Science of Computer Programming 76 (2011) 181–203

Contents lists available at ScienceDirect

Science of Computer Programming

journal homepage: www.elsevier.com/locate/scico

Internal axioms for domain semirings

Jules Desharnais [a], Georg Struth [b,*]

[a] Département d'informatique et de génie logiciel, Pavillon Adrien-Pouliot, 1065, avenue de la Médecine, Université Laval, Québec, QC, G1V 0A6, Canada
[b] Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello, Sheffield S1 4DP, United Kingdom

ARTICLE INFO

Article history:
Available online 8 June 2010

Keywords:
Semiring
Kleene algebra
Modal semiring
Domain operator
Codomain operator
Antidomain operator
Algebra of domain elements
Distributive lattice
Boolean algebra
Heyting algebra
Löb's formula
Automated theorem proving

ABSTRACT

New axioms for domain operations on semirings and Kleene algebras are proposed. They generalise the relational notion of domain – the set of all states that are related to some other state – to a wide range of models. They are internal since the algebras of state spaces are induced by the domain axioms. They are simpler and conceptually more appealing than previous two-sorted external approaches in which the domain algebra is determined through typing. They lead to a simple and natural algebraic approach to modal logics based on equational reasoning. The axiomatisations have been developed in a new style of computer-enhanced mathematics by automated theorem proving, and the approach itself is suitable for automated systems analysis and verification. This is demonstrated by a fully automated proof of a modal correspondence result for Löb's formula that has applications in termination analysis.

1. Introduction

Semirings and Kleene algebras are foundational structures in computing, with applications covering program semantics, synthesis, verification and refinement, rewriting, discrete control and combinatorial optimisation. Most variants are close relatives of Kozen's elegant axiomatisation [16,17] and share some important features. They focus on the core operations for modelling programs and discrete sequential or concurrent systems. They allow abstract and concise reasoning about such systems within first-order equational logic. They have rich model classes that include relations, languages, paths in graphs, execution traces, processes and predicate transformers. And they enable a new kind of automated program analysis and verification supported by automated theorem provers, so-called ATP systems [11].

To link this algebraic approach in a uniform way with popular logics of programs such as dynamic, temporal or Hoare logics, modal operators have been added to semirings and Kleene algebras by axiomatising a notion of domain [5]. This generalises the domain operation for a relation $R \subseteq A \times A$, which is defined as $d(R) = \{a \in A : (a, b) \in R \text{ for some } b \in A\}$. More abstractly, an element of a semiring S can be interpreted as an action of a system, and the domain of this action models the set of all states from which the action is enabled. In a previous *external* two-sorted approach, the domain operation has therefore been axiomatised as a map of type $S \to B$, where B is a Boolean algebra – the state space – which is embedded as a subset into S [5]. Intuitively, by this embedding, each set of states is identified with a special action that corresponds to an observation or test of these states.

* Corresponding author. Tel.: +44 114 2221846; fax: +44 114 2221810.
E-mail addresses: Jules.Desharnais@ift.ulaval.ca (J. Desharnais), g.struth@dcs.shef.ac.uk (G. Struth).

0167-6423/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.scico.2010.05.007

It is well known that each relation R induces a map $\langle R\rangle : 2^A \to 2^A$ that is defined, for all $P \subseteq A$, as $\langle R\rangle P = \{a \in A : (a, b) \in R \text{ for some } b \in P\}$, and thus as the preimage of the subset P of A under R. This map is an operator on the powerset Boolean algebra 2^A in the sense of Jónsson and Tarski [15], and hence a (multi)modal diamond operator. This links Boolean algebras with operators and the standard modal semantics based on labelled transition systems and Kripke frames. In the semiring abstraction, diamonds can be defined from the domain operation as $\langle x\rangle p = d(xp)$, where $x \in S$, $p \in B$ and xp refers to the semiring product [21]. Modal semirings and modal Kleene algebras thus arise from semirings and Kleene algebras in a simple, natural way.

Nevertheless, the external approach to domain semirings left some questions open. First, the Boolean algebra in the definition of domain is not free. It must be the *maximal* Boolean subalgebra that can be embedded into the semiring [5]. Second, the property of being an element of the Boolean algebra can be expressed within the language of domain semirings. This rather unusual interdependency between types and axioms challenged us to develop *internal* one-sorted axiomatisations of domain on semirings where the algebras of domain elements are induced by the domain axioms and not declared through typing.

We provide several internal axiomatisations of domain operations on semirings. The first one consists of a simple set of equations that induces distributive lattices as domain algebras. It yields a new approach to distributive lattices with operators and modal algebras over distributive lattices. The second one uses an antidomain operation to induce a Boolean domain algebra from three equations plus the semiring axioms. This recovers the external approach to domain semirings and modal semirings in a much simpler setting. The introduction of the antidomain operation and the discovery of its fundamental role in axiomatising modal algebras is a particular contribution of this paper. We also further investigate the structure of domain algebras, for instance their existence and uniqueness. In addition, to demonstrate the flexibility of our approach, we show how (co-)Heyting algebras can be obtained as domain algebras and how (co-)Heyting algebras with operators or modal (co-)Heyting algebras can be defined in this setting.

The simplicity, generality and flexibility of the internal approach pays off in mathematical investigations and computing applications. Here, we present an automated proof of a modal correspondence result for Löb's formula that has applications in termination analysis. Free domain semirings and associated decision procedures can more easily be developed in the internal setting, too [13]. Furthermore, the approach is readily adaptable to variants of Kleene algebras, some of them based on near-semirings, with applications in program refinement, probabilistic protocol verification, game theory and concurrent systems analysis in the context of process algebras and action systems [6]. Finally, the approach has promising applications in automated termination analysis [24].

The development of small, irredundant, yet algebraically meaningful axioms may seem a tedious and time-consuming combinatorial task. Using ATP systems and model generators – in particular Prover9 in combination with Mace4 [20] and Waldmeister [10] – we could significantly speed up and simplify our analysis. Prover9 is an automated theorem prover which is complete for first-order equational logic. Mace4 is a counterexample generator which accepts the same input syntax as Prover9. Waldmeister is a theorem prover for pure equations. Many of the proofs in this paper can be obtained by either ATP system. This off-the-shelf ATP technology allowed us to focus almost entirely on concepts and delegate routine proofs to a machine. We therefore do not present calculational proofs, unless they provide important insights or reveal some useful technique in the way the domain operation can be used. All calculational proofs in this paper have been automatically verified, and most counterexamples have been automatically generated. ATP systems are push-button technology: they take a set of hypotheses and a proof goal as an input and either automatically produce an output, or terminate with failure, or run forever. No further user interaction is required. Note, however, that the proofs provided by ATP systems are not intended to be readable for humans. We therefore only provide some examples in an Appendix. All the tools used are easy to install on a computer and well documented. Using the templates provided in the Appendices, the hints in the text, and additional information and examples provided at our web site [23], everybody can instantly reproduce all the proofs and counterexamples in the paper. Additional proofs of our main theorems and the most important counterexamples have been published in a preliminary version [7].

The remainder of this text is organised as follows. Sections 2–6 introduce domain semirings which induce distributive lattices as domain algebras, develop their basic calculus and study some properties of domain algebras. Section 7 sets up the link between domain semirings, Kleene algebras with domain and distributive lattices with operators. Sections 8 and 9 introduce antidomain operations that turn domain algebras into Boolean algebras and relate these structures with the external axiomatisation. In Section 10, the domain algebras of domain semirings are extended to (co-)Heyting algebras. Section 11 presents an application of the new axiomatisation in automated termination analysis. Section 12 presents a conclusion and lists some questions for future research. Additional material is collected in the Appendices.

2. Semirings and dioids

Semirings are essentially rings without subtraction. Formally, a *semiring* is a structure $(S, +, \cdot, 0, 1)$ such that $(S, +, 0)$ is a commutative monoid, $(S, \cdot, 1)$ is a monoid, multiplication distributes over addition from the left and right and 0 is a left and right zero of multiplication. As usual in algebra, variants without 0 and 1 can easily be defined, but these are less interesting for our purposes.

The explicit semiring axioms as input for Prover9, Mace4 and Waldmeister can be found in Appendix A and Appendix B, respectively. We stipulate that multiplication binds more strongly than addition and we omit the multiplication symbol.

A standard semiring duality is *opposition*. It is obtained by swapping the order of multiplication (or by reading expressions from right to left). The opposite S^o of a semiring S is again a semiring, $S^{oo} = S$, and the opposite of each statement about a semiring holds in its opposite.

A semiring S is *idempotent*, or a *dioid*, if $1 + 1 = 1$. This is the case if and only if $x + x = x$ holds for all $x \in S$ (an ATP warm-up). For dioids, the relation $\leq$ defined, for all $x, y \in S$, by $x \leq y \Leftrightarrow x + y = x$ is a partial order. $(S, \leq)$ is a semilattice with addition as join and least element 0. Addition and multiplication are isotone with respect to that order. The dioid elements below 1 are called *subidentities*.

We will see that idempotency is a crucial ingredient of our domain axiomatisations. On the one hand, this excludes an adaptation of the approach to ring theory, since there are no nontrivial idempotent rings: it can easily be seen that the supremum $x + y$ can only be zero if both x and y are zero, and hence nontrivial elements cannot have inverses. On the other hand, this does not rule out meaningful axiomatisations for weaker structures such as near-semirings or even monoids and semigroups.

It is well known that dioids have many computationally meaningful models. We elaborate two of them to motivate our domain axioms.

