UNIVERSITÉ
LAVAL

FACULTÉ DES SCIENCES ET DE GÉNIE
Département d'informatique et génie logiciel

Pavillon Adrien-Pouliot, local 3908
Cité universitaire
Québec, Canada
G1K 7P4

MODAL KLEENE ALGEBRA AND APPLICATIONS A SURVEY

PAR

JULES DESHARNAIS,
BERNHARD MÖLLER ET
GEORG STRUTH

RAPPORT DE RECHERCHE
DIUL-RR-0401

DÉPARTEMENT D'INFORMATIQUE ET DE GÉNIE LOGICIEL
FACULTÉ DES SCIENCES ET DE GÉNIE

Pavillon Adrien-Pouliot
Université Laval
Sainte-Foy, Québec, Canada
G1K 7P4

MARS 2004

Modal Kleene Algebra and Applications — A Survey

Jules Desharnais[1] Bernhard Möller[2] Georg Struth[2]

[1] Département d'informatique et de génie logiciel, Université Laval,
Québec QC G1K 7P4 Canada
Jules.Desharnais@ift.ulaval.ca

[2] Institut für Informatik, Universität Augsburg,
Universitätsstr. 14, D-86135 Augsburg, Germany
{moeller,struth}@informatik.uni-augsburg.de

Abstract Modal Kleene algebras are Kleene algebras with forward and backward modal operators defined via domain and codomain operations. They provide a concise and convenient algebraic framework that subsumes various other calculi and allows treating quite a variety of areas. We survey the basic theory and some prominent applications. These include, on the system semantics side, Hoare logic and PDL (Propositional Dynamic Logic), wp calculus and predicate transformer semantics, temporal logics and termination analysis of rewrite and state transition systems. On the derivation side we apply the framework to game analysis and greedy-like algorithms.

1 Introduction

Kleene algebras are fundamental structures in computer science, with applications ranging from program development and analysis to rewriting theory and concurrency control. Initially conceived as algebras of regular events [28], they have recently been extended in several directions. The first direction includes omega algebra, which is a Kleene algebra with an additional operator for infinite iteration [8], demonic refinement algebra [51] and lazy Kleene algebra [33]. The second direction adds tests to Kleene algebra [29]. This allows reasoning about regular programs. All these variants are similar to relational reasoning. Most of them offer a nice balance between economy in concepts and proofs and algorithmic power. The equational theory of Kleene algebra, for instance, can be decided by automata. The third direction, in contrast, is modal in spirit. Here Kleene algebra is combined with a Boolean algebra in a module-based approach [17], the scalar product modelling the application of a modal operator to a state. This yields a calculus that is very similar to certain algebraic approaches to propositional dynamic logics.

The modal and the relational approaches to reasoning about programs and state transition systems have been reconciled in Kleene algebra with domain [12]. The three simple equational domain axioms open a new door: they allow the definition of modal operators semantically via abstract image and preimage operations. But expressions that mention modalities can still in many cases be reduced to pure Kleene algebra with tests. This preserves the algorithmic power of the latter and also provides a very symmetric approach to reasoning about actions and propositions or transitions and states. Compared with relation algebra, modal Kleene algebra does not need the full power of a complete Boolean algebra as the carrier set, of full additivity of sequential composition, of a converse operation and of residuation; it can make do with very lean versions of these concepts only.

In this survey, we discuss modal Kleene algebras both from the theoretical and the practical point of view. On the theoretical side, we introduce the main concepts and review the most important facets of a calculus. Modal Kleene algebras are mathematically quite simple: for actions, they provide only the regular operations of addition, multiplication and reflexive transitive closure; for propositions we have a Boolean algebra. Through their combination via modalities, they become expressive enough for a variety of applications. We also try to point

out that our algebraic approach to modal reasoning provides some advantages over a logical one. Algebra in general is particularly suited for structuring and abstracting. Here, structure is imposed via symmetries and dualities, for instance in terms of Galois connections. Abstraction is provided, for instance, by lifting modal expressions to the algebras of modal operators, which are again algebraically well-behaved. This often allows a very brief and concise point-free style of reasoning. We will also see that exploiting modal correspondences, switching between relational and modal reasoning, can be very simple in modal Kleene algebra. In some cases, there is a one-to-one translation between modal and relational proofs. This is interesting in particular when relational reasoning is visualized by diagrams.

On the practical side, we show that modal Kleene algebra may serve both for giving abstract program semantics and as a unifying tool that subsumes various popular program calculi and hence admits cross-theory reasoning. Here, we show that both the weakest liberal precondition semantics and the weakest precondition semantics can be modelled in modal Kleene algebra. It therefore supports both partial and total program correctness. We also show how modal Kleene algebra induces predicate transformer algebras with convenient properties. In the field of calculi, we present subsumption and completeness results for (propositional) Hoare logic, propositional dynamic logic and temporal logics.

In the field of system development we show that our combination of relational and modal reasoning can be applied to reasoning about greedy algorithms, in modelling termination conditions, to analysing games and in reconstructing a considerable part of the theory of abstract rewriting in a simple and convenient way. Since we do not consider equational rewriting but its non-symmetric extension [46], our results are immediately relevant to concurrent systems that interact via commutation or semi-commutation properties.

Although the aim of this paper is to form an overall picture of the usefulness of modal Kleene algebra, we do not claim completeness of our survey. E.g., an approach to pointer analysis based on Kleene algebra [16], though highly relevant, will not be treated here. Many of the results presented here have appeared elsewhere, and we just quote the original papers which should be consulted for full details. However, there is also a substantial amount of new material.

Modal Kleene algebra is a quite recent development. Although we believe that the core of the theory is now well understood and the examples outlined below point out its universality and practical relevance, still many questions are open. In particular as far as applications are concerned, we feel that we have so far only scratched the surface.

The remainder of this text is organised as follows. Section 2 introduces modal semirings. Section 3 shows game analysis as a first application of modal semirings. Section 4 extends modal semirings to Kleene algebra with domain and to modal Kleene algebras. Section 5 relates the approach to propositional dynamic logic and its relatives. Section 6 shows how partial and total correctness of regular programs can be modelled in modal Kleene algebra. Section 7 lifts modal Kleene algebra to predicate transformer algebras. Section 8 relates the approach with temporal logics. Section 9 reconstructs various results from the area of termination analysis, including properties of abstract rewrite systems. Section 10 discusses connections to modal correspondence theory. Section 11 develops a generic greedy-like algorithm. Section 12 summarises the applications and points out further directions for the approach.

2 Domain Semirings and Modalities

2.1 Test Semirings and Domain

A *semiring* is a structure $(K, +, \cdot, 0, 1)$ such that $(K, +, 0)$ is a commutative monoid, $(K, \cdot, 1)$ is a monoid, multiplication distributes over addition from the left and right and zero

is a left and right annihilator, i.e., $a0 = 0 = 0a$ for all $a \in K$ (the operation symbol $\cdot$ is omitted here and in the sequel). The semiring is *idempotent* if it satisfies $a + a = a$ for all $a \in K$. Then K has a *natural ordering* $\leq$ defined for all $a, b \in K$ by $a \leq b$ iff $a + b = b$. It induces a semilattice with $+$ as join and 0 as the least element; addition and multiplication are isotone with respect to the natural ordering.

In many contexts these operations can be interpreted as follows:

$$\begin{array}{lcl} + & \leftrightarrow & \text{choice,} \\ \cdot & \leftrightarrow & \text{sequential composition,} \\ 0 & \leftrightarrow & \text{abortion,} \\ 1 & \leftrightarrow & \text{identity,} \\ \leq & \leftrightarrow & \text{increase in information or in choices.} \end{array}$$

Programs and state transition systems can be described in a bipartite world in which propositions describe sets of states and actions or events model transitions between states. Propositions live in a Boolean algebra and actions in an idempotent semiring with the operations interpreted as above. In fact, to model regular programs, an additional operation of iteration or reflexive transitive closure is required. The corresponding extension of semirings to Kleene algebras is described in Section 4. In many formalisms, propositions and actions cooperate via modal operators that view actions as mappings on propositions in order to describe state-change and via test operators that embed propositions into actions in order to describe measurements on states and to model the usual program constructs.

To motivate this modal view, let a semiring element a describe an action or abstract program and a test p a proposition or assertion, also called a *test*. Then pa describes a restricted program that acts like a when the initial state satisfies p and aborts otherwise. Symmetrically, ap describes a restriction of a in its possible final states. We now introduce an abstract domain operator $\ulcorner$ that assigns to a the test that describes precisely its enabling states. In combination with restriction, the domain operation yields an abstract preimage operation. This provides the semantic basis for defining modalities.

Let us now axiomatise the corresponding notions. A *Boolean algebra (*BA*)* is a complemented distributive lattice. By overloading, we usually write $+$ and $\cdot$ also for the Boolean join and meet operation and use 0 and 1 for the least and greatest elements of the lattice. The symbol $\neg$ denotes the operation of complementation. We will consistently use the letters $a, b, c, \ldots$ for semiring elements and $p, q, r, \ldots$ for Boolean elements. We will freely use the concepts and laws associated with Boolean algebra, including relative complement $p - q = p \sqcap \neg q$ and implication $p \rightarrow q = \neg p + q$.

A *test semiring* is a two-sorted structure $(K, \mathsf{test}(K))$, where K is an idempotent semiring and $\mathsf{test}(K) \subseteq K$ is a Boolean algebra embedded into K such that the operations of $\mathsf{test}(K)$ coincide with the restrictions of the operations of K to $\mathsf{test}(K)$. In particular, $p \leq 1$ for all $p \in \mathsf{test}(K)$. But in general, $\mathsf{test}(K)$ is only a subalgebra of the subalgebra of all elements below 1 in K.

A *semiring with domain* [12] (a $\ulcorner$-semiring) is a structure $(K, \ulcorner)$, where K is an idempotent semiring and the *domain operation* $\ulcorner\colon K \rightarrow \mathsf{test}(K)$ satisfies for all $a, b \in K$ and $p \in \mathsf{test}(K)$

$$a \leq (\ulcorner a)a, \tag{d1}$$

$$\ulcorner(pa) \leq p. \tag{d2}$$

Let us explain these axioms. As in the algebra of relations, multiplication with a test from the left or right means domain or range restriction, respectively. Now first, since $\ulcorner a \leq 1$ by $\ulcorner a \in \mathsf{test}(K)$, isotonicity of multiplication shows that the first axiom can be strengthened to an equality expressing that restriction to the full domain is no restriction at all. The second

axiom means that after restriction the remaining domain must satisfy the restricting test. An important consequence of the axioms is that $\ulcorner$ preserves arbitrary existing suprema [35].

To further explain (d1) and (d2) we note that their conjunction is equivalent to each of

$$\ulcorner a \leq p \Leftrightarrow a \leq pa, \tag{llp}$$

$$\ulcorner a \leq p \Leftrightarrow \neg pa \leq 0, \tag{gla}$$

which constitute elimination laws for $\ulcorner$. (llp) says that $\ulcorner a$ is the least left preserver of a. (gla) says that $\neg\ulcorner a$ is the greatest left annihilator of a. Both properties obviously characterize domain in set-theoretic relations.

Because of (llp), domain is uniquely characterised by the two domain axioms. Moreover, if $\mathsf{test}(K)$ is complete then a domain operation always exists. If $\mathsf{test}(K)$ is not complete, this need not be the case.

A prominent example of a domain semiring is the algebra REL of concrete homogeneous binary relations over some set. There the domain operation is given by $\ulcorner R = R\,;R^\smile \cap I$, where I is the identity relation, $R^\smile$ is the converse of R and ; is relational composition.

Further important domain semirings are the algebra PAT of path sets in a directed graph (see e.g. [32]) and Kleene's original algebra of formal languages, the latter ones being not very interesting, because its test algebra is *discrete*, i.e., consists of 0 and 1 only.

Many natural properties follow from the axioms. Domain is uniquely defined. It is strict ($\ulcorner a = 0 \Leftrightarrow a = 0$), additive ($\ulcorner(a+b) = \ulcorner a + \ulcorner b$), isotone ($a \leq b \Rightarrow \ulcorner a \leq \ulcorner b$), stable on tests ($\ulcorner p = p$) and satisfies the import/export law ($\ulcorner(pa) = p\ulcorner a$). Finally, domain commutes with all existing suprema. See [12] for further information.

