be a program. However, when  $R$  and  $S$  are programs, then so are  $R; S$  and  $R^\cup; S$ . The following theorem shows that every program in this sense gives rise to an (intuitionistic) Galois algebra.

**Theorem 8.2.1.** Let  $R$  be any fixed program and define

$$\oplus S \stackrel{def}{=} R; S \quad \text{and} \quad \ominus S \stackrel{def}{=} R^\cup; S$$

for every program  $S$ . Then  $(\mathcal{P}, \oplus, \ominus)$  is a Galois algebra.

*Proof.* The proof follows from Lemma 2.5.2 and the fact that

$$X; Y \cap Z \subseteq X; (Y \cap X^U; Z)$$

holds for arbitrary relations  $X, Y$  and  $Z$ . □

### 8.3. Prefix-closed sets of traces

In the previous section we encountered the idea that a semantics should only distinguish between two programs if there is at least one specification that is satisfied by only one of them, and we saw how this principle can give rise to a closure condition.

In a similar vein, it might be desirable to distinguish between two specifications only when there is (or at least might be) a program or system satisfying the one but not the other. Specifications are often given as sets, each element of which represents a piece of information (a *token*) about the system to be described. Tokens need not be independent. For instance any system admitting the token

On input of 1 Volt the output will be between 9 and 11 Volt.

must also admit the token

On input of 1 Volt the output will be between 8 and 12 Volt.

Therefore token sets should be deductively closed (this is one of Scott's basic postulates in his treatment of domains for denotational semantics (Scott 1982)). In general, the complement of a deductively closed set is not deductively closed. As a consequence, the lattice of all deductively closed token sets is usually not Boolean, but in most practical cases it is still a Heyting algebra.

The trace model of the process algebra CSP (see Olderog and Hoare (1986) for an overview of models for CSP) is a good example. If a process is capable of engaging in a sequence  $t$  of events, it can certainly also engage in any prefix (initial subsequence) of  $t$ . Thus a process is described not by an arbitrary set of traces, but by a prefix-closed one.

Take a fixed alphabet  $A$  and let  $A^*$  be the set of all finite sequences (traces) over  $A$ . A subset  $p$  of  $A^*$  is *prefix-closed* if we have, for all  $r, s \in A^*$

$$rs \in p \quad \Rightarrow \quad r \in p.$$

Now let  $\mathcal{H}$  denote the set of all prefix-closed subsets of  $A^*$ ; this is a Heyting algebra. Define a transition relation  $R \subseteq A^* \times A^*$  by

$$R \stackrel{def}{=} \{(s, as) \mid s \in A^*, a \in A\} \cup \{(\epsilon, \epsilon)\}$$

so that every trace has exactly one predecessor. Then  $\oplus$  and  $\ominus$  can be defined in terms of the transition relation as before

$$\oplus p = R \triangleleft p \quad \text{and} \quad \ominus p = R^U \triangleleft p. \tag{82}$$

Then we have the following result.

**Theorem 8.3.1.** With  $\oplus$  and  $\ominus$  defined as above, the set of all prefix-closed sets of traces over  $A$  is an (intuitionistic) Galois algebra.

*Proof.* We have seen in Section 2.2 that the operators  $R\triangleleft$  and  $R^{\cup}\triangleleft$  are universally disjunctive and satisfy the Dedekind laws. As the reader may check,  $\oplus$  and  $\ominus$  preserve prefix-closedness. We can therefore appeal to Lemma 2.5.2.  $\square$

#### 8.4. Monotonic predicate transformers

A mapping from a Boolean algebra to itself is often called a *predicate transformer*. With pointwise ordering, the set of all predicate transformers over a given complete Boolean algebra  $\mathcal{B}$  is itself a complete Boolean algebra. Since monotonicity is an indispensable condition for inequational reasoning, attention is usually restricted to the set  $Mon(\mathcal{B})$  of all monotonic predicate transformers. Clearly,  $Mon(\mathcal{B})$  is a Heyting algebra.

In Dijkstra’s theory of programs, the weakest liberal precondition operator associates with every program a universally conjunctive predicate transformer. The following theorem shows that every universally conjunctive predicate transformer induces an intuitionistic Galois algebra.