Example 2.1. Consider the set $2^{A\times A}$ of all binary relations over A. For $R, S \in 2^{A\times A}$, let $R + S$ be $R \cup S$, let RS be the relative product $R \circ S = \{(a, b) : (a, c) \in R \text{ and } (c, b) \in S \text{ for some } c \in A\}$, let 0 be the empty relation $\emptyset$ and let 1 be the identity relation $1_A = \{(a, a) : a \in A\}$. Then $2^{A\times A}$ forms a dioid under these operations, the *full relation semiring* over A. Each of its subalgebras is again a dioid, a *relation semiring* over A.

Example 2.2. Let Σ_P and Σ_A be two disjoint alphabets. A *trace* over Σ_P and Σ_A is a finite sequence $\tau = p_0a_0p_1a_1 \ldots p_{n-1}a_{n-1}p_n$ in which elements of Σ_P and Σ_A alternate, and which begins and ends with an element of Σ_P. We write T for the set of all traces over Σ_P and Σ_A. Traces arise, for instance, when traversing automata or labelled transition systems. Elements of Σ_P then represent states, and elements of Σ_A are actions or labels. The product of two traces $\tau_1 = \sigma_1 p$ and $\tau_2 = q\sigma_2$ is the trace $\tau_1\tau_2 = \sigma_1 p\sigma_2$ if $p = q$; it is undefined if $p \neq q$. Intuitively, the trace $\tau_1\tau_2$ starts as τ_1 and finishes as τ_2.

Consider now the set 2^T of all sets of traces over T. For $X, Y \in 2^T$, let $X + Y$ be $X \cup Y$, let XY be the complex product $\{\tau_1\tau_2 : \tau_1 \in X \text{ and } \tau_2 \in Y\}$, let 0 be the empty set $\emptyset$ and let 1 be the set Σ_P of all traces of length one. Then 2^T is a dioid under these operations, the *full trace semiring* over T. Each of its subalgebras is again a dioid, a *trace semiring* over T.

Language and path semirings are special cases of trace semirings in which states or actions are forgotten. Alternatively, it is convenient to interpret the elements of a dioid as the actions of some system. Addition then models the choice between actions, multiplication their sequential composition, 0 the destructive action and 1 the ineffective action. Extensive discussions of dioid models (including domain) can be found in the literature [5,9].

3. The domain operation: motivation and external axioms

Before abstractly axiomatising the domain operation on semirings, we motivate it through three examples. We also introduce the previous external axiomatisation [5] as a point of reference.

Example 3.1. In set theory, the domain of a binary relation $R \subseteq A \times A$ is given by the set of all $a \in A$ such that $(a, b) \in R$ for some $b \in A$. The map that takes a relation in $2^{A\times A}$ to its domain – a set in 2^A – can be turned into an endomorphism d on the relation semiring $2^{A\times A}$ by defining

$$d(R) = \{(a, a) \in A \times A : (a, b) \in R \text{ for some } b \in A\}.$$

Thus $d(R)$ "encodes" the domain of R as a subset of the identity relation on A. It is easy to see that $d(R)$ can explicitly be defined as $R \circ U \cap 1_A$, where U is shorthand for the universal relation $A \times A$. Intuitively, $R \circ U$ links all inputs to R with any other output from A, and intersecting with 1_A projects that relation to the set of pairs (a, a) with $(a, b) \in R$.

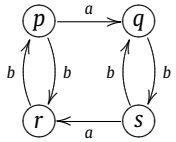
Example 3.2. The domain operation on a set X of traces on a trace semiring 2^T yields the set of all starting states of traces in X; that is,

$$d(X) = \{p \in \Sigma_P : p\sigma \in X \text{ for some sequence } \sigma\}.$$

All elements of $d(X)$ are traces of length one. Compared to the case of relations, it is now less clear how a domain operation should be defined algebraically in terms of equations.

More generally, domain elements can be associated with *enabledness conditions* that indicate when an action of a system can be executed.

Example 3.3. Consider the following labelled transition system, in which the labels a and b of the arrows represent actions or transitions of the system and the nodes p, q, r and s are states.



The domain of a is the set $\{p, s\}$. It is the set of those states from which action a is enabled. Similarly, the domain of b is $\{p, q, r, s\}$.

The examples suggest that the domain operation should map actions of a dioid S to sets of states, for which a Boolean algebra B is a suitable abstraction. Following Kozen's approach to *Kleene algebras with tests* [17], it also seems appropriate to require that B be embedded into S as a subalgebra such that the maximal element of B is identified with 1 of S, the minimal element of B with 0, and such that Boolean join and meet coincide with addition and multiplication. This yields a domain operation of type $S \to B$.

Obviously, a main defining property of an enabledness condition $d(x)$ for an action $x \in S$ is that executing $d(x)$ before x leaves x unchanged: $d(x)x = x$. But since $d(x) \leq 1$ by definition, only $x \leq d(x)x$ really matters. We call $d(x)$ a *left preserver* of x because it satisfies this property. Note that – in the tradition of Kleene algebra with tests – the embedding in the product $d(x)x$ has not been spelled out. A more careful notation would require us to write $\iota(d(x))x$, where $\iota : B \to S$ is the embedding of B into S.

Furthermore, $d(x)$ is not an arbitrary left preserver. It is the *least* element with that property, since decreasing $d(x)$ any further would restrict x. In Example 3.3, for instance, enabling the action a from state p only restricts a, because it can no longer be executed from state s. Algebraically, the fact that $d(x)$ is a least left preserver of x can, for all $x \in S$ and $p \in B$, be expressed as

$$x \leq px \Leftrightarrow d(x) \leq p.$$

This *least left preservation* property is in all dioids equivalent to the identities $x \leq d(x)x$ and $d(px) \leq p$ [5].

Lemma 3.4. *Domain elements are a least left preserver in relation and trace semirings.*

Proof. The proof for relations is folklore. First, an ordered pair (a, b) is in $d(R) \circ R$ if and only if $(a, a) \in d(R)$ and $(a, b) \in R$, and hence if and only if $(a, b) \in R$. Second, suppose that some subidentity P does not contain $d(R)$; that is, $(a, a) \notin P$ for some $(a, a) \in d(R)$. Then $P \circ R \neq R$, since R contains ordered pairs with left coordinate a, but these are not in $P \circ R$.

The proof for traces is similar. First, each trace in $d(X)X$ must be of the form $p\sigma$ with $p \in d(X)$ and $p\sigma \in X$. By the definition of domain on trace semirings, this is the case if and only if $p\sigma \in X$, and therefore $d(X)X = X$. Second, suppose that some set Y does not contain $d(X)$; that is, $p \notin Y$ holds for some $p \in d(X)$. Then $YX \neq X$ because all traces in X that start with p will either vanish or become strictly longer in the product YX. Hence $d(X)$ is indeed the *least* left preserver of X. □

Another important property is the *locality* of domain. Consider the enabledness of ba in Example 3.3. Clearly, the sequence of actions ba is enabled in the states q and r, but not in p and s (b is enabled in all states). But it turns out that the target states of a-arrows are irrelevant for determining $d(ba)$. Only those states in which a is enabled need to be considered. Hence $d(ba)$ should be the set of all states from which execution of b leads to states in which a is enabled: $d(ba) = d(bd(a))$.

Lemma 3.5. *The domain operation is local in relation and trace semirings.*

Proof. By the definition of the relative product and the domain operation on relation semirings, $d(R \circ S) = d(R \circ d(S))$ holds because

$$\begin{aligned}
(a, a) \in d(R \circ S) &\Leftrightarrow \exists b.(a, b) \in R \circ S \\
&\Leftrightarrow \exists b, c.((a, c) \in R \wedge (c, b) \in S) \\
&\Leftrightarrow \exists c.((a, c) \in R \wedge \exists b.(c, b) \in S) \\
&\Leftrightarrow \exists c.((a, c) \in R \wedge (c, c) \in d(S)) \\
&\Leftrightarrow \exists c.(a, c) \in R \circ d(S) \\
&\Leftrightarrow (a, a) \in d(R \circ d(S)).
\end{aligned}$$

The proof for traces is again very similar:

$$\begin{aligned}
p \in d(XY) &\Leftrightarrow \exists \tau.p\tau \in XY \\
&\Leftrightarrow \exists \tau.\exists \tau_1, q, \tau_2.(p\tau_1 q \in X \wedge q\tau_2 \in Y \wedge p\tau = p\tau_1 q\tau_2) \\
&\Leftrightarrow \exists \tau_1, q, \tau_2.(p\tau_1 q \in X \wedge q\tau_2 \in Y \wedge \exists \tau.p\tau = p\tau_1 q\tau_2) \\
&\Leftrightarrow \exists \tau_1, q, \tau_2.(p\tau_1 q \in X \wedge q\tau_2 \in Y) \\
&\Leftrightarrow \exists \tau_1, q.(p\tau_1 q \in X \wedge \exists \tau_2.q\tau_2 \in Y) \\
&\Leftrightarrow \exists \tau_1, q.(p\tau_1 q \in X \wedge q \in d(Y)) \\
&\Leftrightarrow \exists \tau.p\tau \in Xd(Y) \\
&\Leftrightarrow p \in d(Xd(Y)).
\end{aligned}$$

□

These observations motivate the following external axiomatisation [5].

Let S be a dioid with an embedded Boolean algebra B such that 0 is the minimal element of B and 1 its maximal element. A *domain function* $d : S \to B$ satisfies, for all $x, y \in S$ and $p \in B$, the axioms

$$x \leq d(x)x, \tag{1}$$
$$d(px) \leq p, \tag{2}$$
$$d(xd(y)) \leq d(xy). \tag{3}$$

Every dioid has an embedded Boolean algebra (take $B = \{0, 1\}$). It turns out that, unlike for Kleene algebras with tests, the axioms force B to be maximal. Indeed, it can be shown that each complemented subidentity of S must be in B [5] (see also Theorem 6.12 below).