2.2 Modal Semirings

A domain semiring is called *modal* if additionally it satisfies

$$\ulcorner(a\ulcorner b) \leq \ulcorner(ab). \tag{d3}$$

This axiom serves to make composition of multimodal operators below well-behaved. In a modal semiring, domain is *local* in the following sense:

$$\ulcorner(ab) = \ulcorner(a\ulcorner b).$$

Without (d3), only the inequality $\ulcorner(ab) \leq \ulcorner(a\ulcorner b)$ holds. The additional axiom (d3) guarantees that the domain of ab is independent from the inner structure of b or its codomain; information about the domain of b in interaction with a suffices.

A codomain operation $\urcorner$ can easily be defined as a domain operation in the opposite semiring, where, as usual in algebra, opposition just swaps the order of multiplication. We call a semiring K with local domain and codomain simply a *modal semiring*.

Let K be a modal semiring. We can now introduce forward and backward diamond operators by modelling their semantics as abstract preimage and image operations:

$$|a\rangle p = \ulcorner(ap), \qquad\qquad \langle a|p = (pa)\urcorner, \tag{1}$$

for all $a \in K$ and $p \in \mathsf{test}(K)$. Let us explain why this definition is adequate. For a state transition system a, the term ap restricts a to that part for which all final states satisfy p. Then $\ulcorner(ap)$ selects all starting states of this remaining part; they indeed form the inverse image of p under a. Symmetric arguments apply to the backward diamond.

The definition implies that the diamond operators are strict additive mappings (or *hemimorphisms*) on the algebra of tests. Hence for arbitrary index set I and arbitrary family

$(a_i)_{i\in I}$ of elements $a_i \in K$ the structures $(\mathsf{test}(K), \{|a_i\rangle : i \in I\})$ and $(\mathsf{test}(K), \{\langle a_i| : i \in I\})$ are Boolean algebras with operators à la Jónsson and Tarski [25]. Such structures are called *modal algebras* in [20].

Duality with respect to opposition transforms forward diamonds into backward diamonds and vice versa. It follows that they satisfy an *exchange law*, a weak analogue of the relational Schröder law. For all $a \in K$ and $p, q \in \mathsf{test}(K)$,

$$|a\rangle p \leq \neg q \Leftrightarrow \langle a|q \leq \neg p. \tag{2}$$

De Morgan duality transforms diamonds into boxes and vice versa:

$$|a]p \stackrel{\text{def}}{=} \neg|a\rangle\neg p, \qquad\qquad |a\rangle p \;=\; \neg|a]\neg p.$$

It follows that diamonds and boxes are lower and upper adjoints of Galois connections:

$$|a\rangle p \leq q \Leftrightarrow p \leq [a|q, \qquad\qquad \langle a|p \leq q \Leftrightarrow p \leq |a]q, \tag{3}$$

for all $a \in K$ and $p, q \in \mathsf{test}(K)$. The Galois connections are useful as theorem generators and the dualities as theorem transformers.

The above-mentioned import/export law entails

$$p\,(|a\rangle q) = |pa\rangle q, \qquad\qquad p\,(\langle a|q) = \langle ap|q. \tag{4}$$

The modal axiom (d3) implies

$$|ab\rangle p = |a\rangle|b\rangle p, \qquad \langle ab|p = \langle b|\langle a|p, \qquad |ab]p = |a]|b]p, \qquad [ab|p = [b|[a|p. \tag{5}$$

Thus multiplication acts covariantly on forward modalities and contravariantly on backward ones. In the sequel, when the direction of diamonds and boxes does not matter, we will use the notation $\langle a\rangle$ and $[a]$.

If $\mathsf{test}(K)$ is complete then $\ulcorner$ always exists; moreover, since it commutes with all suprema, it has a unique upper adjoint which is $\urcorner$. So in this case, the modal algebra is completely characterised by the domain axioms and the Galois connection. If $\mathsf{test}(K)$ is not complete, this need not be the case. In the non-complete case, diamonds (boxes) commute with all existing suprema (infima) of the test algebra. These and further properties are inherited from those of domain. Further useful properties are immediate from the Galois connection. They include cancellation laws and isotonicity and antitonicity properties for modalities. Of particular interest are the following demodalisation laws that follow from the domain elimination law (gla) and its dual for codomain.

$$|a\rangle p \leq q \Leftrightarrow \neg qap \leq 0, \qquad\qquad \langle a|p \leq q \Leftrightarrow pa\neg q \leq 0. \tag{6}$$

For a test p we have

$$\langle p\rangle q = pq, \qquad [p]q = p \to q. \tag{7}$$

Hence, $\langle 1\rangle = [1]$ is the identity function on tests. Moreover, $\langle 0\rangle p = 0$ and $[0]p = 1$.

To set up the connection to relational algebra, we define a *modal semiring with converse* to be a modal semiring K with an additional operation $\_^\smile : K \to K$ that is an involution, distributes over addition, is the identity on tests and is contravariant with respect to multiplication. One can show (see again [12]) that over a modal semiring with converse the axioms (d1) and (d2) imply the Galois connection

$$|a^\smile\rangle p \leq q \Leftrightarrow p \leq |a]q. \tag{8}$$

It follows that in a modal semiring with converse all predicate transformers $\lambda p\,.\,|a\rangle p$ are universally disjunctive, i.e., preserve all existing suprema (and that all predicate transformers $\lambda p\,.\,|a]p$ are universally anti-disjunctive, i.e., transform all existing suprema into corresponding infima). This generalizes to modal semirings, since there one can replace $|a^\smile\rangle$ by the backward diamond of a. Therefore in a modal semiring with converse $^\smile$ we have

$$|a^\smile\rangle = \langle a|, \qquad |a^\smile] = [a|. \tag{9}$$

3 Two-Player Game Analysis

3.1 Introduction

To illustrate what we can already achieve with modal semirings, we take up part of the two-player game analysis in [42] and [3]. Such a game is given by a set of positions with a binary relation describing the admissible moves. A position is terminal if it does not have a successor under the move relation. The two players take turns. A player whose turn it is but who is in a terminal position has lost the game. There are no special assumptions about positions and moves; in particular, the move relation need not be Noetherian.

The aim is to characterize positions that mean guaranteed win (under optimal play) or guaranteed loss (even under optimal play) and to compute a winning strategy if possible. We do not focus particularly on computing a winning strategy, which will nevertheless come as a byproduct from an algorithm for iteratively computing the winning and losing positions. The following conditions are obvious:

- Every terminal position is a losing position.
- A position is a losing position iff all moves from it lead to winning positions (for the opponent), because then there is no possibility to prevent her from winning (provided she plays optimally).
- A position is a winning position iff at least one move from it leads to a losing position (for the opponent), because the player may move into that position and hence force the opponent to lose.

We abstract from the relational case and represent the move relation as an element a of a modal semiring. Moreover, we want to represent terminal, winning and losing positions by semiring tests t, w and l. We obtain t from a as $t = \neg\lceil a$, whereas w and l have yet to be determined. To this end we rewrite the above informal conditions into modal notation:

$$\begin{aligned} &t \leq l, \\ &l = |a]\, w, &(10)\\ &w = |a\rangle\, l. &(11)\end{aligned}$$

Conditions (10) and (11) are mutually recursive. Separate them by substitution yields

$$l \;=\; |a]\,|a\rangle\, l, \qquad w \;=\; |a\rangle\,|a]\, w.$$

What kind of solutions do these recursive equations have?

3.2 Existence of Solutions: Fixpoints of Dual Functions

We define the functions

$$f(p) \stackrel{\text{def}}{=} |a]\,|a\rangle\, p, \qquad g(p) \stackrel{\text{def}}{=} |a\rangle\,|a]\, p.$$

By the properties of $\langle\_\rangle$ and $[\_]$ both functions are isotone. We now assume that in the underlying modal semiring K the sublattice $\mathsf{test}(K)$ is complete. Then, by the Knaster/Tarski fixpoint theorem, f and g each have both a least and a greatest fixpoint. To investigate their relation we recall that two functions $h, k : M \to M$ on a Boolean lattice $(M, \leq)$ are *(de Morgan) duals* if for all $x \in M$

$$h(x) = \neg k(\neg x).$$

By definition, the functions $|a\rangle$ and $|a]$ are dual, and a quick calculation shows that the above functions f and g are dual as well. Now we can use the following result (see e.g. [38]).

Lemma 3.1 *Let h, k be dual functions over a Boolean lattice such that μ_h, μ_k and ν_h, ν_k exist. Then*

$$\mu_h = \neg\nu_k, \qquad \mu_k = \neg\nu_h.$$

From this it is immediate that μ_h, μ_k and $z \stackrel{\text{def}}{=} \neg(\mu_h \sqcup \mu_k) = \nu_h \sqcap \nu_k$ form a partition of the lattice, i.e., if $\bot$ and $\top$ are the least and greatest element then

$$\mu_h \sqcap \mu_k = \mu_h \sqcap z = \mu_k \sqcap z = \bot,$$
$$\mu_h \sqcup \mu_k \sqcup z = \top.$$

Likewise, ν_h, ν_k and $\neg(\nu_h \sqcup \nu_k) = \mu_h \sqcap \mu_k$ form a partition of the lattice.

Let us apply this to our dual functions f and g above. The set of positions is to be partitioned into winning, losing and tie positions. By the above observation there are two possible choices: either $l = \mu_f$ as the set of losing positions and $w = \mu_g$ as the set of winning positions, or $l = \nu_f$ and $w = \nu_g$.

In [3] it is shown that the first of these choices is the adequate one. The remainder $\nu_f \sqcap \nu_g = \nu_f\nu_g$ represents the set of tie positions, i.e., the set of positions from which under optimal play of both opponents none will reach a winning or losing position. Note that from a tie position there needs to emanate at least one infinite path in the game graph; if the set of positions is finite, this path necessarily has to be cyclic.

One has to ensure that the (separately found) solutions $l = \mu_f$ and $w = \mu_g$ also satisfy the original mutual recursion

$$l = |a]\, w \qquad w = |a\rangle\, l$$

(which need not be the case for arbitrary fixpoints of f and g). This can be done by the rolling rule of fixpoint calculus (see again [38]).

3.3 Iterative Computation of Win/Lose

We now want to obtain an algorithm for actually computing the winning and losing positions. For this we remember Kleene's fixpoint theorem, the proof of which shows the following:

Lemma 3.2 *Let $h : M \to M$ be an isotone function on a complete lattice $(M, \leq)$. Then*

$$\mathsf{sup}\,\{h^i(\bot) : i \in \mathbb{N}\} \leq \mu_h.$$

So let us consider the first steps of the fixpoint iteration for μ_f and μ_g. In the semiring setting we have $\bot = 0$; moreover, let $t = \neg\ulcorner a$ again be the set of terminal positions.

$$\begin{array}{rcl@{\qquad}rcl}
f^1(0) &=& |a]\,|a\rangle\,0 = |a]\,0 & g^1(0) &=& |a\rangle\,|a]\,0 = |a\rangle\,t \\
&=& \neg\ulcorner a = t & &=& |a\rangle\,(f^1(0)) \\
f^2(0) &=& f(f^1(0)) = f(t) & g^2(0) &=& g(g^1(0)) = g(|a\rangle\,t) \\
&=& |a]\,|a\rangle\,t = |a]\,(g^1(0)) & &=& |a\rangle\,|a]\,|a\rangle\,t = |a\rangle\,(f^2(0)) \\
&\vdots & & &\vdots & \\
f^{i+1}(0) &=& |a]\,(g^i(0)) & g^{i+1}(0) &=& |a\rangle\,(f^{i+1}(0))
\end{array}$$

This can be explained informally as follows. The set $f^1(0)$ of losing positions of "order 1" is the set t of terminal positions. The set $g^1(0)$ of winning positions of "order 1" consists of all immediate predecessors of t. The set $f^{i+1}(0)$ of losing positions of "order $i+1$" consists of the positions all successors of which are winning positions of "order i", the set $g^{i+1}(0)$ of winning positions of "order $i+1$" consists of the positions that have at least one losing position of "order $i+1$" as a successor.

Hence the fixpoint iteration describes the following algorithm.

1. Start with the terminal positions which are marked as losing positions.
2. Then traverse the game graph backwards while adapting the markings according to the above equations.

But what about termination of the algorithm? And under which circumstances does it really reach the least fixpoints $l = \mu_f$ and $w = \mu_g$? It is obvious that for an infinite set of positions there always will be games for which the algorithm doesn't terminate. So we now restrict our attention to games with finitely many positions. This can abstractly be reflected by considering only modal semirings K in which all chains in $\mathsf{test}(K)$ are finite, so that all isotone functions are also continuous and hence the fixpoint iteration yields the desired result. It can be stopped as soon as it gets stationary, i.e., as soon as a fixpoint has been reached. Recording in every iteration step which moves lead into winning or losing positions, one also obtains all possible winning strategies.