**Theorem 8.4.1.** Suppose that  $\mathcal{B}$  is a complete Boolean algebra and that  $g : \mathcal{B} \rightarrow \mathcal{B}$  is universally conjunctive. Define  $\oplus, \ominus : Mon(\mathcal{B}) \rightarrow Mon(\mathcal{B})$  by

$$(\oplus h).b \stackrel{def}{=} \neg g.(\neg h.b) \quad \text{and} \quad (\ominus h).b \stackrel{def}{=} g^b.h.b$$

Then  $(Mon(\mathcal{B}), \oplus, \ominus)$  is an intuitionistic Galois algebra.

*Proof.* Let  $\mathcal{G}$  denote the set of all functions on  $\mathcal{B}$  and define the function  $f$  by  $f.b = \neg g.\neg b$  (for all  $b \in \mathcal{B}$ ). Then  $\mathcal{G}$  is a complete Boolean algebra, and we can extend the definitions of  $\oplus$  and  $\ominus$  to  $\mathcal{G}$  by letting

$$\oplus h = f \circ h \quad \text{and} \quad \ominus h = g^b \circ h$$

for all  $h \in \mathcal{G}$ . Clearly,  $(\mathcal{G}, \oplus, \ominus)$  is a classical Galois algebra. In particular,  $\oplus$  and  $\ominus$  satisfy the Dedekind laws. Therefore Lemma 2.5.2 implies that  $(Mon(\mathcal{B}), \oplus, \ominus)$  is an (intuitionistic) Galois algebra.  $\square$

#### 8.5. Timed relations

A relation may be seen as a specification that prescribes for every possible input the set of allowed outputs. A *timed* relation specifies, in addition, an upper time bound for each allowed computation. Just as an ordinary relation can be seen as a Boolean matrix (F for forbidden and T for allowed), a timed relation can be seen as a matrix with entries drawn from the set of nonnegative reals augmented by the special values  $\perp$  (forbidden transition) and  $\top$  (allowed transition without an upper time bound).

The set of all such matrices forms a Heyting algebra. Composition is defined by

$$(R; S)_{\rho\tau} \stackrel{def}{=} \max\{R_{\rho\sigma} + S_{\sigma\tau} \mid \sigma \in \Sigma\}. \tag{83}$$

For this definition to make sense, we must extend the addition on  $\mathbb{R}$  to the exceptional values  $\perp$  and  $\top$ . Recall that  $R_{\rho\sigma} = \perp$  means that  $\sigma$  is not reachable from  $\rho$ . Thus if either of the two terms  $R_{\rho\sigma}$  or  $S_{\sigma\tau}$  equals  $\perp$ , their sum should not contribute to the right-hand

side of (83). The way to ensure this is by defining

$$\perp + r = \perp = r + \perp \quad \text{for all } r \in \mathbb{R} \cup \{\perp, \top\}.$$

On the other hand, if one of the two terms  $R_{\rho\sigma}$  and  $S_{\sigma\tau}$  equals  $\top$  and the other is different from  $\perp$ , the right-hand side of (83) should be  $\top$  as well, because the time it will take to get from  $\rho$  to  $\tau$  cannot be bounded. Therefore we define

$$\top + r = \top = r + \top \quad \text{for all } r \in \mathbb{R} \cup \{\top\}.$$

Finally, the converse of a timed relation is defined in the same way as in the untimed case by

$$(R^\cup)_{\sigma\tau} \stackrel{\text{def}}{=} R_{\tau\sigma}.$$

Virtually every law valid for ordinary relations (and not involving complementation) holds for timed relations as well. This is true, in particular for the Dedekind laws, and from that one obtains the following result.

**Proposition 8.5.1.** Let  $R$  be a fixed timed relation and define

$$\oplus S \stackrel{\text{def}}{=} R; S \quad \text{and} \quad \ominus S \stackrel{\text{def}}{=} R^\cup; S,$$

for every timed relation  $S$ . Then the set of all timed relations over  $\Sigma$ , equipped with these operators, forms an (intuitionistic) Galois algebra.