Another important observation is that the property "x is a domain element" can be expressed not only *externally* as $x \in B$, but also *internally* in the language of dioids with domain as $x = d(x)$. The proof of a corresponding fact in the internal setting (Corollary 4.3) can easily be adapted.

This suggests that the domain operation can take care of itself and the rather unusual interdependency between the external axioms and the typing constraints of domain should be avoided.

4. Domain semirings

Based on the examples and discussions of the previous section, we now present a set of *internal* domain axioms for semirings in which the domain operation is a one-sorted map, a semiring endomorphism. Domain now does take care of itself in the sense that its axioms induce an appropriate algebraic structure on the state space. The axioms can further be justified by the large set of "natural" properties they imply, in particular the least left preserver property. To simplify notation and statements, we do not add the adjective "internal" to our definitions.

A *domain semiring* is a semiring S extended by a *domain operation* $d : S \to S$ which, for all $x, y \in S$, satisfies the axioms

$$x + d(x)x = d(x)x, \tag{D1}$$
$$d(xy) = d(xd(y)), \tag{D2}$$
$$d(x) + 1 = 1, \tag{D3}$$
$$d(0) = 0, \tag{D4}$$
$$d(x + y) = d(x) + d(y). \tag{D5}$$

The following fact can easily be shown by any ATP system. We present the proof generated by Waldmeister as an example in Appendix C, and encourage the reader to redo the proof using the templates provided in the Appendices and at our web site [23].

Proposition 4.1. *Domain semirings are idempotent.*

Hence every domain semiring can be ordered, and we can write $x \leq d(x)x$ in (D1) and $d(x) \leq 1$ in (D3). By axiom (D1), all domain elements are left preservers, and Lemma 5.1(xi) below shows that they are even least left preservers, as expected. By axiom (D2), domain is local (with respect to multiplication). By axiom (D3), domain elements are subidentities. By axiom (D4) and (D5), domain is strict and additive. All domain axioms capture natural properties which hold in all relation and trace semirings.

The set of domain elements is $d(S)$, the image of S under the domain operation. We will consistently use the letters p, q, r to denote domain elements. The following general fixpoint lemma allows us to characterise domain elements internally within the language of domain semirings.

Proposition 4.2. *Let $f : A \to A$ be a projection function ($f \circ f = f$). Then*

$$x \in f(A) \Leftrightarrow f(x) = x.$$

Proof. If $x \in f(A)$, then $x = d(y)$ for some $y \in S$, and therefore $f(x) = f(f(y)) = f(y) = x$. Conversely, $f(x) = x$ trivially implies that $x \in f(A)$. □

This simple fact can widely be applied for typing elements.

Corollary 4.3. *The domain elements of a domain semiring are the fixpoints of the domain operation.*

Proof. Domain is a projection: $d(d(x)) = d(1d(x)) = d(1x) = d(x)$ by (D2). □

Codomain semirings arise as the opposites of domain semirings. A *codomain semiring* is a semiring S extended by the *codomain operation* $d^o : S \to S$ that satisfies, for all $x, y \in S$,

$$\begin{aligned} x + xd^o(x) &= xd^o(x), \\ d^o(xy) &= d^o(d^o(x)y), \\ d^o(x) + 1 &= 1, \\ d^o(0) &= 0, \\ d^o(x + y) &= d^o(x) + d^o(y). \end{aligned}$$

By duality, the opposites of all statements about domain semirings hold in codomain semirings. Only the interaction of domain and codomain deserves further investigation.

It is certainly interesting to determine whether the domain axioms of domain semirings are irredundant; that is, whether no domain axiom is entailed by the semiring axioms and the remaining domain axioms. Although there is no general guarantee that irredundancy of an axiom set can be established through (small) finite models alone, Mace4 made this task an easy exercise.

Proposition 4.4. *The domain axioms of domain semirings are irredundant.*

Proof. We used Mace4 to find models that satisfy the semiring axioms plus all combinations of four domain axioms except the fifth one. The interested reader can again use the templates in Appendix A to reproduce them.

We first show the irredundancy of (D1). Consider the *Boolean semiring* with elements 0 and 1 and with addition and multiplication defined by the tables below. Let domain be defined as in the following table.

d	0	1
	0	0

$+$	0	1
0	0	1
1	1	1

$\cdot$	0	1
0	0	0
1	0	1

Then (D2)–(D5) hold, but not (D1), since $0 \cdot 1 = 0 \neq 1 = 1 + 0 \cdot 1$.

The irredundancy of (D2) follows by setting $x = y = 2$ in the model

d	0	1	2
	0	1	1

$+$	0	1	2
0	0	1	2
1	1	1	1
2	2	1	2

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	0

The irredundancy of (D3) follows by setting $x = 1$ in the model

d	0	1	2
	0	2	2

$+$	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	2

The irredundancy of (D4) can be established in the model

d	0	1
	1	1

$+$	0	1
0	0	1
1	1	1

$\cdot$	0	1
0	0	0
1	0	1

The irredundancy of (D5) follows by setting $x = 1$ and $y = 2$ in the model

d	0	1	2	3
	0	1	3	3

$+$	0	1	2	3
0	0	1	2	3
1	1	1	2	1
2	2	2	2	2
3	3	1	2	3

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	2	2
3	0	3	2	3

□

It usually takes far less time to generate these models than to understand them. It is equally easy to show with Mace4 that the additivity and the locality axioms cannot further be weakened to inequalities.

5. Basic domain calculus

We now present a basic calculus of domain semirings. All laws stated hold on relation and trace domain semirings. They have all been automatically verified with Waldmeister and Prover9.

Lemma 5.1. *Let S be a domain semiring. Let $x, y \in S$ and let $p \in d(S)$. Then*

(i) $d(x)x = x$ *(domain is a left invariant),*
(ii) $d(p) = p$ *(domain is a projection),*
(iii) $d(xy) \leq d(x)$ *(domain increases for prefixes),*
(iv) $x \leq 1 \Rightarrow x \leq d(x)$ *(domain expands subidentities),*
(v) $d(x) = 0 \Leftrightarrow x = 0$ *(domain is very strict),*
(vi) $d(1) = 1$ *(domain is co-strict),*
(vii) $x \leq y \Rightarrow d(x) \leq d(y)$ *(domain is isotone),*
(viii) $d(px) = pd(x)$ *(domain elements can be exported),*
(ix) $d(x)d(x) = d(x)$ *(domain elements are multiplicatively idempotent),*
(x) $d(x)d(y) = d(y)d(x)$ *(domain elements commute),*
(xi) $x \leq px \Leftrightarrow d(x) \leq p$ *(domain elements are least left-preservers),*
(xii) $xy = 0 \Leftrightarrow xd(y) = 0$ *(domain is weakly local).*

As an example, we present the input file and proof output of the least left preserver property (Lemma 5.1(xi)) with Prover9 in Appendix D.

Lemma 5.1(viii) displays a variant of the external domain axiom (2). There, additivity of domain is redundant in the presence of this axiom. Here, the situation is different. If the additivity axiom (D5) is replaced by Lemma 5.1(viii), then (D5) cannot be derived. Mace4 presents the following four-element model, for which $d(1 + 2) = d(2) = 3 \neq 1 = 1 + 3 = d(1) + d(2)$.

d	0	1	2	3
	0	1	3	3

$+$	0	1	2	3
0	0	1	2	3
1	1	1	2	1
2	2	2	2	2
3	3	1	2	3

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	2	2
3	0	3	2	3

Lemma 5.1(xi), least left preservation, is a particularly important property that underpins that our axioms are natural. By this property,

$$d(x) = \inf(p \in d(S) : x \leq px).$$

Mace4 can show that the restriction of p to domain elements cannot be relaxed. Finally, the weak locality property in Lemma 5.1(xii) is also very intuitive for trace or relation semirings, where it models the pointwise behaviour of relative products and trace products when glueing relations or traces together. In the external axiomatisation, weak locality is equivalent to locality and it holds if and only if $xy = 0 \Leftrightarrow d^o(x)d(y) = 0$. With the internal axioms, a four-element model proves that weak locality does not imply locality. Section 9 shows that equivalence holds for Boolean domain semirings.

Additivity of domain extends to arbitrary suprema, which may seem surprising. In general, a semilattice endomorphism $f : L \to L$ is *completely additive* if $f(\sup(X : X \subseteq L) = \sup(f(X) : X \subseteq L)$, whenever $\sup(X : X \subseteq L)$ exists.

Proposition 5.2. *Domain operators on domain semirings are completely additive.*

Proof. Let S be a domain semiring and let $y = \sup(A)$ for some $A \subseteq S$. We show that $d(y)$ is the least upper bound of $d(A)$. Since domain is isotone, $d(y) \geq d(x)$ for all $x \in A$ and $d(y)$ is an upper bound of $d(A)$.

So let $p \in d(S)$ be another upper bound of $d(A)$; that is, $p \geq d(x)$ for all $x \in A$. By least left preservation (Lemma 5.1(xi)), therefore, $x \leq px$, and $x \leq py$ follows since $x \leq y$. Hence py is also an upper bound of A. But then $y \leq py$, and therefore $d(y) \leq p$ by least left preservation; that is, $d(y)$ is a least upper bound of $d(A)$. Thus we can conclude that $d(\sup(A)) = d(y) = \sup(d(A))$, whence d is completely additive. □

The domain calculus also helps us to investigate the existence of domain semirings.

Proposition 5.3. *Some semirings cannot be extended to domain semirings.*

Proof. Mace4 can verify that the following tables define a dioid.