The basic fixpoint iteration algorithm reads as follows:

$r := 0$;
$\{\,\mathsf{inv}\ r \leq f(r) \wedge r \leq \mu_f\}$
$\mathsf{while}\ (f(r) \neq r)$
  $\mathsf{do}\ \ r := f(r)$;
  $\mathsf{od}\ \ \{r = \mu_f\}$

The least fixpoint $\mu_g = w$ then results as $w = |a\rangle\, l$.

3.4 Efficiency Improvement

Let us now show that the algebra of modal semirings is also very useful in formal transformation of the basic algorithm into more efficient (but much less understandable) versions.

The main technique employed is that of *formal differentiation* or *strength reduction* (see e.g. [39]), where expensive recomputation of a quantity in every step of an iteration is replaced by computation of the increments between the values of that quantity. By their many distributive laws, modal semirings are an ideal setting for this technique.

In the algorithm above, we first introduce an auxiliary variable s that always has the value $f(r)$ and is incremented correspondingly:

$r := 0\,; s := f(0)$;
$\{\,\mathsf{inv}\ s = f(r) \wedge r \leq s \wedge r \leq \mu_f\}$
$\mathsf{while}\ (s \neq r)$
  $\mathsf{do}\ \ (r,\, s) := (s,\, f(s))$;
  $\mathsf{od}\ \ \{r = \mu_f\}$

Because of $r \leq s$ we have $s = r + (s - r)$ and $s \neq r \;\Leftrightarrow\; s - r \neq 0$. (This only needs isotonicity of f.) To simplify the assignment $s := f(s)$ we have to consider the special form of f. We obtain

$$f(s) = f(r + (s - r)) \;=\; |a]\,|a\rangle\,(r + (s - r)) = |a]\,(|a\rangle\, r + |a\rangle\,(s - r)). \qquad (*)$$

Now we set $u = |a\rangle\, r$ and examine $|a]\,(u + x)$ for arbitrary x:

$$|a]\,(u + x) = \neg^{\Gamma}(a\neg(u + x)) = \neg^{\Gamma}(a\neg u\neg x) = |a\neg u]\, x.$$

If we now carry the part $a\neg u$ in a variable m, the assignment $s := f(s)$ simplifies to $s := |m]\, x$ with $x = |a\rangle\, (s - r)$. Our new invariant reads

$$\{\,\mathsf{inv}\ s = f(r) \ \wedge\ r \leq s \ \wedge\ r \leq \mu_f \ \ \wedge\ u = |a\rangle\, r \ \wedge\ m = a\neg u\}$$

This is established by the initialisation

$$u := 0\,;\, m := a\ ;$$

How can we maintain it?

The calculation $(*)$ shows that after the assignment $r := s$ variable u has to have the new value $u + x$, so m needs the new value

$$a\neg(u + x) = a\neg u\neg x = m\neg x$$

This yields

$$\begin{array}{l} r := 0\,;\, s := f(0)\ ; \\ u := 0\,;\, m := a\ ; \\ \{\,\mathsf{inv}\ s = f(r) \ \wedge\ r \leq s \ \wedge\ r \leq \mu_f \ \ \wedge\ u = |a\rangle\, r \ \wedge\ m = a\neg u\} \\ \mathsf{while}\ (s - r \neq 0) \\ \quad \mathsf{do}\ \ \mathsf{let}\ x = |a\rangle\, (s - r) \\ \qquad\ \ \mathsf{in}\ \ (r,\, s,\, u,\, m)\ :=\ (s,\, |m]\, x,\, u + x,\, m\,\neg x)\ ; \\ \quad \mathsf{od}\ \ \{r = \mu_f \ \ \wedge\ u = \mu_g\} \end{array}$$

The simultaneous assignment can be sequentialised from left to right.

Our final improvement results from examining the expressions involving m, notably $|m]\, x = \neg\lceil(m\,\neg x)$. Since we need $n \stackrel{\text{def}}{=} m\,\neg x$ anyway, it makes sense to compute n and $\neg\lceil n$ simultaneously.

For this we maintain a new variable d that always contains $\neg\lceil m$. It is initialised to $\neg\lceil a$ and is incrementally adjusted using a vector of out-degrees:

1. Loop through x.
2. For each position $p \in x$ and every predecessor q of p under m:
3. decrease q's outdegree by 1 and remove the edge from q to p.
4. If the outdegree of q becomes 0 that way, add q to d.

Of course, all these steps can be done algebraically.

4 Modal Kleene Algebras

While modal semirings suffice for some applications, others require an explicit notion of iteration. This is provided by extending idempotent semirings to Kleene algebras.

A *Kleene algebra* [28] is a structure $(K, {}^*)$ such that K is an idempotent semiring and the *star* * satisfies, for $a, b, c \in K$, the *unfold* and *induction laws*

$$1 + aa^* \leq a^*, \tag{$*$-1}$$

$$1 + a^*a \leq a^*, \tag{$*$-2}$$

$$b + ac \leq c \Rightarrow a^*b \leq c, \tag{$*$-3}$$

$$b + ca \leq c \Rightarrow ba^* \leq c. \tag{$*$-4}$$

Therefore, a^* is the least pre-fixpoint and the least fixpoint of the mappings $\lambda x.ax + b$ and $\lambda x.xa + b$. The star is isotone with respect to the natural ordering.

Two important properties that follow from these axioms are the laws

$$ba \leq ac \ \Rightarrow\ b^*a \leq ac^*, \qquad ab \leq ca \ \Rightarrow\ ab^* \leq c^*a. \tag{12}$$

A *Kleene algebra with tests (*KAT*)* [29] is a test semiring $(K, \mathsf{test}(K))$ such that K is a KA. For all $p \in \mathsf{test}(K)$ we have that $p^* = 1$.

In a KAT one can give (angelic) abstract semantics of regular programs as follows:

$$\begin{aligned}
\mathsf{abort} &\stackrel{\text{def}}{=} 0 \\
\mathsf{skip} &\stackrel{\text{def}}{=} 1 \\
a \,[]\, b &\stackrel{\text{def}}{=} a + b \\
a \,;\, b &\stackrel{\text{def}}{=} ab \\
\mathsf{if}\ p\ \mathsf{then}\ a\ \mathsf{else}\ b &\stackrel{\text{def}}{=} pa + \neg pb \\
\mathsf{assert}\ p &\stackrel{\text{def}}{=} \mathsf{if}\ p\ \mathsf{then}\ \mathsf{skip}\ \mathsf{else}\ \mathsf{abort} \quad = \quad p \\
\mathsf{while}\ p\ \mathsf{do}\ a &\stackrel{\text{def}}{=} (pa)^* \neg p
\end{aligned}$$

Note that full Kleene algebra is needed for modelling loops, i.e., iteration. The definition of assert p via if then else is the usual one from assertion macro packages in programming languages like *C* or *Java*; algebraically it simplifies to p alone.

If the underlying test semiring of a KAT K is a domain (codomain), we speak of a KA *with domain (codomain)*, briefly ⌜-(¬-)KA. Finally, a *modal Kleene algebra (*MKA*)* is a KAT in which the underlying test semiring is modal.

Examples of MKAs are again REL and PAT.

Using the star induction axioms, one can show the following induction principle for the diamond operator (cf. [12]):

$$|a\rangle p + q \leq p \;\Rightarrow\; |a^*\rangle q \leq p. \tag{13}$$

5 Kleene Modules and PDL

Most previous algebraic approaches to modeling programs or state transition systems show an asymmetric treatment of propositions and actions. On the one hand, propositional dynamic logic (PDL) [21] and its algebraic relatives dynamic algebras [27,36,41] and test algebras [36,41,49] are proposition-based. Dynamic algebra has only modalities, test algebra also has propositions. Most axiomatisations do not even contain explicit axioms for actions: their algebra is only implicitly induced via the definitions of the modalities. On the other hand, KAT has both actions and propositions, but, complementarily to dynamic algebra, it lacks modalities, i.e., the possibility to combine actions and propositions into new propositions. Therefore, reasoning about actions in dynamic algebra and test algebra and about propositions in KAT is indirect and restricted.

These rather artificial asymmetries and limitations have already been questioned by Pratt [41], but persisted for several decades. They are overcome in MKA in a very smooth and simple way; therefore MKA provides an algebraic alternative to PDL that supports both proposition- and action-based reasoning and admits both tests and modalities. In a more abstract sense, MKA reconciles relational and modal reasoning about programs. However, the defining axioms of MKA are quite different from and more economic than those of dynamic algebra and test algebra. We will now briefly describe the precise relation between MKA and PDL and its algebraic relatives. This can best be done by introducing an additional intermediate structure which we call a *Kleene module*. Kleene modules are on the one hand straightforward adaptations of the standard modules of algebra that allow us to introduce modal operators via scalar products. On the other hand, the coupling between actions and propositions in Kleene modules is not as tight as in modal Kleene algebra.

5.1 Definition of Kleene Modules

Kleene modules are natural variants of the usual modules from algebra [24], where the ring is replaced by a Kleene algebra and the Abelian group by a Boolean algebra. Certain variants of Kleene modules have already been studied in [5,30].

Definition 5.1. *A Kleene left-module is a two-sorted algebra* $(K, \mathsf{test}(K), :)$*, where* $K \in \mathsf{KA}$ *and* $B \in \mathsf{BA}$ *and where the left scalar product* $:$ *is a mapping* $K \times B \to B$ *such that for all* $a, b \in K$ *and* $p, q \in B$*,*

$$a : (p + q) = a : p + a : q, \tag{km1}$$
$$(a + b) : p = a : p + b : p, \tag{km2}$$
$$(ab) : p = a : (b : p), \tag{km3}$$
$$1 : p = p, \tag{km4}$$
$$0 : p = 0, \tag{km5}$$
$$q + a : p \leq p \Rightarrow a^* : q \leq p. \tag{km6}$$

As usual, we do not distinguish between the Boolean and Kleenean zeros and ones. KM_l denotes the class of Kleene left-modules. In accordance with the relation-algebraic tradition, we also call the scalar products of KM_l *Peirce products*.

Axioms of the form (km1)–(km4) also occur in algebra. For rings, an analogue of (km5) is redundant, whereas for semirings — in absence of inverses — it is independent. Axiom (km6) is of course beyond ring theory. It is the star induction rule ($*$-3) with the semiring product replaced by the Peirce product and the sorts of elements adjusted, i.e., b and c replaced by Boolean elements.

Analogously to the situation for domain and codomain we define *Kleene right-modules* as Kleene left-modules on the opposite semiring. A *Kleene bimodule* is a Kleene left-module that is also a Kleene right-module. We will henceforth consider only Kleene left-modules.

5.2 Calculus of Kleene Modules

The relation between Kleene left-modules and modal Kleene algebra is straightforward.

Proposition 5.2. *Let* K *be a modal Kleene algebra. Setting* $a : p = |a\rangle p$*, the structure* $(K, \mathsf{test}(K), :)$ *is a Kleene left-module.*

The left-module axioms and also the right-module axioms can easily be shown to be theorems of modal Kleene algebra. Consequently, these axioms establish further properties of MKA in a well-structured way.

We first present some further properties that do not mention the star. The scalar product is right-strict, i.e., $a : 0 = 0$ and left- and right-isotone. It is disjunctive, $a : (pq) \leq (a : p)(a : q)$ and satisfies

$$a : p - a : q \leq a : (p - q).$$

The following variants of the star unfold laws ($*$-1) and ($*$-2) hold.

$$p + a : (a^* : p) = a^* : p, \qquad p + a^* : (a : p) = a^* : p. \tag{14}$$

Therefore, of course, they do not have to be explicitly added to the module axioms. Finally, the module axiom (km6), which is a quasi-identity, is equivalent to the identity

$$a^* : p - p \leq a^* : (a : p - p). \tag{15}$$

This identity appears in PDL (cf. [21]), but also in axiomatisations of temporal logics as an induction law. In [17], we present various additional properties that all can easily be translated to theorems of PDL.

5.3 Relatives of Kleene Modules

We now position the Kleene modules within the context of Kleene algebra with domain and algebraic variants of propositional dynamic logic.