### 8.6. Fuzzy relations

Fuzzy sets and fuzzy relations have proved extremely useful for quantitative reasoning about imperfect knowledge. Given a set  $M$  (the elements of which are referred to as properties), a fuzzy subset of  $M$  is a mapping from  $M$  to the real interval  $[0, 1]$ . A fuzzy relation on  $M$  is a mapping from  $M \times M$  to  $[0, 1]$ . Various products of fuzzy sets with fuzzy relations are studied in the literature (and have been implemented in hardware). The most popular seems to be the so-called Max–Min product. For a fuzzy set  $p$  and a fuzzy relation  $R$ , the fuzzy set  $R \triangleleft p$  is defined as follows:

$$(R \triangleleft p).x = \max\{\min(R(x, y), p.y) \mid y \in M\}. \tag{84}$$

(When  $M$  is infinite, supremum and infimum have to be used instead of max and min). The converse  $R^\cup$  of a fuzzy relation  $R$  is defined by  $R^\cup.(x, y) = R.(y, x)$ . If ordinary sets/relations are regarded as fuzzy sets/relations that take no fractional values, these definitions specialize to the familiar operations of the relational calculus. Therefore, the following result is not unexpected.

**Proposition 8.6.1.** Let  $M$  be a set and let  $\mathcal{G}$  denote the set of all fuzzy subsets of  $M$ . If  $R$  is a fuzzy relation on  $M$  and the operators  $\oplus$  and  $\ominus$  are defined by

$$\oplus p = R \triangleleft p \quad \text{and} \quad \ominus p = R^\cup \triangleleft p \quad \text{for all } p \in \mathcal{G},$$

then  $(\mathcal{G}, \oplus, \ominus)$  is a Galois algebra.

There is, of course, no complement operator.



### 9. A reachability algorithm

In this section, we apply Galois algebra in the derivation of a simple graph-theoretic algorithm. The algorithm is well-known and has been derived computationally before (Möller 1991; Backhouse *et al.* 1994), but never, we think, in such an elegant and concise manner.

A *directed graph* is a pair  $(M, R)$  where  $M$  is a set and  $R$  is a relation on  $M$ . The elements of  $M$  are called vertices and  $R$  is the set of edges. We consider the following task:

For  $q \subseteq M$ , compute the set of all vertices from which a vertex in  $q$  can be reached.

Recall that the powerset of  $M$  is a classical Galois algebra with the ‘next’ operator defined by

$$\oplus p = R \triangleleft p = \{x \mid \exists y : (x, y) \in R \text{ and } y \in p\}.$$

In graph-theoretical terms,  $\oplus p$  consists of all predecessors of nodes in  $p$ . In Galois algebra, we can state the specification more succinctly as

$$\text{given a set } q \text{ of vertices, compute the set } reach.q \stackrel{def}{=} \diamond q.$$

It is assumed that we know how to compute  $\oplus p$  from  $p$ , but the specification  $reach.q = \diamond q$  is not algorithmic. To make it executable we must express it in terms of  $\oplus$  and recursion. Given the fact that  $\diamond = id \cup \diamond \circ \oplus$ , we might propose

$$reach.q = q \cup reach.\oplus q.$$

This specification can be executed immediately as a tail-recursive program. It will certainly find all reachable vertices; unfortunately it will not terminate, but keep calling itself even after all reachable vertices have been found. We must add a way of terminating the recursion. Observing that  $\diamond F = F$ , we modify our program by testing for a termination case

$$reach.q = \text{if } (q = F) \text{ then } F \text{ else } q \cup reach.\oplus q.$$

Note that we continue to denote the top and bottom elements by  $T$  and  $F$ , that is,  $T = M$  and  $F = \emptyset$ . The above recursive program works for a graph without cycles. To ensure termination in the general case we have to break cycles. We propose to do this by keeping track of the vertices that have been visited so far. So let us introduce a second parameter,  $p$ , to hold the set of all vertices visited so far.