$+$	0	1	2
0	0	1	2
1	1	1	1
2	2	1	2

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	0

By Lemma 5.1(v), $d(x) = 0 \Leftrightarrow x = 0$. Now consider $d(2)$. If $d(2) = 1$, then $d(2 \cdot d(2)) = 1 \neq 0 = d(2 \cdot 2)$ and (D2) is violated. If $d(2) = 2$, then $d(2) \cdot 2 = 0 \not\geq 2$ and (D1) is violated. □

The fact that domain cannot be defined on some very small finite semirings might seem surprising. It is well known that each complete semilattice is also a complete lattice; that is, arbitrary infima and suprema exist. In particular, finite semilattices are complete. Therefore, all infima needed for defining least left preservers exist in the above three-element dioid. In fact, the clash in the proof of Proposition 5.3 is with respect to (D2). But an analysis with Prover9 and Mace4 also shows that (D2) is needed in the proof of the least left preservation property. The identification of existence conditions for domain on semirings is certainly an interesting subject for further investigation. This seems to be closely related to finding meaningful axiomatisations of domain that imply least left preservation, but not locality.

6. Domain algebras

Another main criterion for the quality of the domain axioms are the domain algebras they induce: that is, the algebraic structure of the state space corresponding to domain elements. According to the discussion in Section 3, Boolean algebras would be natural candidates, but since our domain axioms seem rather unrelated to complementation, we should perhaps expect a weaker kind of lattice. By Corollary 4.3, domain elements are fixpoints of the domain operation. This yields a powerful tool for analysing the structure of the set of domain elements.

It is well known that the set of subidentities of a dioid forms a dioid under addition and multiplication. This is the starting point for the following investigation.

Proposition 6.1.

(i) *Let S be a domain semiring. Then $(d(S), +, \cdot, 0, 1)$ is a dioid.*
(ii) *A semiring S is a distributive lattice if $x + 1 = 1$ and $xx = x$ hold for all $x \in S$.*
(iii) *Let S be a domain semiring. Then $(d(S), +, \cdot, 0, 1)$ is a bounded distributive lattice.*

Proof.

(i) Corollary 4.3 implies that $d(S)$ is closed under addition and multiplication if 0, 1, $p + q$ and pq are fixpoints of d. Waldmeister or Prover9 could show that this is indeed the case.
(ii) See Birkhoff's textbook on lattice theory [1].
(iii) By (i), (ii), Lemma 5.1(ix) and axiom (D3). □

We henceforth call $d(S)$ the *domain algebra* of S. According to Proposition 6.1(iii), domain semirings have state spaces with an appropriate algebraic structure. The following statements consider this structure in more detail.

We now relate the domain algebras of domain semirings with three other sets: the set of all subidentities, the set of all multiplicatively idempotent subidentities and the set of all commutative subidentities (that is, subidentities x and y that satisfy $xy = yx$). We also investigate the uniqueness of domain algebras.

Let S be a dioid. We write $1(S)$ for its set of subidentities, $C(S)$ for its subset of commutative subidentities and $I(S)$ for its set of multiplicatively idempotent subidentities. The relationship between these sets is very heterogeneous.

Lemma 6.2. *In the class of domain semirings, the following properties are satisfiable; that is, for each property, there exists at least one domain semiring where it holds.*

(i) $d(S) \subset 1(S)$.
(ii) $d(S) \subset I(S)$.
(iii) $d(S) \subset C(S)$.
(iv) $I(S) \subset 1(S)$.
(v) $C(S) \subset 1(S)$.
(vi) $C(S) \not\subseteq I(S)$ *and* $I(S) \not\subseteq C(S)$.

Proof. This is an exercise in constraint programming with first-order logic. The formulas provided in the proofs below must be given as goals to Mace4 together with the template in Appendix A.

(i) $d(S) \subseteq 1(S)$ by definition; hence we must show that $d(S) = 1(S)$ need not hold. Giving the formula

$$\forall x.(x \leq 1 \Rightarrow d(x) = x)$$

as a goal to Mace4 yields a three-element counterexample: that is, a domain semiring S in which

$$\exists x.(x \leq 1 \wedge d(x) \neq x)$$

is true. This proves the claim.
(ii)–(v) The proofs are very similar. Mace4 yielded three-element models for (ii) and (iii) and four-element models for (iv) and (v).
(vi) Mace4 gave a five-element counterexample to the goal

$$\forall x, y.(x \leq 1 \wedge y \leq 1 \wedge xy = yx \Rightarrow xx = x) \vee \forall x, y.(x \leq 1 \wedge y \leq 1 \wedge xx = x \wedge yy = y \Rightarrow xy = yx). \quad \square$$

Lemma 6.2(iv)–(vi) show that neither $1(S)$, $I(S)$ nor $C(S)$ must form a lattice.

Lemma 6.3. *Let S be a domain semiring.*

(i) *$C(S)$ is closed under addition and multiplication.*
(ii) *$I(S)$ is closed under addition, but not necessarily under multiplication.*

The closure conditions have been verified by Prover9. Mace4 presented a five-element model that refutes closure under multiplication.

The next result concerns uniqueness of domain algebras.

Lemma 6.4. *Domain algebras of domain semirings need not be unique, and need not include each other.*

Proof. Let the maps f and g satisfy the domain axioms. Mace4 yielded a three-element counterexample for $f(x) = g(x)$ and a four-element counterexample for $(\forall x.g(f(x)) = f(x)) \vee (\forall x.f(g(x)) = g(x))$. □

Again, the formulas given can directly be provided as goals for Mace4.

Trivially, if $d(S) = 1(S)$, then the domain algebra is unique, since the set of all subidentities is obviously unique in each dioid. But this result can be sharpened.

Proposition 6.5. *Let S be a domain semiring.*

(i) *d has a unique extension to all $x \in S$ if it is uniquely defined on $I(S)$.*
(ii) *d is uniquely defined if $I(S) \subseteq d(S)$.*

Proof. Again, the formulas below only need to be copied as goals in the template for Prover9 given in Appendix A.

(i) Let the maps f and g satisfy the domain axioms. Then Prover9 showed that

$$\forall x.(x \leq 1 \wedge xx = x \Rightarrow f(x) = g(x)) \Rightarrow \forall x.f(x) = g(x).$$

(ii) Let the maps f and g satisfy the domain axioms and also $I(S) \subseteq f(S)$ and $I(S) \subseteq g(S)$. By the definition of set inclusion and the fixpoint lemma (Corollary 4.3), this is equivalent to

$$\forall x.(x \leq 1 \wedge xx = x \Rightarrow f(x) = x) \quad \text{and} \quad \forall x.(x \leq 1 \wedge xx = x \Rightarrow g(x) = x).$$

Since the formula $\forall x.(\phi \Rightarrow f(x) = x) \wedge \forall x.(\phi \Rightarrow g(x) = x) \Rightarrow \forall x.(\phi \Rightarrow f(x) = g(x))$ holds in first-order logic, the assumption of (i) holds and therefore $f = g$. □

Lemma 6.6. *Let S be a domain semiring. If $C(S) \subseteq d(S)$, then $I(S) \subseteq d(S)$, but not necessarily vice versa.*

The proof is by Prover9; a four-element counterexample has been given by Mace4.

Corollary 6.7. *Let S be a domain semiring.*

(i) *d has a unique extension to all $x \in S$ if it is uniquely defined on $C(S)$.*
(ii) *d is uniquely defined if $C(S) \subseteq d(S)$.*

Proof. Let $C(S) \subseteq d(S)$. Then $I(S) \subseteq d(S)$ by Lemma 6.6, and (i) and (ii) follow from Proposition 6.5. □

In the external axiomatisation, the domain algebra is a Boolean subalgebra by definition. A three-element counterexample produced by Mace4 shows that for domain semirings this need not be the case.

Lemma 6.8. *The domain algebras of domain semirings need not be Boolean algebras.*

Proof. Mace4 shows that it is not the case that for all x there exists a y such that $d(x) + d(y) = 1$ and $d(x)d(y) = 0$. Consider $x = 2$ in the following domain semiring:

d	0	1	2
	0	1	2

$+$	0	1	2
0	0	1	2
1	1	1	1
2	2	1	2

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	2

□

Are all domain lattices (co-)Heyting algebras? This cannot not be refuted by a finite counterexample, since all finite distributive lattices are (co-)Heyting algebras. We leave it as an open question.

We now present a sufficient condition that characterises certain domain elements without even mentioning the domain operation. This also links subidentities with domain elements and relates domain algebras with Boolean subalgebras of subidentities.

Proposition 6.9. *An element x of a domain semiring S is a domain element if some $y \in S$ satisfies*

$$x + y = 1 \quad \textit{and} \quad yx = 0.$$

Proof. We first show that $xd(x) = x$ and that $yd(x) = 0$. For the first identity, $xd(x) \leq x$ holds by (D3) and the converse inequality holds, since, by Lemma 5.1(i), $x = (x + y)x = xx + yx = xx = xd(x)x \leq xd(x)$. The second identity holds, since, by Lemma 5.1(i), (D2) and (D4),

$$yd(x) = d(yd(x))yd(x) = d(yx)yd(x) = 0yd(x) = 0.$$

Then $d(x) = (x + y)d(x) = xd(x) + yd(x) = x + 0 = x$. □

An automated proof with Prover9 took less than one second. The condition $yx = 0$ is quite sensitive: it does not imply that $xy = 0$, although $y \leq 1$ since $x + y = 1$. Mace4 found a five-element counterexample.

d	0	1	2	3	4
	0	1	1	3	3

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	1	1	1	1
2	2	1	2	1	2
3	3	1	1	3	3
4	4	1	2	3	4

$\cdot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	2	0	0
3	0	3	4	3	4
4	0	4	4	0	0

It is, however, straightforward to further constrain the assumptions to enforce that both x and y are domain elements.