First, the class of *dynamic algebras* [41] can be obtained as a variant of Kleene modules by requiring, instead of a Kleene algebra, an absolutely free algebra of Kleenean signature (without 0 and 1), by removing (km4) and (km5), by adding right-strictness and the star unfold law of (14) and by replacing (km6) by (15). Consequently, the algebra of actions is implicitly axiomatised in dynamic algebra. We call a dynamic algebra or Kleene module *extensional* if

$$\forall\, p\,.\,(a : p \leq b : p) \Rightarrow a \leq b.$$

This property is independent from the module axioms. The relation induced by the left-hand side of this quasi-identity is a precongruence on Kleene modules. It can also be interpreted as a notion of *observational equivalence*. Intuitively, it is a point-wise measurement of the behaviour of the actions. In the extensional case, the action is completely determined by this scanning.

The following result shows that Kleene modules subsume dynamic algebras.

Theorem 5.3. *Every Kleene module is a dynamic algebra.*

Moreover, Kleene modules yield an optimal representation of equational reasoning about Kripke frames.

Theorem 5.4. *The equational theories of extensional Kleene modules and extensional dynamic algebras coincide.*

Hence, these equational theories coincide with that of finite Kripke structures.

Second, there are two extensions of dynamic algebras that also include tests. In Pratt's variant, the test axiom $p? : q = pq$ is added to the axioms of dynamic algebra, where ? models an embedding of tests into actions. Again, therefore, the Kleene algebra is implicitly defined. It is straightforward to show that modal Kleene algebra subsumes Pratt's variant of test algebra.

Proposition 5.5. *Every modal Kleene algebra is a test algebra à la Pratt.*

Note the notational economy inherited from KAT which makes the embedding operator ? implicit.

Hollenberg [23] has given a variant of test algebra which makes explicit use of the Kleene algebra axioms and also of the embedding operator ?. This test algebra subsumes Pratt's variant.

Theorem 5.6. *The classes of modal Kleene algebras and test algebras à la Hollenberg coincide.*

Theorem 5.7. *The equational theories of extensional modal Kleene algebras and finite Kripke structures coincide.*

It follows from results for test algebra that the equational theory of extensional modal Kleene algebra is EXPTIME-complete. Here the advantage of modal Kleene algebra over test algebra is its economy. It is defined via three axioms, whereas Hollenberg's test algebra has eight.

For further technical details as well as further discussion of related work we refer to [17].

6 Modelling Program Correctness

6.1 Hoare Logic and wlp

We now return to the Kleene semantics of simple while programs introduced in Section 4. As is well known, *partial program correctness* can be modelled using the weakest liberal precondition $\mathsf{wlp}(a, q) = |a]\, q$. Then a *Hoare triple* $\{p\}\ a\ \{q\}$ is *valid* if $p \leq |a]\, q$.

Kozen has shown that already in KAT one can formulate validity of $\{p\}\ a\ \{q\}$ as $pa\neg q = 0$. Although this allows proving soundness of the rules of propositional Hoare logic, i.e., Hoare logic without the assignment rule, the MKA formulation leads to still simpler and readable encodings of Hoare triples and rules and also to simpler and more concise soundness proofs. Moreover, in contrast to KAT, the MKA formulation also admits a simple, fully algebraic proof of relative completeness of propositional Hoare logic [35]. This is due to the fact that while backward modalities can be used for encoding Hoare triples, MKA also provides forward modalities for encoding their standard semantics.

Example 6.1 As an example consider the while-rule:

$$\frac{\{p \wedge q\}\ a\ \{q\}}{\{q\}\ \mathsf{while}\ p\,\mathsf{do}\,a\ \{\neg p \wedge q\}}$$

Its translation into MKA reads

$$pq \leq |a]\, q \ \Rightarrow\ q \leq |(pa)^*\neg p]\, (\neg p\, q).$$

Dualising to diamonds using (3) we obtain the version

$$\langle a|\, (pq) \leq q \ \Rightarrow\ \langle (pa)^*\neg p|\, q \leq \neg p\, q. \tag{16}$$

Now the soundness proof of this rule proceeds as follows:

$$\begin{aligned} \langle a|\, (pq) \leq q \ &\Leftrightarrow\ \langle pa|\, q \leq q \\ &\Rightarrow\ \langle (pa)^*|\, q \leq q \\ &\Rightarrow\ \neg p\, (\langle (pa)^*|\, q) \leq \neg p\, q \\ &\Leftrightarrow\ \langle (pa)^*\neg p|\, q \leq \neg p\, q \end{aligned}$$

The first step uses the definition of diamond twice, the second one induction (13), the third one isotonicity, the fourth one import/export (4). An even shorter proof is possible in predicate transformer algebra (see Section 7). □

The result of encoding Hoare rules and showing that they are theorems of modal Kleene algebra can be expressed as follows.

Theorem 6.1. *Propositional Hoare logic is sound with respect to the modal Kleene algebra semantics.*

Note that it is not surprising that this is possible in a formalism that subsumes KAT. However, it can be done in a much more succinct way. Therefore, the specialised syntax of Hoare logic can easily be abandoned in favour of the simple and more universal algebraic calculus of modal Kleene algebra.

Using the demodalisation rules of modal Kleene algebra that arise as generalisations of (llp) and (gla), there is, however, a simple translation of the modal encoding of Hoare rules into KAT. The resulting rules have a special shape and their validity can be decided by automata in PSPACE [9]. Thus the gain of expressiveness and flexibility introduced by MKA does not compromise the nice algorithmic properties of KAT.

Using the MKA encoding of the weakest liberal precondition semantics for Hoare logic, we can carry out an entirely algebraic and fully formal relative completeness proof of propositional Hoare logic. This proof is by far shorter than the standard textbook proofs that are based on set theory and usually leave many assumptions implicit.

Theorem 6.2. *Propositional Hoare logic is relatively complete for the partial correctness semantics of deterministic programs in modal Kleene algebra.*

6.2 Total Correctness and wp

For modelling *total correctness*, an MKA element a now receives the following interpretation: its domain $\ulcorner a$ represents the set of starting states for which all a-computations are guaranteed to terminate; a itself represents the set of all these computation paths [13]. Under this interpretation, the weakest precondition is given by

$$\mathsf{wp}(a,q) \stackrel{\text{def}}{=} \ulcorner a\, \mathsf{wlp}(a,q),$$

the refinement relation by

$$c \sqsubseteq a \quad \Leftrightarrow \quad \ulcorner a \leq \ulcorner c \wedge \ulcorner a\, c \leq a.$$

This entails the following properties of the angelic programming constructs:

$$\begin{aligned}
\mathsf{wp}(a,0) &= 0,\\
\mathsf{wp}(a,1) &= \ulcorner a,\\
\mathsf{wp}(\mathsf{abort},q) &= 0,\\
\mathsf{wp}(\mathsf{skip},q) &= q,\\
\mathsf{wp}(\mathsf{if}\ r\ \mathsf{then}\ a\ \mathsf{else}\ b,q) &= r\,\mathsf{wp}(a,q) + \neg r\,\mathsf{wp}(b,q),\\
\mathsf{wp}(a+b,q) &= \mathsf{wp}(a,q)\mathsf{wlp}(b,q) + \mathsf{wlp}(a,q)\mathsf{wp}(b,q).
\end{aligned}$$

The demonic programming constructs can be defined as follows:

- Demonic join (choice):
$$a \sqcup b = \ulcorner a \ulcorner b\,(a+b).$$
- Demonic meet: If $a \sqcap b$ exists and $\ulcorner(a \sqcap b) = \ulcorner a \ulcorner b$ then
$$a \sqcap b = (a \sqcap b) + \neg \ulcorner a\, b + \neg \ulcorner b\, a.$$
- Demonic composition:
$$a \mathbin{\square} b \stackrel{\text{def}}{=} ([a](\ulcorner b))\, ab.$$

A demonic redefinition of loop is also possible, see [13] for details. These definitions imply the following properties that all can be shown by fairly concise algebraic calculation. First, demonic refinement is the natural order associated with demonic choice, i.e., $a \sqsubseteq b \Leftrightarrow a \sqcup b = b$. Hence we have an upper semilattice (which is even complete if the underlying MKA is). Second, $\square$ distributes through $\sqcup$ in both arguments and hence is $\sqsubseteq$-isotone in both arguments. Third, demonic composition is associative.

In the semantic model just given, angelic choice is not $\sqsubseteq$-isotone. A generalised relationally based program semantics that integrates isotone angelic and demonic choice was presented in [14]. The idea is to model a program as a pair consisting of a transition relation between states and a subset of the domain of that relation from which no divergence is possible.

We again abstract to a modal Kleene algebra K and let the elements of K represent transition behaviour of programs, regardless of termination. Programs are then modelled by pairs (a,p) with $p \in \mathsf{test}(K)$ and $p \leq \ulcorner a$ includes termination information about the starting states of a. Then the essential program constructors are the following:

– Demonic composition: $(a,p)\mathbin{▫}(b,q) \stackrel{\text{def}}{=} (ab,\, p\,(|a]q))$.
– Demonic choice: $(a,p) \mathbin{⨅} (b,q) \stackrel{\text{def}}{=} (a+b,\, pq)$.
– Angelic choice: $(a,p) \mathbin{⨆} (b,q) \stackrel{\text{def}}{=} (a+b,\, p+q)$.

Then $▫$ is associative, has annihilator $(0,0)$, neutral element $(1,1)$ and distributes through $⨅$. Moreover, both choices are idempotent and associative. However, there is no neutral element w.r.t $⨅$, since the obvious candidate $(0,1)$ does not satisfy the restriction imposed on our pairs. So we do not have a full semiring structure. The refinement order is given by

$$(a,p) \sqsubseteq (b,q) \;\stackrel{\text{def}}{\Leftrightarrow}\; (a,p) ⨅ (b,q) = (b,q) \;\Leftrightarrow\; a \leq b \wedge q \leq p.$$

Both choice operators are isotone w.r.t. $\sqsubseteq$.

As an example for the use of the MKA laws in this setting, we prove associativity of composition. It is immediate that it suffices to consider the second components of the pairs, for which we calculate:

$$p\,|a](q\,|b]r) \;=\; p\,(|a]q)\,(|a]|b]r) \;=\; p\,(|a]q)\,(|ab]r).$$

The first step uses conjunctivity of $|a]$, the second one locality. The proofs of the other properties mentioned are slightly longer but again entirely straightforward calculations using the laws of modal Kleene algebra.

7 Beyond PDL: Predicate Transformer Algebras

Assume a test semiring $(K,+,,0,1)$. A *predicate transformer* is a function $f : \mathsf{test}(K) \to \mathsf{test}(K)$. It is *disjunctive* if $f(p+q) = f(p)+f(q)$ and *conjunctive* if $f(pq) = f(p)f(q)$. It is *strict* if $f(0) = 0$. Finally, id is the identity transformer and $\circ$ denotes function composition.

Let P be the set of *all* predicate transformers, M the set of isotone and D the set of strict and disjunctive ones. Under the pointwise ordering $f \leq g \stackrel{\text{def}}{\Leftrightarrow} \forall\, p\,.\, f(p) \leq g(p)$, P forms a lattice where the supremum $f+g$ and infimum $f \sqcap g$ of f and g are the standard pointwise liftings of $+$ and $\cdot$. The least element of P (and D) is the constant 0-valued function $\mathbf{0}(p)$. The structure $(D,+,\circ,,id)$ is an idempotent semiring. In fact, in its left argument $\circ$ even preserves arbitrary existing suprema and infima, as the following calculation and a dual one for infima show:

$$((\sqcup F)\circ g)(x) \;=\; (\sqcup F)(g(x)) \;=\; \sqcup F(g(x)) \;=\; \sqcup((F\circ g)(x)).$$

The modal operator $|\_\rangle$ provides a left semiring homomorphism from K into D.

If $\mathsf{test}(K)$ is a complete Boolean algebra then P is a complete lattice with D as a complete sublattice. Hence we can extend P and D by a star operation via a least fixpoint definition:

$$f^* \stackrel{\text{def}}{=} \mu g\,.\, id + f\circ g,$$

where μ is the least-fixpoint operator. It turns out that by this definition D does satisfy the Kleene algebra axioms except the second star induction law ($*$-4). Only the subalgebra of universally disjunctive predicate transformers is a full Kleene algebra.

We will now show that the algebra of modal operators that arises from a pointwise lifting is a lattice-ordered monoid that contains an idempotent semiring or a variant of a KA as a retract. This abstraction allows a more succinct pointfree style of reasoning.