More specifically, we aim at a function with two parameters where the first parameter,  $p$ , is the set of vertices visited so far, and the second parameter,  $q$ , is the set of vertices currently in hand. The result has to contain the vertices in  $p$  and also those vertices from which a vertex in  $q$  can be reached without revisiting any vertex in  $p$ . From this informal discussion, we ‘guess’ the following specification

$$f(p, q) \stackrel{def}{=} p \cup (\neg p \text{ until } q).$$

To check that an implementation of  $f$  will indeed solve the original problem, we calculate

$$f(F, q) = F \cup ((\neg F) \text{ until } q) = T \text{ until } q = \diamond q = reach.q.$$

It remains to implement  $f$ . Clearly,  $f(p, \mathbb{F}) = p$ ; this is the termination case. For arbitrary second argument  $q$ , we calculate

$$\begin{aligned}
 & f(p, q) \\
 = & \quad \{ \text{definition of } f \} \\
 & p \cup (\neg p \text{ until } q) \\
 = & \quad \{ x \text{ until } y = (x \cap \neg y) \text{ until } y \text{ (very easy exercise)} \} \\
 & p \cup ((\neg p \cap \neg q) \text{ until } q) \\
 = & \quad \{ \text{let } r \stackrel{\text{def}}{=} p \cup q \} \\
 & p \cup (\neg r \text{ until } q) \\
 = & \quad \{ \text{expand until, using (39)} \} \\
 & p \cup q \cup (\neg r \cap \oplus(\neg r \text{ until } q)) \\
 = & \quad \{ \text{definition of } r \} \\
 & r \cup (\neg r \cap \oplus(\neg r \text{ until } q)) \\
 = & \quad \{ \text{rolling, using (41)} \} \\
 & r \cup (\neg r \text{ until } (\neg r \cap \oplus q)) \\
 = & \quad \{ \text{definition of } f \} \\
 & f(r, \neg r \cap \oplus q).
 \end{aligned}$$

With this we are done. The implementation is now given as

$$f(p, q) = \text{if } (q = \mathbb{F}) \text{ then } p \text{ else } (\text{let } r = p \cup q \text{ in } f(r, \neg r \cap \oplus q)).$$

Termination is assured because the first argument of  $f$  increases with every call.

### 10. Concluding remarks

We have advocated a calculational style for reasoning about temporal propositions, but it would be misguided to suggest that Galois algebra can replace temporal logic. The logical approach is invaluable for investigating decidability, axiomatizability, expressiveness and related issues. Thanks to the existing decision procedures, temporal logic is also very useful for automatic verification, for example by model checking.

On the other hand, we would like to suggest that Galois algebra is better suited to the human user. Mathematicians (and human beings in general) think in concepts, not in formulae. Rather than storing seventeen axioms in our head – no problem for a computer! – we prefer to remember that  $\ominus$  and  $\oplus$  are conjugate, and (in the linear case) a perfect Galois connection. Analogies and structural properties like symmetry and duality are more readily recognized and proved in algebra; they form the core of our, human, understanding of what temporal reasoning is about.

Let us contrast the style of algebraic calculation with that of logical deduction. Equational or inequational rewriting is *the* most fundamental technique of mathematics and everybody learns it at school. We have tried to show (as others have before us) that calculations are guided by the shape of the formulae, and proofs flow naturally from the dynamics of the symbols. Rarely, if ever, do we need to make an unexpected step ('pull a rabbit from a hat'). In contrast, logical deductions often build up to the final theorem in

a bottom up fashion, giving little pieces at first without saying how they will fit together. In some deduction styles, proofs have little structure; in others the intermediate results form a tree, which is hard to write down on paper. It is no accident that chains of (in)equations or implications are in common use throughout the mathematical literature, whereas logical deduction systems are hardly ever employed for convincing readers of theorems (except, of course, in books about logic).

The trade-off between the algebraic and the logical style may be compared to the relation between specification languages and programming languages. The latter are designed for the use of computers, and up to this day they have been far more successful than the former. However, painful experience has taught us that errors occur most easily, and are most costly, in the initial, informal phase of system design. To minimize the risk of error at this stage, specification languages are tailored for human use – often at the price of sacrificing mechanical niceties, such as an LR(1) syntax and automatic executability. The art of turning specifications into implementations is gradually becoming an established engineering craft and, as a consequence, specification languages are steadily gaining acceptance. In a similar way, we expect that the calculational and human-oriented method of algebraic manipulation will gain its place beside the established machinery of logical deduction.

### Acknowledgments

I gratefully acknowledge the valuable discussions in ProCoS and IFIP WG 2.2, where this work was first presented. I had extensive comments from Jaap van der Woude. He and Roland Backhouse suggested using regular algebra. Rudolf Berghammer proposed using temporal operators for deriving the reachability algorithm. Rutger Dijkstra pointed out a couple of errors in the examples. The anonymous MSCS referees suggested several improvements. Special thanks are due to Tony Hoare for encouragement and helpful comments on earlier drafts.