The following fact is immediate from Proposition 6.9.

Corollary 6.10. *Two elements x and y of some domain semiring are domain elements if*

$$x + y = 1, \quad xy = 0 \quad \textit{and} \quad yx = 0.$$

We call an element x of a dioid S *complemented* if some $y \in S$ satisfies the three conditions above. The set of all complemented elements in S is denoted B_S.

Lemma 6.11. *Let S be a dioid. Then $(B_S, +, \cdot, 0, 1)$ is a Boolean algebra.*

Proof. Let $x, y \in S$ be complemented. Prover9 or Waldmeister could then show that $xx = x$, $xy = yx$, that $x + y$ and xy are complemented and that each complemented element is complemented. Therefore, since $x \leq 1$ and by Lemma 6.1(ii), B_S is closed under the Boolean operations and forms a distributive lattice. It is complemented by assumption. But Boolean algebras are complemented distributive lattices. □

Theorem 6.12. *Let S be a domain semiring. Then $d(S)$ contains the greatest Boolean subalgebra of S bounded by* 0 *and* 1.

Proof. By Corollary 6.10 and Lemma 6.11, $B_S \subseteq d(S)$. □

Theorem 6.12 provides further interesting insights into the structure of domain algebras. In relation semirings and trace semirings, for instance, where the entire subalgebra of subidentities is a Boolean algebra, the domain algebra is fixed. It is the full algebra of subidentities, as expected.

The three equalities in Corollary 6.10 were used by von Wright to define *guards*[1] in refinement algebras [27], which are semirings with additional operators. Guards form a Boolean algebra. Our purpose here is not to define a Boolean algebra of domain elements (this will be done with internal axioms in Section 8), but to prove Theorem 6.12, because it reveals an important fact about the distributive lattice of domain elements.

By duality, the results of this section immediately translate to codomain semirings. In particular, the codomain algebras of codomain semirings are also distributive lattices. But domain and codomain algebras need not coincide.

Lemma 6.13. *Some domain and codomain semirings have different domain and codomain algebras.*

Proof. Mace4 presents three-element models in which the identities $d(d^o(x)) = d^o(x)$ and $d^o(d(x)) = d(x)$ do not hold or in which $d(x) = x$ and $d^0(x) = x$ are not equivalent. Hence some antidomain elements are not domain elements and vice versa. □

To enforce that $d(S) = d^o(S)$ it is therefore necessary to add the law $d(x) = x \Leftrightarrow d^o(x) = x$, or the two identities $d(d^0(x)) = d^0(x)$ and $d^o(d(x)) = d(x)$, which are equivalent to this law. Whether this is desirable or not may depend on the application, but it seems very natural to require that domain and codomain elements live in the same state space. It can also be shown that adding just one identity does not suffice, since none of them implies the other.

7. Semiring modules and modal semirings

It has been shown that modal operators can be defined via domain and codomain operations in the external axiomatisation and applied in various contexts [5,21]. This can best be explained for relation semirings.

Example 7.1. In set theory, the preimage of a set $P \in 2^A$ under a relation $R \subseteq A \times A$ is defined as the set $\{a \in A : (a, b) \in R \text{ for some } b \in P\}$. This can be adapted to relation semirings by defining

$$\langle R\rangle P = \{(a, a) \in A \times A : (a, b) \in R \text{ for some } (b, b) \in P\} = d(R \circ P).$$

Let I_A denote the subidentities on A; that is, the set of relations on A included in the identity relation. It can be shown that the endomorphism $\langle R\rangle : I_A \to I_A$ on the Boolean algebra I_A is strict ($\langle R\rangle\emptyset = \emptyset$) and additive ($\langle R\rangle(P \cup Q) = \langle R\rangle P \cup \langle R\rangle Q$). The structure $(I_A, (\langle R\rangle : R \subseteq A \times A))$ therefore forms a *Boolean algebra with operators* à la Jónsson and Tarski [15]. More common models of modal logics are *Kripke frames*: that is, relational structures $(A, (R_i \subseteq A \times A : i \in I))$, where subsets of A are assigned to logical variables (the set of states in which this variable is true) and in which each R_i interprets a modal connective. Via the notion of preimage, a Boolean algebra with operators can therefore be constructed from the powerset algebra of a given frame. This *algebraises* the underlying modal logic in the sense of Blok and Pigozzi [2].

[1] Guards are similar to tests in Kleene algebra with tests [17].

More abstractly, a map $\langle x\rangle : d(S) \to d(S)$ can be defined for each element x of a domain semiring S by

$$\langle x\rangle p = d(xp),$$

where $p \in d(S)$. Obviously, $d(xp)$ abstractly models the preimage of the set of states p under the action x.

Dually, a map $\langle x\rangle^o : d^o(S) \to d^o(S)$ can be defined on codomain semirings as $\langle x\rangle^o p = d^o(px)$. It models the image of p under x.

In order to justify that these diamond operators are indeed modal operators in the sense of Boolean algebras with operators, we must show that the maps $\langle x\rangle$ are strict and additive. To link domain semirings even more tightly with computational algebras and logics, we present a more general result.

A *semiring module* [8] is a structure $(S, L, :)$, such that S is a dioid, L is a semilattice (with least upper bound operation $+$ and least element 0) and the scalar product : of type $S \times L \to L$ satisfies, for all $x, y \in S$ and $p, q \in L$, the axioms

$$\begin{aligned}(x + y) : p &= x : p + y : p,\\ x : (p + q) &= x : p + x : q,\\ (xy) : p &= x : (y : p),\\ 1 : p &= p,\\ x : 0 &= 0.\end{aligned}$$

The scalar product : binds most strongly among the binary operators.

Proposition 7.2. *Let S be a domain semiring. Then $(S, d(S), (\lambda x, p.\langle x\rangle p))$ is a semiring module.*

Proof. It is routine work to verify $d((x + y)d(z)) = d(xd(z)) + d(yd(z))$ or $d((x + y)z) = d(xz) + d(yz)$ and the other identities with Prover9 or Waldmeister. □

The second and the fifth module axiom say that all diamonds $\langle x\rangle$ are additive and strict. Domain semirings therefore induce distributive lattices with operators à la Jónsson and Tarski, and hence modal algebras. In order to emphasise this fact we call domain semirings *modal semirings*.

To link modal semirings more strongly with logics of programs such as dynamic, temporal and Hoare logics, and in order to prepare the application in Section 11, a notion of iteration is needed.

A *Kleene algebra* is a dioid S extended by the *star operation* $^* : S \to S$ that satisfies the unfold axiom and the induction axiom

$$1 + xx^* = x^* \quad \text{and} \quad y + xz \leq z \Rightarrow x^*y \leq z,$$

and their opposites [16]. A *Kleene algebra with domain* – also called *modal Kleene algebra* – is a domain semiring that is also a Kleene algebra. A *Kleene module* [18] is a semiring module $(K, L, :)$ over a Kleene algebra K that satisfies, for all $x \in K$ and $p, q \in L$, the induction axiom

$$p + x : q \leq q \Rightarrow x^* : p \leq q.$$

Proposition 7.3. *Let K be a Kleene algebra with domain. Then $(K, d(K), (\lambda x, p.\langle x\rangle p))$ is a Kleene module.*

Proof. By Proposition 7.2, K is a semiring module. It remains to show that

$$d(y) + d(xd(z)) \leq d(z) \Rightarrow d(x^*d(y)) \leq d(z)$$

holds for all Kleene algebras with domain. This could be done by Prover9, but the proof took more than 30 min and it has about 150 resolution steps. □

This automation is a first indication of the power of the internal axiomatisation. Previous attempts to automatically prove the above induction axiom with the external axiomatisation failed.

When L is a Boolean algebra, Kleene modules are essentially algebraic variants of propositional dynamic logics, and the operators of the temporal logics LTL and CTL can of course be defined in this setting. In this sense, the modal Kleene algebras of this section correspond to propositional dynamic logics defined over distributive lattices of propositions. The absence of Boolean complementation certainly increases both the efficiency of proof search and the range of applicability.

It is easy to show with Mace4 that the module axioms are too weak to imply the domain axioms. To this end, of course, we must assume that the semilattice L contains a greatest element 1 in order to define $d(x) = x : 1$. It is then also easy to show that the module laws imply that $d^2(x) = d(x)$, that domain elements are closed under addition but not under multiplication, and that domain elements are not idempotent and do not commute with respect to multiplication. So, in the finite case, the structure induced by the image of domain defined via the scalar product is a complete semilattice, and hence a lattice, but meet and multiplication need not coincide. In the infinite case, the domain algebra induced is only a semilattice.

Similar arguments can, of course, be applied to the dual diamonds $\langle x\rangle^o$. Therefore, on domain and codomain semirings, two different kinds of diamonds exist. In modal logical parlance, $\langle x\rangle$ is a *forward diamond operator*, and we sometimes write $|x\rangle$ to indicate this fact. Similarly, $\langle x\rangle^o$ is a *backward diamond operator*, for which we sometimes write $\langle x|$. In the context of Boolean algebras with operators, an important symmetry between forward and backward diamonds is expressed by conjugation. Formally, two endomorphisms f and g on a lattice L are *conjugates* if $f(p)q = 0 \Leftrightarrow pg(q) = 0$ holds for all $p, q \in L$ (juxtaposition denoting meet). This fact holds already in the weaker setting of domain algebras that are distributive lattices.