A *lattice-ordered monoid* is a structure $(K,+,\sqcap,\cdot,1)$ such that $(K,+,\sqcap)$ is a lattice, $(K,\cdot,1)$ is a monoid and multiplication is additive in both arguments. These structures have

extensively been studied in [4]. If the lattice reduct of the monoid is distributive (Boolean), we call the respective structure a *d-monoid* (a *b-monoid*).

Let $\langle K\rangle$ be the sets of all mappings $\lambda x.\langle a\rangle x$ with $a \in K$ on some domain or codomain semiring K. We define addition (or join), meet and multiplication on $\langle K\rangle$ pointwise by

$$\begin{aligned}(\langle a\rangle + \langle b\rangle)p &= \langle a\rangle p + \langle b\rangle p,\\ (\langle a\rangle \sqcap \langle b\rangle)p &= (\langle a\rangle p)(\langle b\rangle p),\\ (\langle a\rangle\langle b\rangle)p &= \langle a\rangle\langle b\rangle p.\end{aligned}$$

Then the structures $(\langle K\rangle, +, \sqcap, , \langle 0\rangle, \langle 1\rangle)$ are d-monoids. Dually, with addition, meet and multiplication defined by

$$\begin{aligned}([a] + [b])(p) &= ([a]p)([b]p),\\ ([a] \sqcap [b])(p) &= [a]p + [b]p,\\ ([a][b])(p) &= [a][b]p,\end{aligned}$$

the structures $([K], +, \sqcap, , [0], [1])$ are d-monoids. In both cases the pointwise ordering coincides with the natural semiring ordering which also is the lattice ordering. Using both mappings $\lambda x.[a]x$ and $\lambda x.\langle a\rangle x$, we can extend the d-monoids to b-monoids, defining

$$(\neg\langle a\rangle)(p) = [a]\neg p, \qquad (\neg[a])(p) = \langle a\rangle\neg p.$$

We will also use the pointwise liftings of $-$ and $\rightarrow$ to the operator level.

Many properties of modal operators can now be presented much more succinctly in the respective algebra of operators. First, the test-level Galois connections can be lifted to operators $f, g : \mathsf{test}(K) \rightarrow \mathsf{test}(K)$:

$$|a\rangle f \leq g \Leftrightarrow f \leq [a|g, \qquad \langle a|f \leq g \Leftrightarrow f \leq |a]g, \tag{17}$$

for all $a \in K$. From this we get the cancellation and shunting laws

$$|a\rangle[a| \leq \langle 1\rangle \leq [a||a\rangle, \qquad \langle a||a] \leq \langle 1\rangle \leq |a]\langle a|, \tag{18}$$

$$|a\rangle f \leq g \Leftrightarrow f \leq [a|g, \qquad \langle a|f \leq g \Leftrightarrow f \leq |a]g, \tag{19}$$

$$f|a] \leq g \Leftrightarrow f \leq g\langle a| \qquad f[a| \leq g \Leftrightarrow f \leq g|a\rangle. \tag{20}$$

Semiring expressions inside of operators can be decomposed by the laws

$$\begin{aligned}\langle a + b\rangle = \langle a\rangle + \langle b\rangle, \qquad |ab\rangle = |a\rangle|b\rangle, \qquad \langle ab| = \langle b|\langle a|,\\ [a + b] = [a] \sqcap [b], \qquad |ab] = |a]|b], \qquad [ab| = [b|[a|.\end{aligned}$$

Note that the decomposition with respect to multiplication is covariant for forward modalities and contravariant for backward modalities. This results from the symmetry between domain and codomain via opposition. The decomposition can be used to transform expressions into normal form and to reason entirely at the level of modal algebra in the sense of [20].

Diamonds are isotone, i.e., $a \leq b$ implies $\langle a\rangle \leq \langle b\rangle$. Dually, boxes are antitone, i.e., $a \leq b$ implies $[b] \leq [a]$.

In the case of an MKA, the algebras of operators can be extended to KAs because of the following unfold and induction laws at the operator level (cf. [12]).

$$|1\rangle + |a\rangle|a^*\rangle \leq |a^*\rangle, \qquad |1\rangle + |a^*\rangle|a\rangle \leq |a^*\rangle, \tag{21}$$

$$f + |a\rangle g \leq g \Rightarrow |a^*\rangle f \leq g. \tag{22}$$

Setting $f = g = \langle 1 \rangle$ we obtain, from the analogue of this for the backward diamond,

$$\langle a| \leq \langle 1 \rangle \;\Rightarrow\; \langle a^*| \leq \langle 1 \rangle. \tag{23}$$

These laws for the "inner star" induce an "outer star" $|a\rangle^*$ that coincides with $|a^*\rangle$ and turns the algebra of boxes into a left KA. Analogous laws hold for the backward modal operators. They imply the star fixpoint laws

$$|a^*\rangle = |1\rangle + |a\rangle|a^*\rangle, \qquad |a^*] = |1] \sqcap |a]|a^*]. \tag{24}$$

Let us now give a point-free soundness proof for the while-rule of the propositional Hoare calculus. The proof obligation (16) translates into

$$\langle pa| \leq \langle 1 \rangle \;\Rightarrow\; \langle (pa)^* \neg p| \leq \langle \neg p \rangle,$$

which follows immediately from (23), isotonicity, locality and neutrality of $\langle 1 \rangle$.

We conclude by giving lifted versions of the semi-commutation properties (12). The first of these becomes, for the forward diamond,

$$|b\rangle f \leq f|c\rangle \;\Rightarrow\; |b^*\rangle f \leq f|c^*\rangle; \tag{25}$$

it is easily shown using (22). The second one lifts only for the case where f is a diamond:

$$\langle a||b\rangle \leq |c\rangle\langle a| \;\Rightarrow\; \langle a||b^*\rangle \leq |c^*\rangle\langle a|. \tag{26}$$

This is established by shunting the two occurrences of $\langle a|$ in the conclusion of this implication to the respective other side of the inequation using (19) and (20) and then again using (22).

8 Beside PDL: Temporal Logic

While propositional dynamic logic contains explicit statements for actions or programs and therefore allows one to compare different programs, temporal logics reason about runs of one particular program at a time. This is particularly interesting for the analysis of concurrent programs and reactive systems. Originally, temporal logics used Prior's future tense modality G with the reading "at all future states including the present one", F with the reading "at some future state including the present one" and X with the reading "at the next state". Later, the binary operator U was added with the reading $p\,\mathsf{U}q$ as "p until q", i.e., "q will eventually be true and from the present state on p will be true until q is". This system is also known as propositional linear temporal logic.

It is well known that these temporal operators can be defined in PDL, whence also in MKA: for abstract program a,

$$\mathsf{X} = |a\rangle, \tag{27}$$
$$\mathsf{F} = |a^*\rangle, \tag{28}$$
$$\mathsf{G} = |a^*], \tag{29}$$
$$p\,\mathsf{U} = |(pa)^*\rangle. \tag{30}$$

Note that X, F and G can also be defined in Kleene modules, whereas U requires a product of a test and an action which cannot be expressed in the module setting. It is obvious that, interpreted over traces, these operators have the desired semantics. It follows immediately that $\mathsf{F} = 1\mathsf{U}$ that $\mathsf{G} = \neg\mathsf{F}\neg$ and that — by the unfold laws for the Kleene star — the following unfold laws for eventually and until hold:

$$\mathsf{F} = |1\rangle + \mathsf{X}\,\mathsf{F}, \tag{31}$$
$$p\,\mathsf{U} = |1\rangle + (|p\rangle \sqcap \mathsf{X}\,(p\,\mathsf{U})). \tag{32}$$

Manna and Pnueli [31] have given an axiom system for linear temporal logic (LTL). More recently, von Karger [50] has derived these axioms as theorems in a much leaner formalism called *temporal algebra* that defines modal operators by Galois connections similar to ours over a complete Boolean algebra and uses the Theorem of Knaster and Tarski to model iteration via fixed-points on this algebra. The reconstruction by von Karger also provides a nice modular presentation of the LTL axioms. Some of them are general laws of modal logic. They therefore hold a fortiori in MKA. Some further axioms, like the above until law, are fixpoint properties and hence hold not only for von Karger's calculus, but also for the more general case of MKA. In particular, all the laws that do not involve U even hold in Kleene modules. A particular instance of such a law is

$$|a^*](p \to |a]p) \leq |a^*](p \to |a^*]p), \tag{33}$$

which can be obtained by dualising the induction law (15).

There is, however, a series of LTL axioms that depends on the particular structure of models and the way that temporal formulas are interpreted over runs of a program. Also here, we can immediately generalise von Karger's reconstruction to MKA. Von Karger shows, for instance, that some further LTL axioms are *implied* in models that satisfy a confluence property. We have seen how this can be expressed in MKA. Some further axioms hare implied in models in which every state has precisely one successor state. This can be modeled using the well-known properties of being a partial function or *simple* (or *deterministic*) and being total or *entire* (cf. [18]), expressed in MKA as

$$\langle a||a\rangle \leq \langle 1\rangle, \qquad\qquad \langle 1\rangle \leq |a\rangle\langle a|.$$

The element a is a *map* if it is simple and entire. For maps, in particular, $|a\rangle = |a]$, which is a direct translation of one of the LTL axioms, and $|a]1 = 1$. A co-simplicity property is also imposed on backward modalities, whereas this is not the case for entirety. Just in contrast, the model of linear temporal logic is assumed to be a discrete linear ordering with a left but with no right endpoint. It remains to model the initial state.

Intuitively, a test p characterises initial states if it is contained in the complement of the codomain of a, whence $p \leq \neg\langle a|1 = [a|0$. Dually, p characterises terminal states if it is contained in the complement of the domain of a, whence $p \leq \neg|a\rangle 1 = |a]0$ (see also Section 3.1). Terminality, however, is of no further interest here. Let now init be the greatest such element, i.e.,

$$\mathsf{init}_a \stackrel{\text{def}}{=} [a|0.$$

This initiality condition is important for modeling validity of a modal implication $p \to q$ as $\mathsf{init}_a \cdot p \leq q$.

The following generalisation of von Karger's completeness result for propositional linear temporal logic can then easily be generalised to modal Kleene algebra.

Theorem 8.1. *Modal Kleene algebra is complete for the Manna and Pnueli axioms for propositional linear temporal logic. The additional conditions for linear models and validity of modal implications can be expressed in modal Kleene algebra (as additional axioms).*

Von Karger also sketches a completeness result for computational tree logic. We conjecture that also this result can be generalised to MKA.

9 Termination Analysis

9.1 Termination in Modal Kleene Algebra

We now deal with the question whether a transition system admits infinite transition paths. To this end we abstract a notion of termination for modal semirings from set-theoretic

relations. A similar characterisation has been used, for instance, in [20] for related structures. A set-theoretic relation $R \subseteq A \times A$ on a set A is well-founded if there are no infinitely descending R-chains, that is, no infinite chains $x_0, x_1, \ldots$ such that $(x_{i+1}, x_i) \in R$. It is Noetherian if there are no infinitely ascending R-chains, i.e., no infinite chains $x_0, x_1, \ldots$ such that $(x_i, x_{i+1}) \in R$. Thus R is *not* well-founded if there is a non-empty set $P \subseteq A$ (denoting the infinite chain) such that for all $x \in P$ there exists some $y \in P$ with $(y, x) \in R$. Equivalently, therefore, P is contained in the image of P under R, i.e., $P \subseteq (PR)^\urcorner$. Consequently, if R *is* well-founded, then only the empty set may satisfy this condition.

Abstracting to a modal semiring K we say that a is *well-founded* if

$$p \leq \langle a|p \Rightarrow p \leq 0 \tag{34}$$

for all $p \in \mathsf{test}(K)$. Dually, a is *Noetherian* if for all $p \in \mathsf{test}(K)$,

$$p \leq |a\rangle p \Rightarrow p \leq 0. \tag{35}$$

Note that by de Morgan duality a is Noetherian iff, for all $p \in \mathsf{test}(K)$,

$$|a]p \leq p \Rightarrow 1 \leq p. \tag{36}$$

Let us look at these definitions from another angle. According to the standard definition, a relation R on a set A is well-founded iff every non-empty subset of A has an R-minimal element. In a $^\urcorner$-semiring K the minimal part of $p \in \mathsf{test}(K)$ w.r.t. some $a \in K$ can algebraically be characterised as $p - \langle a|p$, i.e., as the set of points that have no a-predecessor in p. So, by contraposition, the well-foundedness condition holds iff for all $p \in \mathsf{test}(K)$

$$p - \langle a|p \;\leq\; 0 \;\Rightarrow\; p \leq 0,$$

which by simple Boolean algebra can be transformed into (34).