### References

- Aarts, C. J. (1992) Galois connections presented calculationally. Eindhoven University of Technology. Available at [ftp.win.tue.nl](ftp.win.tue.nl/pub/math.prog.construction/galois.dvi.Z) in `/pub/math.prog.construction/galois.dvi.Z`.
- Backhouse, R. C., Aarts, C. J., Hoogendijk, P., Voermans, E. and van der Woude, J. (1992) A relational theory of datatypes. Manuscript, Eindhoven University of Technology.
- Backhouse, R. C., van den Eijnde, J. and van Gasteren, A. J. M. (1994) Calculating path algorithms. *Science of Computer Programming*.
- Clarke, E. M. and Emerson, E. A. (1981) Design and synthesis of synchronization skeletons using branching time temporal logic. In: Proc. Workshop on Logics of Programs. *Springer-Verlag Lecture Notes in Computer Science* **131** 52–71.
- Emerson, E. A. (1990) Temporal and modal logic. In: van Leeuwen, J. (ed.) Formal Models and Semantics. *Handbook of Theoretical Computer Science* Volume **B**, Chapter 16, Elsevier 995–1072.
- Everett, C. J. (1944) Closure operators and Galois theory in lattices. *Trans. Amer. Math. Soc.* **55** 514–525.

- van Gasteren, A. J. M. (1991) On the Shape of Mathematical Argument. *Springer-Verlag Lecture Notes in Computer Science* **445**.
- Herrlich, H. and Hušek, M. (1985) Galois connections. Proceedings of MFPS. *Springer-Verlag Lecture Notes in Computer Science* **239** 122–134.
- Hoare, C. A. R. and He Jifeng (1985) The weakest prespecification. Technical Report PRG-44, Oxford University.
- Hoare, C. A. R. and von Karger, B. (1995) Sequential calculus. *Information Processing Letters* **53** (3) 123–130.
- von Karger, B. (1997). Sequential calculus. (In preparation, preliminary version available at <http://informatik.uni-kiel.de/~bvk/>.)
- von Karger, B. and Berghammer, R. (1996) A relational model for temporal logic. (Accepted for Publication in the Bulletin of the IGPL, available at [www.informatik.uni-kiel.de/~bvk/](http://www.informatik.uni-kiel.de/~bvk/).)
- Manna, Z. and Pnueli, A. (1991) *The Temporal Logic of Reactive and Concurrent Systems – Specification*, Springer-Verlag.
- Melton, A., Schmidt, D. A. and Strecker, G. E. (1986) Galois connections and computer science applications. In: Pitt, D., Abramsky, S., Poigné, A. and Rydeheard, D. (eds.) *Category Theory and Computer Programming. Springer-Verlag Lecture Notes in Computer Science* **240** 299–312.
- Melton, A., Schröder, B. S. W. and Strecker, G. E. (1994) Lagois connections – a counterpart to Galois connections. *Theoretical Computer Science* **136** (1) 79–108.
- Möller, B. (1991) Relations as a program development language. *Proc. IFIP TC2 Conference*, North Holland.
- Olderog, E. R. and Hoare, C. A. R. (1986) Specification oriented semantics for communicating processes. *Acta Inf.* **23** 9–66.
- Ore, O. (1944) Galois connexions. *Trans. Amer. Math. Soc.* **55** 493–513.
- Rasiowa, H. and Sikorski, R. (1963) *Mathematics of Metamathematics*, Polish Scientific Publishers, Warsaw.
- Scott, D. (1982) Domains for denotational semantics. ICALP '82. *Springer-Verlag Lecture Notes in Computer Science* **140** 577–613.
- Stirling, C. (1992) Modal and temporal logics. In: Abramsky, S., Gabbay, D. M. and Maibaum, T. S. E. (eds.) *Background: Computational Structures. Handbook of Logic in Computer Science Volume 2*, Clarendon Press 478–551.
- Tarski, A. and Jónsson, B. (1951/52) Boolean algebras with operators, Parts I–II. *Amer. J. Math.* **73** 891–939, **74** 127–162.
- Vickers, S. (1988) *Topology via Logic*, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press.