Proposition 7.4. *Let S be a domain and codomain semiring such that $d(S) = d^o(S)$. Then, for all $x \in S$, the operators $|x\rangle$ and $\langle x|$ are conjugates; that is, for all $p, q \in d(S)$,*

$$p|x\rangle q = 0 \Leftrightarrow q\langle x|p = 0.$$

Proof. By weak locality (Lemma 5.1(xii)) and its dual, or by Prover9,

$$p|x\rangle q = 0 \Leftrightarrow pd(xq) = 0 \Leftrightarrow pxq = 0 \Leftrightarrow d^o(px)q = 0 \Leftrightarrow q\langle x|p = 0. \quad \square$$

In the external axiomatisation, where the domain algebra is a Boolean algebra, the corresponding conjugation relation between forward and backward diamonds is equivalent to a Galois connection that uses dual modal box operators [21]. Here, of course, in the absence of Boolean complementation, this equivalence fails. Alternatively, however, modal box operators could be axiomatised independently and then related to the diamonds. But this further exploration of modal algebras over distributive lattices based on domain semirings is left for future work.

8. Boolean domain semirings

The domain algebras of the external axiomatisation are Boolean algebras by definition. This raises the question of adapting the internal axioms such that they induce a Boolean domain algebra, too. Obviously, the domain algebras considered so far were only distributive lattices, because the concept of domain alone does not lead to a notion of complementation on the domain algebra. In order to achieve that, a notion of antidomain can be introduced.

Example 8.1. On relation semirings, the complement of the domain of a relation R is the set that can be defined by the map $a : 2^{A\times A} \to 2^{A\times A}$ defined by

$$a(R) = \{(a, a) \in A \times A : \text{ there is no } b \in A \text{ such that } (a, b) \in R\}.$$

It models the antidomain of R, and a can therefore be called an *antidomain operation*. It follows from this definition that $(a, a) \in a(R) \Leftrightarrow (a, a) \notin d(R)$, that $d(R) \cup a(R) = 1_A$ and that $d(R) \circ a(R) = \emptyset$. Hence domain elements and antidomain elements are indeed complements. Moreover,

$$(a, a) \in a(a(R)) \Leftrightarrow \neg\exists b.(a, b) \in a(R) \Leftrightarrow (a, a) \notin a(R) \Leftrightarrow (a, a) \in d(R),$$

and therefore $a(a(R)) = d(R)$. This definability of domain in terms of antidomain suggests that antidomain is a more fundamental operation than domain. One of the most important properties of antidomain elements is that they are *left annihilators*, $a(R) \circ R = \emptyset$, since elements in $a(R)$ are not related to anything by R, and in fact they are even *greatest* left annihilators; that is, for all $P \subseteq 1_A$,

$$P \circ R = \emptyset \Leftrightarrow P \subseteq a(R).$$

Obviously, $[R]P = a(R \circ a(P))$ yields the link with modal algebras and Kripke frames.

Example 8.2. For a set X of traces on some trace semiring, the antidomain operation is defined as

$$a(X) = \{p \in \Sigma_P : p\sigma \notin X \text{ for all sequences } \sigma\}.$$

Again, $d(X) \cup a(X) = \Sigma_P$, $d(X)a(X) = \emptyset$, $a(a(X)) = d(X)$ and, for all $P \subseteq \Sigma_P$,

$$PX = \emptyset \Leftrightarrow P \subseteq a(X).$$

Example 8.3. In the labelled transition system of Example 3.3, the antidomain of action a is the set $\{q, r\}$ and the antidomain of action b is the empty set. Hence the antidomain of an action yields the set of all those states where the action is not enabled.

The obvious approach for obtaining a Boolean domain algebra would therefore be to extend a domain semiring by an antidomain operation and stipulate that domain and antidomain elements are complemented. However, it should be possible to replace the domain operation entirely by the antidomain operation. An analysis with Prover9 or Waldmeister and Mace4 was substantial in converging to the following axiomatisation, which is very compact, but still natural.

A *Boolean domain semiring* is a semiring extended by an *antidomain operation* $a : S \to S$ which, for all $x, y \in S$, satisfies the axioms

$$a(x)x = 0, \tag{BD1}$$

$$a(xy) + a(xa^2(y)) = a(xa^2(y)), \tag{BD2}$$

$$a^2(x) + a(x) = 1. \tag{BD3}$$

The first axiom is called the *annihilation* axiom, the second one the *locality* axiom and the third one the *excluded middle* axiom. Setting $d(x) = a^2(x)$, we could show the following facts with Waldmeister and Prover9. Adapting the templates in the Appendices by replacing the domain axioms with the antidomain axioms is straightforward. In fact, Theorem 8.7 justifies this replacement, since its proof implies showing that the map a^2 satisfies the domain axioms.

Lemma 8.4. *Let S be a Boolean domain semiring. Then, for all $x \in S$,*

(i) $x + x = x$, *and hence S is a dioid;*
(ii) $d(a(x)) = a(x)$, *and hence antidomain elements are domain elements (the equation also conveys that a triple negation is the same as a single negation);*
(iii) $a(x)d(x) = 0$, *and hence $a(x)$ is the Boolean complement of $d(x)$.*

The next theorem then follows from the facts of Section 6.

Theorem 8.5. *The domain algebra of a Boolean domain semiring is the maximal Boolean subalgebra of the semiring of subidentities.*

Proof. By Theorem 6.12 and Lemma 8.4, the maximal Boolean subalgebra of the subalgebra of subidentities and the domain algebra of a Boolean domain semiring must be the same. □

Given, the compactness of the axioms, the following fact is not too surprising.

Lemma 8.6. *The axioms of Boolean domain semirings are irredundant.*

Proof. The irredundancy of (BD1) can be verified on the Boolean semiring with the following antidomain function for $x = 1$.

a	0	1
	1	1

$+$	0	1
0	0	1
1	1	1

$\cdot$	0	1
0	0	0
1	0	1

The irredundancy of (BD2) can be verified on the following 3-element semiring for $x = y = 2$.

a	0	1	2
	1	0	0

$+$	0	1	2
0	0	1	2
1	1	1	1
2	2	1	2

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	0

The irredundancy of (BD3) can again be verified again on the Boolean semiring with the antidomain function for $x = 0$.

a	0	1
	0	0

$+$	0	1
0	0	1
1	1	1

$\cdot$	0	1
0	0	0
1	0	1

□

We now compare Boolean domain semirings with domain semirings extended by an antidomain operation that satisfies the obvious axioms of Boolean complementation.

Theorem 8.7. *A semiring S is a Boolean domain semiring if and only if it can be extended by a domain operation and an operation $a : S \to S$ that satisfies $d(x) = a^2(x)$,*

$$d(x) + a(x) = 1, \quad \text{and} \quad d(x)a(x) = 0.$$

Again, the entire proof can be automated, and an alternative equational proof can be found in the preliminary version of this paper [7].

We now verify that the internal axiomatisation of Boolean domain semirings is precisely as expressive as the external one. Thus all properties that have previously been derived can be reused and the entire machinery of external Boolean modal semirings can be translated to the internal setting.

Theorem 8.8. *Every theorem of external Boolean domain semirings can be translated into a theorem of internal Boolean domain semirings and vice versa.*

Proof. Let S be an internal Boolean domain semiring. Then $d(S)$ is a legitimate test algebra for the external axiomatisation (in particular, it is the maximal Boolean subalgebra of the algebra of subidentities by Proposition 6.12). Also, d is by definition of type $S \to d(S)$. We have already verified that the external domain axioms follow from the internal ones via Lemma 5.1 and Theorem 8.7. In particular, $d(px) = pd(p) \leq p$ by Lemma 5.1(viii) and (D3), which is a law of Boolean domain semirings.

Let now S be an external Boolean domain semiring and define, for each $x \in S$, $a(x) = \neg d(x)$ on B, where $\neg d(x)$ is the complement of $d(x)$ in B. It has already been shown that (BD1) holds [5]. (BD2) holds by axiom (3). (BD3) holds by definition. □

Theorem 8.8 also provides some important intuition about the antidomain operation. This is motivated by the fact that, in relation and trace semirings, the antidomain $a(x)$ of an element x, which is the complement of the domain of x, is a greatest left annihilator. The following fact is immediate from Theorem 8.8 and results from [5].

Lemma 8.9. *Let S be a Boolean domain semiring and let $x \in S$. Then $a(x)$ is a greatest left annihilator in the set of subidentities. For all $y \in S$ with $y \leq 1$,*

$$yx = 0 \Leftrightarrow y \leq a(x).$$

It is also straightforward to show with Mace4 that the assumption $y \leq 1$ cannot be relaxed. The relationship between least left preservation and greatest left annihilation is slightly asymmetrical. Least left preservers must be taken from the domain algebra – even in the case of Boolean domain semirings – whereas greatest left annihilators can be chosen from the larger subsemiring of subidentities. This might be another indication that antidomain is more fundamental than domain.

The internal axiomatisation of Boolean domain semirings is certainly simpler and conceptually more appealing than the external one. Most significantly, the previous need to explicitly specify and embed the Boolean algebra led to far more axioms which created a significant overhead for ATP systems. A main contribution of this paper is the shift of focus from domain to antidomain and the realisation that the latter is axiomatically more fundamental than the former. The concept of antidomain is also immediately relevant to the verification of action systems and similar models of parallel computation [6].

9. Antidomain

In the first part of this section we study the antidomain operation in isolation. Natural axioms can be obtained by implicitly complementing the domain axiom (D1)–(D5). We can then extend a semiring S by a map $a : S \to S$ that satisfies the identities

$$a(x)x = 0, \tag{4}$$
$$a(xy) = a(xd(y)), \tag{5}$$
$$a(x) \leq 1, \tag{6}$$
$$a(0) = 1, \tag{7}$$
$$a(x + y) = a(x)a(y). \tag{8}$$

We call these axioms the *antidomain axioms* and the map a defined by these axioms an *antidomain operation*. As before, semirings extended by an antidomain operation are automatically idempotent. The following lemma provides a sanity check for the antidomain axioms. They could easily be proved with Waldmeister.