It is easy to prove some of the well-known properties of well-founded and Noetherian relations in modal Kleene algebra [12]. First, 0 is the only Noetherian test. Second, the property of being Noetherian is downward closed. Third, every Noetherian element is irreflexive and non-dense, provided it is non-trivial. Fourth, an element is Noetherian iff its transitive closure is, but no reflexive transitive closure is Noetherian. Finally, Noethericity of a sum implies Noethericity of its components, whereas the converse direction does not hold in general. We will later present commutativity conditions that enforce this converse implication.

9.2 Termination via Löb's Formula

We now investigate two alternative equational characterisations of Noethericity. The first one uses the star. The second one is without the star. It holds for the special case of a *transitive* Kleenean element a, i.e., when $aa \leq a$.

Let K be a $^\urcorner$-semiring. Consider the equations

$$|a\rangle \leq |a\rangle^+(|1\rangle - |a\rangle), \tag{37}$$

$$|a\rangle \leq |a\rangle(|1\rangle - |a\rangle). \tag{38}$$

The equation (38) is a translation of Löb's formula from modal logic (cf. [7]) which expresses well-foundedness in Kripke structures. We say that a is *pre-Löbian* if it satisfies (37). We say that a is *Löbian* if it satisfies (38).

In the relational model, Löb's formula states that a is transitive and that there are no infinite a-chains. We will now relate Löb's formula and Noethericity.

Theorem 9.1. *The following statements hold in a modal Kleene algebra.*
(i) Every Löbian and every pre-Löbian element is Noetherian.
(ii) Every Noetherian element is pre-Löbian.
(iii) Every Noetherian element is Löbian if it is transitive.

Properties (i) and (iii) already hold in $\ulcorner$-semirings. A calculational proof of (ii) based on [20] can be found in [12]. A closer analysis of the proof shows that in (iii) it suffices to assume that a is *weakly transitive*, i.e,

$$|aa\rangle \leq |a\rangle.$$

Weak transitivity is a much weaker requirement than transitivity $aa \leq a$. To see this, view the Kleene elements again as sets of computation paths. If a consists of paths with exactly two states each (i.e., is isomorphic to a binary relation on states) then aa consists of paths with exactly three states, and so $aa \leq a$ holds only if $aa = 0$. But a is still weakly transitive if it is transitive considered as a binary relation.

The calculational translation between the Löb-formula and our definition of Noethericity is quite interesting for the correspondence theory of modal logic (see also Section 10). In this view, our property of Noethericity expresses a frame property, which is part of semantics, whereas the Löb formula stands for a modal formula, which is part of syntax. In modal semirings, we are able to express syntax and semantics in one and the same formalism. Moreover, while the traditional proof of the correspondence uses model-theoretic semantic arguments based on infinite chains, the algebraic proof is entirely calculational and avoids infinity. This is quite beneficial for instance for mechanisation.

9.3 Termination via Infinite Iteration

Cohen has extended KA with an $^\omega$ operator for modeling infinite iteration [8]; he has also shown applications in concurrency control. In [48], this algebra has been used for calculating proofs of theorems from abstract rewriting that use simple termination assumptions.

Dually to the Kleene star, the omega operator is defined as a greatest post-fixpoint. An *ω-algebra* is a structure (K, ω) where K is a KA and

$$a^\omega \leq aa^\omega, \tag{39}$$
$$c \leq ac + b \Rightarrow c \leq a^\omega + a^*b, \tag{40}$$

for all $a, b, c \in K$. Consequently, a^ω is also the greatest fixpoint of $\lambda x \,.\, ax$.

Like in Section 7, for an MKA K it seems interesting to lift (39) and (40) to operator algebras, similar to the laws (21), and (22) for the star. This is very simple for (39): for $a \in K$,

$$|a^\omega\rangle \leq |a\rangle|a^\omega\rangle. \tag{41}$$

However, as we will see below, there is no law corresponding to (22) and (40). The proof of (22) uses (llp) and works, since the star occurs at the left-hand sides of inequalities. There is no similar law that allows us to handle the omega, which occurs at right-hand sides of inequalities. But instead, one can axiomatise the greatest fixpoint $\nu|a\rangle$ of $|a\rangle$ for $a \in K$ by

$$\nu|a\rangle \leq |a\rangle\, \nu|a\rangle, \tag{42}$$
$$p \leq |a\rangle p + q \Rightarrow p \leq \nu|a\rangle + |a^*\rangle q. \tag{43}$$

If $\mathsf{test}(K)$ is complete then by the Knaster-Tarski theorem $\nu|a\rangle$ always exists, since $|a\rangle$ is isotone. In that case one can use a weaker axiomatisation (see [20]) from which (43) follows by greatest fixpoint fusion.

It will turn out that $\nu|a\rangle$ is more suitable for termination analysis than a^ω. Since $|a\rangle p = \neg|a\rangle\neg p$, existence of $\nu|a\rangle$ also implies existence of the least fixpoint $\mu|a]$ of $|a]$, since $\mu|a] = \neg\nu|a\rangle$. In the modal μ-calculus, $\mu|a]$ is known as the *halting predicate* (see, e.g., [21]). With the help of $\nu|a\rangle$ we can rephrase Noethericity more concisely as

$$\nu|a\rangle = 0. \tag{44}$$

As an immediate consequence of this we obtain

Corollary 9.2. *Define, for fixed $q \in \mathsf{test}(K)$ and $a \in K$, the function $f : \mathsf{test}(K) \to \mathsf{test}(K)$ by $f(p) = q + |a\rangle p$. If $\nu|a\rangle$ exists and $a \in \mathcal{N}(K)$ then f has the unique fixpoint $|a^*\rangle q$.*

A notion of guaranteed termination can easily be defined in ω-algebra as the absence of infinite iteration. We call a *ω-Noetherian* if $a^\omega \leq 0$.

We now study the relation between Noethericity and ω-Noethericity. Recall that a $\ulcorner$-KA K is extensional if

$$|a\rangle \leq |b\rangle \Rightarrow a \leq b$$

holds for all $a, b \in K$. Note that the language model is not extensional. The following lemma shows that the relation between Noethericity and ω-Noethericity does not depend on extensionality. This is somewhat surprising, since set-theoretic relations are extensional and in the relational model the two notions coincide.

Lemma 9.3. *Consider a modal Kleene algebra K that is also an ω-algebra.*

(i) Every Noetherian element is ω-Noetherian, but not conversely, not even for extensional MKA.

(ii) There is a non-extensional K which is Noetherian and ω-Noetherian.

For the proof see [11]. By the following corollary, (40) cannot in general be lifted to (43).

Corollary 9.4. *There exists an* MKA *K such that $\nu|a\rangle \leq 0$, but $a^\omega > 0$ for some $a \in K$.*

Thus ω-algebra does not entirely capture the standard notion of termination.

We now study the behaviour of the exhaustive finite iteration of an element $a \in K$, given by

$$\mathsf{exh}\, a \stackrel{\text{def}}{=} \mathsf{while}\, \ulcorner a\, \mathsf{do}\, a \;=\; a^*\neg\ulcorner a.$$

Then we can represent the set of points from which a terminal point can be reached via a-steps as

$$\ulcorner(\mathsf{exh}\, a) \;=\; \ulcorner(a^*\neg\ulcorner a) \;=\; |a^*\rangle\neg\ulcorner a. \tag{45}$$

Proposition 9.5. *If a is Noetherian then $\ulcorner(\mathsf{exh}\, a) = 1$, i.e., from every starting point a terminal point can be reached.*

For the proof see again [11]. This shows again that modal Kleene algebra is more adequate for termination analysis than omega algebra. To see this, consider the algebra LAN of formal languages which is both an omega algebra and an MKA with complete test algebra $\mathsf{test}(\mathrm{LAN}) = \{0, 1\}$. In LAN we have $|a\rangle 1 = \ulcorner a = 1 \neq 0$ when $a \neq 0$ and hence a is Noetherian iff $a = 0$. Moreover, distinguishing the cases $a = 0$ and $a \neq 0$, easy calculations show that in LAN we have $\mathsf{exh}\, a = \neg\ulcorner a$. This mirrors the fact that by totality of concatenation a nonempty language can be iterated indefinitely without reaching a terminal element. But we also have $a^\omega = 0$ whenever $1 \sqcap a = 0$. Therefore, unlike in the relational model, $a^\omega = 0 \not\Rightarrow \ulcorner(\mathsf{exh}\, a) = 1$, while still $\nu|a\rangle = 0 \Rightarrow \ulcorner(\mathsf{exh}\, a) = 1$. Hence, for termination analysis in KAs more general than the relational model the element $\nu|a\rangle$ seems more adequate than a^ω.

9.4 Additivity of Termination

It has been shown that many statements of abstract rewriting that depend on termination assumptions can be proved in ω-algebra [48], among them an abstract variant of Bachmair's and Dershowitz's well-founded union theorem [2], but also many of the so-called cooperation theorems. It seems that Kleene algebra and ω-algebra capture the regular fragment of abstract rewriting. However, many other properties of abstract rewriting require context-free reasoning. We will show in this and the following section that modal Kleene algebra provides ways of reasoning also in this larger fragment. Moreover, as we have seen in the previous section, there is a gap between termination in ω-algebra and in $^{\ulcorner}$-KA. Here, we provide a proof of Bachmair's and Dershowitz's theorem in $^{\ulcorner}$-KA.

Consider a KA K and $a, b \in K$. We say that a *semi-commutes* over b if $ba \leq a^+b^*$. a *quasi-commutes* over b if $ba \leq a(a+b)^*$. Semi-commutation and quasi-commutation state conditions for permuting certain steps to the left of others. In general, sequences with a-steps and b-steps can be split into a "good" part with all a-steps occurring to the left of b-steps and into a "bad" part where both kinds of steps are mixed. Semi-commutation implies quasi-commutation; in extensional KAs the reverse implications holds as well (see [48] for proofs).

One of the main results in this area is the Bachmair-Dershowitz well-founded union theorem; it generalizes in the following way from relations to modal Kleene algebra.

Theorem 9.6. *Let K be an extensional modal Kleene algebra with complete test algebra. For all $a, b \in K$, let a quasi-commute over b. Then a and b are Noetherian iff their sum is Noetherian.*

The proof in modal Kleene algebra takes about one page of algebraic calculation, see [11]. This shows that modal Kleene algebra provides proofs for abstract rewriting that are as simple as those in omega algebra. Note that the proofs in [2] are rather informal, while also previous diagrammatic proofs (e.g. [19]) suppress many elementary steps. In contrast, the algebraic proofs are complete, formal and still simple. An extensive discussion of the relation between the proofs in omega algebra and their diagrammatic counterparts can be found in [48]. In particular, the algebraic proofs mirror precisely the diagrammatic ones. This also holds for the modal proofs we present here.

9.5 Newman's Lemma

We now turn from semi-commutation to commutation and confluence. For their direct algebraic characterisation one either has to use converse at the element level or a combination of forward and backward modalities at the operator level. Since we do not have converse available, we have to choose the second alternative.

We say that $a, b \in K$ *commute* if $\langle b^* || a^* \rangle \leq |a^*\rangle\langle b^*|$, and *locally commute* if $\langle b || a \rangle \leq |a^*\rangle\langle b^*|$. The more standard notions of confluence and local confluence are recovered setting $a = b$. Newman's Lemma, originally stated for a single rewrite relation, says that a locally confluent and Noetherian rewrite relation is even confluent. It has been generalised to two relations in [45] for a theory of term-rewriting with pre-congruences that extends the traditional equational case. The generalisation of the equational Church-Rosser theorem is similar. While the Church-Rosser case has already been proved in Kleene algebra in [47], it has been argued in [48] that a proof of Newman's lemma does not work in pure Kleene or omega algebra, since these structures capture only the regular fragment of abstract rewriting while the standard proof of Newman's lemma requires context-free recursion in the centre of a formula with left and right contexts.

In contrast to previous approaches [15,42], modal Kleene algebra allows a calculational proof that mirrors precisely the previous diagrammatic one given in [45].