Lemma 9.1. *The antidomain axioms hold in every Boolean domain semiring.*

The next statement shows that fixpoints of the antidomain operation are rather undesirable.

Lemma 9.2. *A semiring is trivial if it can be extended by an antidomain operation that has a fixpoint.*

Prover9 easily showed that, if $a(x) = x$ holds for some $x \in S$, then $x = y$ for all $x, y \in S$. More interesting are the fixpoints of a^2.

Proposition 9.3. *The antidomain elements of a semiring with an antidomain operation a are the fixpoints of a^2.*

Proof. The map a^2 is a projection, since $a^4(x) = a(a(1a^2(x))) = a(a(1x)) = a^2(x)$, and the result follows from Proposition 4.2. □

This fixpoint lemma for antidomain is again a powerful tool for analysing the structure of antidomain elements, but here with rather negative results.

Lemma 9.4. *The antidomain elements of a semiring with an antidomain operation are closed under multiplication, but need not be closed under addition.*

Proof. Closure under multiplication, that is, $a^2(a(x)a(y)) = a(x)a(y)$, has been shown by Waldmeister and Prover9. For closure under addition, we must consider $a^2(a(x) + a(y)) \neq a(x) + a(y)$. According to Mace4, there is a five-element model that verifies this fact. Thus there are two antidomain elements for which the sum is not again an antidomain element. □

However, the antidomain axioms capture one of the most natural properties of antidomain.

Lemma 9.5. *Let S be a semiring with an antidomain operation a. Then $a(x)$ is the greatest left annihilator of $x \in S$ on $a(S)$.*

There is a six-element model that shows that annihilators cannot be taken from the whole set of subidentities.

The situation can further be explained as follows. If p and q are annihilators of x, then $p + q$ is another annihilator of x, since $(p + q)x = px + qx = 0$. Furthermore, $a(x)a(y)$, which is equal to $a(x + y)$, is the greatest annihilator of $x + y$, and thus $a(x) + a(y)$ is generally not an annihilator of $x + y$. Is $a(x) + a(y)$ an annihilator of xy? Mace4 presents an eight-element counterexample to $(a(x) + a(y))xy = 0$. Note that also $a(a(x)a(y)) = a(a(x)) + a(a(y))$, the dual of the instance $a(a(x) + a(y)) = a(a(x))a(a(y))$ of (8), does not hold (by a counterexample of size five).

These results show that the axioms (4)–(8) are too weak to be convincing. They do not yield a satisfactory antidomain algebra and give no proper notion of state space. Interestingly, the situation is similar to that in Heyting algebras, where only one of the De Morgan laws holds, and where the law of excluded middle fails. In semirings with an antidomain operation, $a^2(x) + a(x)$ need not be equal to 1 and $a^2(x)x$ need not be equal to x, and hence preservation fails, as Mace4 can show. However, strengthening the axioms in meaningful ways seems to lead almost inevitably to Boolean domain semirings. Adding only $a^2(x)x = x$ to the antidomain axioms is not sufficient to enforce closure of antidomain elements under addition (there is a five-element counterexample).

In the next step, we add the antidomain axioms to the axioms of domain semirings. The resulting structures are still weaker than Boolean domain semirings.

Lemma 9.6. *Let S be a domain semiring with an antidomain operation.*

(i) $d(a(x)) = a(x)$ *need not hold for all* $x \in S$.
(ii) $d(a(x)) = a(x)$, *for* $x \in S$, *need not imply* (BD3).

(BD3) can, in fact, be proved if the double negation $d(x) = a^2(x)$ (which is also not implied), is added to the assumptions in (ii). In that case, the resulting structures are precisely the Boolean domain semirings, but the axiomatisation is much less compact.

Adding the antidomain axioms to those of domain semirings suffices to prove some interesting properties which then hold a fortiori in Boolean domain semirings. All of them have been verified by Waldmeister and Prover9.

Lemma 9.7. *Let S be a domain semiring. For all $x, y \in S$, the following properties follow from the antidomain axioms.*

(i) $a(x) \leq a(xy)$.
(ii) $a(x) = a(d(x))$.
(iii) $a(x)d(x) = 0$.
(iv) $a(x) \leq d(a(x))$.
(v) $a(x)a(x) = a(x)$.
(vi) $a(x)a(y) = a(y)a(x)$.
(vii) $a(x) + a(y) \leq a(d(x)y)$.
(viii) $a(1) = 0$.
(ix) $d(x) = a^2(x) \Rightarrow d(x)a(x) = 0$,
(x) $x \leq y \Rightarrow a(y) \leq a(x)$ *(antidomain is antitone).*

It can further be shown that, in this setting, antidomain elements still need not be the Boolean complements of domain elements. Therefore, the interplay between domain and antidomain operation axiomatised by (4)–(8) might be interesting in its own right and deserves further investigation.

Finally, we analyse the relationship between domain semirings with antidomain defined via greatest left annihilation and Boolean domain semirings.

Lemma 9.8. *Some domain semiring with an antidomain operation satisfies the greatest left annihilator law*

$$y \leq 1 \Rightarrow (yx = 0 \Leftrightarrow y \leq a(x))$$

and $d(a(x)) = a(x)$, *but not* $d(x) = a^2(x)$.

This can easily be shown by Mace4.

Lemma 9.9. *Every domain semiring that satisfies the greatest left annihilator law, $d(a(x)) = a(x)$ and $d(x) = a^2(x)$ is a Boolean domain semiring.*

However, the resulting alternative axiomatisation of Boolean domain semirings is not equational, and it is less compact than our standard one.

10. Heyting domain semirings

This section further demonstrates the flexibility of the internal axiomatisation of domain semirings by considering domain algebras that are Heyting algebras. This can easily be enforced by adding the Galois connection

$$pq \leq r \Leftrightarrow p \leq q \to r, \tag{9}$$

where $p, q, r \in d(S)$, to the axioms of a domain semiring S. As usual, $q \to r$ is called the *relative pseudocomplement* of r with respect to q, and the *pseudocomplement* $\neg p$ of $p \in d(S)$ is defined as $p \to 0$. Every lattice extended by an operation of relative pseudocomplementation such that all elements satisfy (9) is a Heyting algebra by definition. Here, the arrow $\to$ is a partial function which is only defined on $d(S)$.

Now, since all finite distributive lattices are Heyting algebras, there is no finite domain algebra that violates the closure condition $d(p \to q) = p \to q$ for $p, q \in d(S)$. However, for the infinite case and for completeness we explicitly add this condition, since otherwise $p \to q$ need not be a domain element (we want the domain elements to constitute the Heyting algebra).

A *Heyting domain semiring* is a domain semiring S extended by an endomorphism $\to$ on domain elements that satisfies the Galois connection (9) and the closure condition $d(p \to q) = p \to q$ for all $p, q \in S$.

Unfortunately, the partiality of the arrow makes the encoding in Mace4 rather delicate, since the tool requires that functions be total. Waldmeister, in contrast, supports sorts.

The following fact is an immediate consequence of standard properties of relative pseudocomplements [14].

Proposition 10.1. *A domain semiring S is a Heyting domain semiring if and only if all $p, q, r \in d(S)$ satisfy the equations*

$$p \to p = 1, \tag{HD1}$$
$$p(p \to q) = pq, \tag{HD2}$$
$$q(p \to q) = q, \tag{HD3}$$
$$p \to qr = (p \to q)(p \to r), \tag{HD4}$$
$$d(p \to q) = p \to q. \tag{HD5}$$

Again, by the results of Section 7, Heyting domain semirings give rise to modal semirings and modal Kleene algebras defined as Heyting algebras with operators; intuitionistic variants of dynamic logics and temporal logics can easily be defined in this setting. This is, however, beyond the scope of this paper.

Alternatively to the development of this section, one might try to replace $q \to r$, which can be expanded to $d(x) \to d(y)$, by $a(x) + d(y)$ in the Galois connection (9) or in the identities axiomatising relative pseudocomplementation in Proposition 10.1. This would replace the arrow again by an antidomain function that is implicitly defined through these laws.

So first, let us add, for all elements x, y and z of a domain semiring S, the Galois connection

$$d(x)d(y) \leq d(z) \Leftrightarrow d(x) \leq a(y) + d(z) \tag{10}$$

to the domain semiring axioms. We also need to add the closure condition $d(a(x)) = a(x)$, since otherwise, by Mace4, antidomain elements need not be domain elements. However, these axioms are already too strong.

Proposition 10.2. *Every domain semiring that satisfies (10) and $d(a(x)) = a(x)$ is a Boolean domain semiring.*

This has been proved by Prover9; hence the approach based on (10) is no real alternative.

Second, therefore, we replace the arrows by addition and antidomain in the identities of Proposition 10.1. But again, these axioms are too strong.

Lemma 10.3. *Let S be a domain semiring extended by an antidomain function that, for all $x, y, z \in S$, satisfies the identities*

$$a(x) + d(x) = 1,$$
$$d(x)(a(x) + d(y)) = d(x)d(y),$$
$$d(y)(a(x) + d(y)) = d(y),$$
$$a(x) + d(y)d(z) = (a(x) + d(y))(a(x) + d(z)).$$

Then S is a Boolean domain semiring and $d(a(x)) = a(x)$.

This has been verified by Prover9.

All results of this section can easily be dualised into facts that hold in co-Heyting algebras.

11. Application: termination analysis

One of the most fundamental program analysis tasks is reasoning about termination and non-termination. Various supporting techniques have already been developed and applied in the context of the external approach. In particular, both the relational notion of wellfoundedness and the modal notion of termination expressed by Löb's formula have been studied [4].