Theorem 9.7. *Let K be a modal Kleene algebra with complete test algebra. If $a+b$ is Noetherian and a and b locally commute then a and b commute.*

Proof. (Sketch) The central idea of our proof is to use a generalised predicate (rc stands for "restricted commutation")

$$rc(p,a,b) \Leftrightarrow \langle b^*|\langle p\rangle|a^*\rangle \leq |a^*\rangle\langle b^*|.$$

$rc(p,a,b)$ states that a and b commute on all points in p. We have used the notation $\langle p\rangle$ to enhance the symmetry of the formulation; this is justified, since $|p\rangle = \langle p|$ for all tests p. Clearly, a and b commute iff $rc(1,a,b)$, so that commutation can be retrieved as a special case. Then the predicate

$$r = \mathsf{sup}\,\{p \mid rc(p,a,b)\}$$

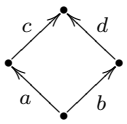
characterizes the set of all points on which a and b commute; it is contracted by $|a+b]$, so that, by the second form (36) of Noethericity, we are done. Again, the calculations take less than a page. For full details see [11]. □

Exhaustive iteration and simplicity, as previously defined, can also be used to show uniquenesss of term normal forms for confluent actions. In the proof, the points in $(\mathsf{exh}\,a)^\neg$ represent term normal forms whereas uniqueness is expressed by simplicity, as introduced in the context of temporal logics. A proof can be found in [11].

10 Modal Kleene Algebra and Correspondence Theory

An important subfield of modal logic is correspondence theory [7,40] that studies translations between relational and modal characterisations of certain properties of the underlying Kripke frames. Usually, the correctness proofs for these translations are done at the semantic level, frequently using pointwise arguments. In this section we will give some examples how such translations can be done purely algebraically using the calculus of MKA.

We start with the commutation formulas used in the previous section and study diagrams of the type



In abstract relational algebra this is expressed as $a^\smile b \leq cd^\smile$. Recalling the notion of extensionality from Section 5.3, we first observe that, in an extensional MKA with converse, by locality and (9) one has

$$a^\smile b \leq cd^\smile \;\Leftrightarrow\; \langle a||b\rangle \leq |c\rangle\langle d|. \tag{46}$$

Although being extensional is not very common for MKAs, the formula $\langle a||b\rangle \leq |c\rangle\langle d|$ still is an adequate generalisation of the relational one; it expresses that any two transition paths along a and b that emanate from a common starting point can be joined extending them by c and d transition paths, respectively.

In many modal logics only forward or only backward modalities are available. So it is interesting which type of formulas can be expressed using only one sort of modalities. For the above commutation property this is possible, resulting in the *Geach formula* [7,40]:

Lemma 10.1 *In general* MKA*s*

$$\langle a||b\rangle \leq |c\rangle\langle d| \;\Leftrightarrow\; |b\rangle|d] \leq |a]|c\rangle.$$

Hence, in extensional MKA*s with converse*

$$a^\smile b \leq cd^\smile \;\Leftrightarrow\; |b\rangle|d] \leq |a]|c\rangle.$$

Proof. Starting from the right-hand side of (46) this is shown very concisely at the operator level using the shunting rules:

$$\langle a||b\rangle \leq |c\rangle\langle d| \;\Leftrightarrow\; |b\rangle \leq |a]|c\rangle\langle d| \;\Leftrightarrow\; |b\rangle|d] \leq |a]|c\rangle.$$

The first step uses (17), the second one (20). □

Consequently, commutation and local commutation are equivalent to the following formulas:

$$|a^*\rangle|b^*] \leq |b^*]|a^*\rangle, \qquad |a\rangle|b^*] \leq |b]|a^*\rangle.$$

However, these are much less intuitive than our original ones. But the proof we sketched above can be carried out in this unidirectional form as well.

Special cases of commutation type properties are determinacy $\langle a||a\rangle \leq \langle 1\rangle$ and totality $\langle 1\rangle \leq |a\rangle\langle a|$ (which is easily shown to be equivalent to $\ulcorner a = 1$); these were already used in Section 8.

For our last example we return to the Löb formula mentioned in Section 9.2. In its original version it takes the form (cf. [6,7])

$$\Box(\Box p \to p) \to \Box p.$$

Defining validity of a formula $P(p)$ about tests p as $\forall\, p\,.\, 1 \leq P(p)$ we first observe that, by Boolean shunting, $Q(p) \to R(p)$ is valid iff $\forall\, p\,.\, Q(p) \leq R(p)$. Hence we can transcribe the validity assertion for the Löb formula over frame a as

$$\forall\, p\,.\, |a](|a]p \to p) \leq |a]p.$$

Lemma 10.2

$$\forall\, p\,.\, |a](|a]p \to p) \leq |a]p \;\;\Leftrightarrow\; \forall\, p\,.\, |a\rangle p \leq |a\rangle(p - |a\rangle p).$$

Proof. We use that $\neg$ is a bijection on tests, so that for any predicate P we have $\forall\, p\,.\, P(p) \Leftrightarrow \forall\, p\,.\, P(\neg p)$. We calculate

$$\begin{array}{ll} & |a](|a]p \to p) \leq |a]p \\ \Leftrightarrow & \quad \{\![\text{ negation }]\!\} \\ & \neg|a]p \leq \neg|a](|a]p \to p) \\ \Leftrightarrow & \quad \{\![\text{ connection between } |a] \text{ and } |a\rangle,\ \neg(q \to r) = q - r \ ]\!\} \\ & |a\rangle\neg p \leq |a\rangle(\neg|a\rangle\neg p - p) \\ \Leftrightarrow & \quad \{\![\text{ definition of subtraction }]\!\} \\ & |a\rangle\neg p \leq |a\rangle(\neg p - |a\rangle\neg p). \end{array}$$

□

Using now the definition of the pointwise ordering on modal operators we can compact the second of these formulas into the one given in Section 9.2. Similarly, one can show the equivalence of (15) and (33).

Although the technique we have shown for translating modal validity is, of course, generally applicable, we refrain from treating further examples in this survey.

11 Greedy-Like Algorithms

11.1 Looping for Optimality

The final application of this survey concerns algorithm derivation. It ties in well with generalised confluence and exhaustive iteration.

Greedy algorithms solve certain optimisation problems, proceeding in a stepwise fashion without backtracking. At each step there is a set of choices from which a greedy algorithm always takes the one that seems best at the moment, i.e., it works locally without backtracking and lookahead to the global optimum that is to be found eventually. Instances of this scheme are shortest path and minimum spanning tree problems in graphs, the construction of Huffman codes and scheduling problems. Of course, the greedy approach only works for certain types of problems: as is well-known from hiking in the mountains, always choosing the steepest path will rarely lead to the highest summit of the whole area. The central correctness requirement for the greedy scheme is that *a local choice must not impair reaching the global optimum*.

We now use modal Kleene algebra for deriving general conditions under which a loop satisfies this principle. It turns out that local optimality is inessential; so we study a more general class of loops that we call *greedy-like*. In [34] a relational derivation was abstracted to modal Kleene algebra via the Geach formula (cf. Lemma 10.1), whence avoiding backward modalities. While this corresponds to the standard approach that a modal logician would take, modal Kleene algebra offers the additional flexibility of simple combined reasoning with forward and backward modalities via Galois connections. Then the development of greediness conditions can be based again on commutation properties that, like in the case of abstract rewriting, immediately reflect the choices that are taken at each step of a run of a greedy-like algorithm. Here, we briefly describe this commutation-based development.

We start with a specification element t that abstracts a relation between inputs and admissible outputs and an element c that abstracts a comparison relation on outputs capturing the notion of (global) optimality. The derivation will exhibit the precise requirements on c.

An element r *improves* t with respect to c if it always relates inputs to outputs that are at least as good as those prescribed by t. If r and t are relations this reads formally $t^\smile r \leq c$, which in MKA immediately translates into the predicate

$$imp(r,t,c) \stackrel{\text{def}}{\Leftrightarrow} \langle t||r\rangle \leq |c\rangle.$$

Since then 0 trivially improves t, we are interested in the greatest improvement. In REL this always exists and is given by the residual $t^\smile \backslash c$. However, since we want to avoid residuals, we will not make use of this representation.

An implementation of specification t that always produces optimal solutions then is a relation that refines and improves t. So we define

$$opt(r,t,c) \stackrel{\text{def}}{\Leftrightarrow} r \leq t \wedge imp(r,t,c)$$

and want to calculate a sufficient criterion under which a loop program $w \stackrel{\text{def}}{=} \mathsf{while}\ p\,\mathsf{do}$ with loop condition $p \in \mathsf{test}(K)$ and body $s \in K$ satisfies $opt(w,t,c)$, i.e.,

$$w \leq t, \qquad (47) \qquad\qquad imp(w,t,c), \qquad (48)$$

where we defer the treatment of (47) to the next section.

Spelling out the definitions in (48) results in $\langle t||(ps)^*\neg p\rangle \leq |c\rangle$. We abstract a bit and try to answer the question when, for $q \in \mathsf{test}(K)$ and $a \in K$, we have $\langle t||a^*q\rangle \leq c$. By the lifted semi-commutation property (26) in Section 7 this can be established if

$$\langle t||a\rangle \leq |c\rangle\langle t|, \qquad (49) \qquad\qquad \langle t|\langle q\rangle \leq |c\rangle, \qquad (50)$$

since then by locality

$$\langle t||a^*q\rangle \leq |c^*\rangle\langle t|\langle q\rangle \leq |c^*\rangle|c\rangle \;=\; |c^+\rangle.$$

If we now assume c to be weakly transitive (cf. Section 9.2), which is reasonable for a comparison relation, we have $|c^+\rangle \leq |c\rangle$ and can draw the desired conclusion.

How can we, in turn, establish (49) and (50), at least in our special case? Translating back we get the proof obligations

$$\langle t||ps\rangle \leq |c\rangle\langle t|, \qquad (51) \qquad\qquad \langle t|\langle\neg p\rangle \leq |c\rangle. \qquad (52)$$

Condition (51) means that every pass through the loop body s preserves the possibility of obtaining a solution that is at least as good as all possible solutions before; (52) means that upon loop termination no possible solution is better than the termination value.

11.2 Iterating Through the Problem Domain

We now decompose the specification relation t into the exhaustive iteration of an element e of a set of elementary steps between points in the problem domain. We admit, as initial approximations, arbitrary inputs, but as outputs only terminal elements from which no further elementary steps are possible. Therefore we assume now that t has the special shape (cf. Section 9.3)

$$t \;=\; \mathsf{exh}\, e \;=\; e^* \,;\, \neg\lceil e \;=\; \mathsf{while}\ \lceil e\ \mathsf{do}\ e. \qquad (53)$$

Such a problem structure is found, e.g., in matroids and greedoids [22,26] where it is additionally assumed that t is a discrete strict-order and that all terminal (or maximal) elements, the *bases*, have the same height (also known as *rank* or *dimension*) in the associated Hasse diagram.

We try to calculate an implementation that traverses the problem domain without backtracking, i.e., using elementary steps only forward. This suggests trying $ps \leq e$. Now, by isotonicity of the star operation, proof obligation (47) can be fulfilled if additionally we can achieve $\neg p \leq \neg\lceil e$ or, equivalently, $\lceil e \leq p$. Sufficient conditions for these properties are

$$ps \leq e \qquad\qquad \lceil(ps) \geq \lceil e. \qquad (54)$$

These are reasonable requirements, since they prevent that the iteration blocks at a non-terminal element. They even imply $\lceil(ps) = \lceil e$.

Next, we deal with proof obligation (52), assuming (53). We calculate

$$\begin{aligned}
\langle t|\langle\neg\lceil e\rangle \leq |c\rangle \;&\Leftrightarrow\; \langle\neg\lceil e\rangle|t\rangle \leq \langle c| \\
&\Leftrightarrow\; \langle\neg\lceil e\rangle|e^*\rangle\langle\neg\lceil e\rangle \leq \langle c| \\
&\Leftrightarrow\; \langle\neg\lceil e\rangle \,;\, (\langle 1\rangle + |e\rangle|e^*\rangle)\langle\neg\lceil e\rangle \leq \langle c| \\
&\Leftrightarrow\; \langle\neg\lceil e\rangle \leq |c\rangle.
\end{aligned}$$

Step one employs properties of converse. Step two uses (53). Step three unfolds the star. Step four uses distributivity, locality, $\neg\lceil e\, e = 0$, idempotence of $\neg\lceil e$ and equality of backward and forward diamonds of a test.

So (52) is established if c is *weakly reflexive* on terminal elements, i.e., if

$$\langle\neg\lceil e\rangle \leq |c\rangle.$$

This holds, in particular, if c is fully reflexive, i.e., a pre-order. But in some applications one may choose to leave c partially reflexive. E.g., when constructing a Huffman code, the non-terminal elements are proper forests, for which a comparison relation is not given as easily as for the terminal elements, which are single code trees.

As for proof obligation (51), it is a generic condition that has to be considered individually in each case. Our derivation can be summed up as follows.

Theorem 11.1 *Suppose that c is weakly reflexive on $\neg\ulcorner e$ and weakly transitive and that $t = \mathsf{exh}\, e$. Then* (54) $\wedge$ (51) $\Rightarrow$ $opt(\mathsf{while}\ \ulcorner e\, \mathsf{do}\, s, t, c)$.

So far we still have a general scheme that does not specifically mention greediness. But we can refine S further to choose in every step a locally optimal element. To this end we need another pre-order l and stipulate $imp(s, e, l)$. This now provides a truly greedy algorithm, the correctness of which is already shown by Theorem 11.1. It corresponds to Curtis's "Best-Global" algorithm [10].

In [34] we show that one can fully reconstruct Curtis's classification of Greedy algorithms [10] in the abstract setting of MKA, even using forward modalities only. The reason for this is that converse enters the derivation only in the limited way of general commutation properties which can be expressed by forward modalities only, using the Geach formula of Lemma 10.1. The modal approach again leads to considerably more concise proofs than the original relational/allegorical ones.

12 Conclusion

We have outlined the calculus of modal Kleene algebra and discussed several applications, most of them in the field of semantics, system calculi and development of programs and algorithms. The proofs that are needed in these examples are abstract, concise and entirely calculational.

Together with previous work [47,48], our case study in abstract rewriting, for instance, shows that large parts of this theory can easily be reconstructed in modal Kleene algebra. This is probably a novel idea. Other practical results, for instance the soundness proof of propositional Hoare logic or the reconstruction of temporal logics, are strongly based on previous work. Here, the main contribution seems to be that modal Kleene algebra may serve as a convenient uniform framework. Sometimes, however, it even yields a drastic cut with Occam's razor: in the cases of propositional dynamic logic and linear temporal logic we can significantly reduce the number of axioms. In the case of linear temporal logics again, we can replace the three independent concepts X, G and U by one modal operator and the star.

Relational algebraists may claim that most of the results presented in this paper could as well be treated in their formalism. While this is certainly true, we believe that modal Kleene algebra nevertheless provides some advantages. It has fewer operations, it is more algorithmic, and the symmetries between the additional relational operators is captured in a more structured way by the modalities. This often leads to a more concise and readable notation. Finally, the lifting to the modal operator algebras provides an additional level of abstraction that is not present in relational algebra.

There is one particular application of modal Kleene algebra that has not been discussed in this survey. Ehm has extended our approach to a calculus for the analysis of pointer algorithms [16]. He has combined modal Kleene algebra with techniques from fuzzy set theory to model the projection onto particular substructures of a given pointer structure. The reachability analysis performed by pointer algorithms, however, works to a large extent in pure modal Kleene algebra. Giving a full account of these results is beyond the scope of this paper. We believe that Ehm's approach can be adapted to the analysis of object structures.

So far, all our proofs are by paper and pencil. However, the simplicity of these proofs makes them ideal candidates for mechanisation. Our case studies in rewriting show that much less structure is needed for formalising proofs with a proof assistant than with previous approaches (e.g. [37,43]). We expect similar results when modal Kleene algebra is integrated into a formal method. Note that a considerable part of formal reasoning with popular methods like Z [44] or B [1] is essentially relational. In particular, Kleene algebra has strong connections to automata-theoretic decision procedures.

The results of this paper contribute to an attempt to establish modal Kleene algebra as a formalism for safe cross-theory reasoning and therefore interoperability between different calculi for program and system analysis, modal or relational. We have tried to support this claim both from the syntactic and the semantic point of view. In the future, we plan extensive case studies, first of all in the area of program and protocol analysis. Due to its simplicity and flexibility, we believe that modal Kleene algebra offers a considerable potential that deserves further exploration, as well for peeling potatoes as for slicing pineapples.

Acknowledgment We are grateful to Roland Backhouse, Ernie Cohen, Sharon Curtis, Thorsten Ehm, Hitoshi Furusawa, Wolfram Kahl, Dexter Kozen, Hans Leiss, Oege de Moor, Gunther Schmidt, Michel Sintzoff and Joakim von Wright for valuable comments and discussions.

> Propose to an Englishman any principle, or any instrument, however admirable, and you will observe that the whole effort of the English mind is directed to find a difficulty, a defect or an impossibility in it. If you speak to him of a machine for peeling a potato, he will pronounce it impossible: if you peel a potato with it before his eyes, he will declare it useless, because it will not slice a pineapple.
>
> *Charles Babbage 1852*

References

1. J.-R. Abrial. *The B-Book*. Cambridge University Press, 1996.
2. L. Bachmair and N. Dershowitz. Commutation, transformation, and termination. In J. H. Siekmann, editor, *8th International Conference on Automated Deduction*, volume 230 of *LNCS*, pages 5–20. Springer, 1986.
3. R. Backhouse and D. Michaelis. Winning strategies. In R. Berghammer and B. Möller, editors, *Relational and Kleene-Algebraic Methods in Computer Science*, LNCS. Springer, 2004. (to appear).
4. G. Birkhoff. *Lattice Theory*, volume 25 of *Colloquium Publications*. American Mathematical Society, 1984. Reprint.
5. C. Brink. Boolean modules. *Journal of Algebra*, 71:291–313, 1981.
6. R. Bull and K. Segerberg. Basic modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, chapter II.1, pages 1–88. D. Reidel, 1984.
7. B. F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, 1980.
8. E. Cohen. Separation and reduction. In R. Backhouse and J. N. Oliveira, editors, *Proc. of Mathematics of Program Construction, 5th International Conference, MPC 2000*, volume 1837 of *LNCS*, pages 45–59. Springer, 2000.
9. E. Cohen, D. Kozen, and F. Smith. The complexity of Kleene algebra with tests. Technical Report 96-1598, Computer Science Department, Cornell University, July 1996.
10. S.A. Curtis. *A Relational Approach to Optimization Problems*. PhD thesis, Oxford University, 1996. Oxford University Computing Laboratory, Technical Monograph PRG-122.
11. J. Desharnais, B. Möller, and G. Struth. Termination in modal Kleene algebra. Technical Report 2004-04, Universität Augsburg, Institut für Informatik, January 2004.
12. J. Desharnais, B. Möller, and G. Struth. Kleene algebra with domain. Technical Report 2003-07, Universität Augsburg, Institut für Informatik, June 2003.
13. J. Desharnais, B. Möller, and F. Tchier. Kleene under a demonic star. In T. Rus, editor, *Algebraic Methodology and Software Technology*, volume 1816 of *LNCS*, pages 355–370. Springer, 2000.
14. H. Doornbos. A relational model of programs without the restriction to Egli-Milner-monotone constructs. In E.-R. Olderog, editor, *Programming Concepts, Methods and Calculi*, pages 363–382. North-Holland, 1994.
15. H. Doornbos, R. Backhouse, and J. van der Woude. A calculational approach to mathematical induction. *Theoretical Computer Science*, 179:103–135, 1997.
16. T. Ehm. Pointer Kleene algebra. In R. Berghammer, B. Möller, and G. Struth, editors, *Relational and Kleene-Algebraic Methods in Computer Science*, LNCS. Springer, 2004. (to appear).
17. T. Ehm, B. Möller, and G. Struth. Kleene modules. In R. Berghammer, B. Möller, and G. Struth, editors, *Relational and Kleene-Algebraic Methods in Computer Science*, LNCS. Springer, 2004. (to appear).
18. P. Freyd and A. Scedrov. *Categories, allegories*. North-Holland, 1990.

19. A. Geser. *Relative termination.* PhD thesis, Fakultät fur Mathematik und Informatik, Universität Passau, 1990.
20. R. Goldblatt. An algebraic study of well-foundedness. *Studia Logica*, 44(4):422–437, 1985.
21. D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic.* MIT Press, 2000.
22. P. Helman, B.M.E. Moret, and H.D. Shapiro. An exact characterization of greedy structures. *SIAM Journal on Discrete Mathematics*, 6:274–283, 1993.
23. M. Hollenberg. Equational axioms of test algebra. In M. Nielsen and W. Thomas, editors, *Computer Science Logic, 11th International Workshop, CSL '97*, volume 1414 of *LNCS*, pages 295–310. Springer, 1997.
24. N. Jacobson. *Basic Algebra*, volume I,II. Freeman, New York, 1985.
25. B. Jónsson and A. Tarski. Boolean algebras with operators, Part I. *American Journal of Mathematics*, 73:891–939, 1951.
26. B. Korte, L. Lovász, and R. Schrader. *Greedoids.* Springer, 1991.
27. D. Kozen. A representation theorem for $*$-free PDL. Technical Report RC7864, IBM, 1979.
28. D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994.
29. D. Kozen. Kleene algebra with tests. *Trans. Programming Languages and Systems*, 19(3):427–443, 1997.
30. Hans Leiß. Kleenean semimodules and linear languages. In Zoltán Ésik and Anna Ingólfsdóttir, editors, *FICS'02 Preliminary Proceedings*, number NS-02-2 in BRICS Notes Series, pages 51–53. Univ. of Aarhus, 2002.
31. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems — Specification.* Springer, 1991.
32. B. Möller. Derivation of graph and pointer algorithms. In B. Möller, H.A. Partsch, and S.A. Schuman, editors, *Formal program development*, volume 755 of *LNCS*, pages 123–160. Springer, 1993.
33. B. Möller. Lazy Kleene algebra. In D. Kozen, editor, *Proc. of Mathematics of Program Construction, 7th International Conference, MPC 2004*, LNCS. Springer, 2004. (to appear). Preliminary version: Report No. 2003-17, Institut für Informatik, Universität Augsburg, December 2003.
34. B. Möller and G. Struth. Greedy-like algorithms in modal Kleene algebra. In R. Berghammer, B. Möller, and G. Struth, editors, *Relational and Kleene-Algebraic Methods in Computer Science*, LNCS. Springer, 2004. (to appear).
35. B. Möller and G. Struth. Modal Kleene algebra and partial correctness. In *Proc. 10th International Conference on Algebraic Methodology and Software Technology AMAST '2004*, LNCS. Springer, 2004. (to appear). Preliminary version: Report No. 2003-08, Institut für Informatik, Universität Augsburg, May 2003.
36. I. Németi. Dynamic algebras of programs. In *Proc. FCT'81 — Fundamentals of Computation Theory*, volume 117 of *LNCS*, pages 281–291. Springer, 1981.
37. T. Nipkow. More Church-Rosser proofs (in Isabelle/HOL). *J. Automated Reasoning*, 26(1):51–66, 2001.
38. Mathematics of Program Construction Group. Fixed point calculus. *Information Processing Letters*, 53:131–136, 1995.
39. B. Paige and S. Koenig. Finite differencing of computable expressions. *ACM Trans. Prog. Lang. and Syst.*, 4(3):402– 454, 1986.
40. S. Popkorn. *First Steps in Modal Logic.* Cambridge University Press, 1994.
41. V. Pratt. Dynamic algebras: Examples, constructions, applications. *Studia Logica*, 50:571–605, 1991.
42. G. Schmidt and T. Ströhlein. *Relations and Graphs.* EATCS Monographs in Computer Science. Springer, 1993.
43. N. Shankar. A mechanical proof of the Church-Rosser theorem. *Journal of the ACM*, 35(3):475–522, 1988.
44. J. M. Spivey. *Understanding Z.* Cambrigde University Press, 1988.
45. G. Struth. Non-symmetric rewriting. Technical Report MPI-I-96-2-004, Max-Planck-Institut für Informatik, 1996.
46. G. Struth. *Canonical Transformations in Algebra, Universal Algebra and Logic.* PhD thesis, Institut für Informatik, Universität des Saarlandes, 1998.
47. G. Struth. Calculating Church-Rosser proofs in Kleene algebra. In H.C.M. de Swart, editor, *Relational Methods in Computer Science, 6th International Conference*, volume 2561 of *LNCS*, pages 276–290. Springer, 2002.
48. G. Struth. An algebraic study of commutation and termination. Technical Report 2003-18, Institut für Informatik, Universität Augsburg, 2003.
49. V. Trnkova and J. Reiterman. Dynamic algebras with tests. *J. Comput. System Sci.*, 35:229–242, 1987.
50. B. von Karger. Temporal algebra. *Math. Struct. Comp. Science*, 8:277–320, 1998.
51. J. von Wright. From Kleene algebra to refinement algebra. In E. Boiten and B. Möller, editors, *Mathematics of Program Construction*, volume 2386 of *LNCS*, pages 233–262. Springer, 2002.