This section presents the first fully automated correspondence proof for Löb's formula. Such modal correspondence theorems generally associate the validity of modal formulas with (relational) properties that hold on Kripke frames. Löb's formula, in particular, corresponds to the wellfoundedness of transitive Kripke frames. Our automation result establishes Löb's formula and related expressions as automatically verified laws that can safely be used for termination analysis. Previous attempts to automate a manual proof based on the external axiomatisation [4] did not succeed [11].

It should come as no surprise that a statement of comparable complexity cannot be proved in one full sweep from the set of axioms. We therefore use *hypothesis learning* [12]. We start with a basis of axioms from which some potentially dangerous axioms have been discarded. Axioms like $x+y = y+x$ or $1+xx^* = x^*$, for instance, easily make the search space explode and distract the prover. Also, further potentially useful hypotheses often need to be given to the prover as lemmas. For reasons of consistency, all additional hypotheses should have previously been verified. By Theorem 8.8, we can freely use the large database of automatically verified theorems for the external axiomatisation [23]. At the beginning of the hypothesis learning phase, the hypotheses given are often too weak to entail the goal. Then, Mace4 can often find a counterexample indicating that the hypotheses need further strengthening. So more axioms and lemmas need to be added until Mace4 does not find a counterexample within reasonable time. Then Prover9 can be called. If a proof fails within reasonable time limits, another combination can be tried. This procedure is currently being implemented, but for the result of this section, the hypotheses have still manually been learned. This is much simpler if, as in the present case, a manual proof is already known.

In its usual form, Löb's formula is written as $\Box(\Box p \to p) \to p$. To represent it algebraically, let K be a modal Kleene algebra and let $d(K)$ be the Boolean domain algebra of K. We first replace $\Box$ by $[x]$ and then dualise it to forward diamonds via $[x]p = (\neg\langle x\rangle\neg p)$, where $\neg$ now denotes Boolean complementation. Writing $p - q = p \cdot \neg q$, Löb's formula can then be algebraised to the identity $\langle x\rangle p \leq \langle x\rangle(p - \langle x\rangle p)$.

In order to reason more concisely, we define $\Omega_x(p) = p - \langle x\rangle p$. In relation Kleene algebras, it denotes that subset of p from which no further x-actions are possible: that is, the *final elements* of p with respect to x. Therefore, we say that an element x of K is *Löbian* [4] if

$$\langle x\rangle p \leq \langle x\rangle \Omega_x(p) \tag{11}$$

holds for all $p \in d(K)$. Intuitively, x is Löbian if all x-actions lead into sets of x-maximal elements.

To set up the correspondence result, we also need to express wellfoundedness and transitivity. We say that x is *wellfounded* [4] if

$$\Omega_x(p) = 0 \Rightarrow p = 0 \tag{12}$$

holds for all $p \in d(K)$. Hence only the empty set has no x-maximal elements. Finally, x is *transitive* if $xx \leq x$ and *diamond transitive* (d-transitive) if $\langle x\rangle\langle x\rangle p \leq \langle x\rangle p$ holds for all $p \in d(K)$. Obviously, each transitive element is d-transitive, but the converse need not hold. Mace4 could find a four-element counterexample.

It has previously been shown (in the external case) that the following two facts hold on a modal Kleene algebra.

Theorem 11.1 (*[4]*)**.** *Every Löbian element is wellfounded.*

With the internal axioms, Prover9 needed just a few seconds. The next theorem establishes the converse direction, and hence the correspondence result. It is the main statement in this section.

Theorem 11.2. *Every wellfounded d-transitive element is Löbian.*

To prove this theorem, the intermediate property $\langle x\rangle p \leq \langle x^+\rangle \Omega_x(p)$, where $x^+ = xx^*$, has previously been introduced [4]. Here, we base the proof on a simpler property. We say that an element x is *pre-Löbian* if

$$p \leq \langle x^*\rangle \Omega_x(p). \tag{13}$$

Intuitively, if x is pre-Löbian, then every finite iteration of x will lead to x-maximal elements. It is obvious, but not important for the proof, that every pre-Löbian element x satisfies $\langle x\rangle p \leq \langle x^+\rangle \Omega_x(p)$ for all domain elements p.

We now compare pre-Löbian elements and wellfounded elements.

Proposition 11.3. *An element x is wellfounded if and only if it is pre-Löbian.*

Proof. Let x be pre-Löbian and assume that $\Omega_x(p) = 0$. Then

$$p \leq \langle x^*\rangle \Omega_x(p) = \langle x^*\rangle 0 = 0.$$

Let now x be wellfounded. Then x is pre-Löbian if $p - \langle x^*\rangle \Omega_x(p) = 0$, whence, by wellfoundedness, it suffices to show that

$$p - \langle x^*\rangle \Omega_x(p) \leq \langle x\rangle(p - \langle x^*\rangle \Omega_x(p)),$$

since $p - q = 0 \Leftrightarrow p \leq q$. In the following argument, we use the fact that

$$p - \Omega_x(p) = p - (p - \langle x\rangle p) = p\langle x\rangle p \leq \langle x\rangle p. \tag{14}$$

We calculate

$$\begin{aligned} p - \langle x^*\rangle \Omega_x(p) &= p - (\Omega_x(p) + \langle x^+\rangle \Omega_x(p)) \\ &= (p - \Omega_x(p)) - \langle x^+\rangle \Omega_x(p) \\ &\leq \langle x\rangle p - \langle x^+\rangle \Omega_x(p) \\ &\leq \langle x\rangle(p - \langle x^*\rangle \Omega_x(p)). \end{aligned}$$

The first step uses $x^* = 1 + x^+$ and $\langle 1\rangle p = p$. The second step uses the fact $p - (q + r) = (p - q) - r$ from Boolean algebra. The third step uses (14). The fourth step uses $f(p) - f(q) \leq f(p - q)$, which holds for all additive functions on a Boolean algebra, and $x^+ = xx^*$. □

Prover9 could show in a few seconds that pre-Löbian elements are wellfounded; it could show in less than 20 s that wellfounded elements are pre-Löbian. This last proof requires a substantial amount of hypothesis learning, which of course makes the running times given less significant. The input file that documents the hypotheses used can be found in Appendix E.

We can now prove the main theorem of this section, Theorem 11.2.

Proof (*Of Theorem* 11.2)**.** Let x be wellfounded. Then it is pre-Löbian by Proposition 11.3 and satisfies $\langle x\rangle p \leq \langle x^+\rangle \Omega_x(p) = \langle x\rangle \Omega_x(p)$, since $\langle x\rangle$ is isotone and x is d-transitive. Whence x is Löbian. □

For automating this proof, Prover9 could show almost instantaneously that $\langle x^+\rangle p \leq \langle x\rangle p$ follows from d-transitivity, when given the additional hypothesis $p + \langle x\rangle q \leq q \Rightarrow \langle x^*\rangle p \leq q$, which holds in modal Kleene algebras by Proposition 7.3. The converse direction, $\langle x\rangle p \leq \langle x^+\rangle p$, could be proved in less than one second, assuming only isotonicity of domain as an additional hypothesis. The resulting identity is needed in the second step of the proof. Theorem 11.2 could then been proved automatically in about one second, using the fact that wellfounded elements are pre-Löbian, d-transitive elements satisfy $\langle x^+\rangle p = \langle x\rangle p$ and the isotonicity of domain. To decrease running times, the Kleene star axioms were also discarded. Proofs without hypothesis learning could possibly have been obtained with more patience.

12. Conclusion

We introduced new axioms for domain semirings and Kleene algebras with domain. The resulting internal approach is more general, simpler, more flexible and better suited for automated reasoning than a previous external approach. It is more general because the domain operation is one-sorted and the previous intricate interdependency between axioms and typing constraints is avoided. It is simpler because it drastically reduces the number and complexity of axioms needed. It is more flexible because it offers freedom of choice for domain algebras: different kinds of domain algebra can be obtained by different adaptations of the basic set of axioms. Its suitability for automated deduction has been demonstrated through a nontrivial example from modal correspondence theory and many smaller proofs in the paper. In particular, the automated analysis of the fine structure of domain algebras would have been much more difficult in the external setting. Beyond the results of this paper it has recently been shown that the approach is stable with respect to variations. The axiomatisations in this paper can be adapted to a family of near-semirings which covers the case of demonic refinement algebras, probabilistic Kleene algebras, game algebras and process algebras, where some of the semiring axioms are no longer valid [6]. It also simplifies the study of free domain semirings [13], and non-trivial verification tasks can be automated in this setting [24]. A significant number of proof tasks from this paper have also been incorporated in the TPTP library [26], which is the standard library for automated theorem proving benchmarks and which supplies problems for the main ATP system competition [25].

A significant contribution of this paper is that most of our results were achieved in an automated game of conjectures and refutations with automated theorem provers and counterexample generators. This new style of computer-enhanced mathematics allowed us to drastically speed up our analysis. The automated proof of Theorem 8.7, for instance, was an afternoon session with Prover9 and Mace4 that required little ability beyond typing in axioms and conjectures, running the tools and waiting. Finding proofs by equational reasoning for the preliminary version of this paper [7], in contrast, took us a couple of days. Moreover, it allowed us to condense the presentation and dispense with routine technical arguments while even gaining in trustworthiness and reproducibility. A current drawback of ATP systems is their proof output. Most systems present resolution and paramodulation proofs that do not reflect the usual calculational style of reasoning used by humans. Waldmeister can display equational proofs, but these are usually poorly structured and at the wrong level of granularity. Extracting readable or even publishable equational proofs from these outputs is often tedious and frustrating. Automated translations would be an important contribution, and they are certainly feasible.

This paper focuses on the mathematical and conceptual foundations of domain semirings and Kleene algebras with domain. Further investigations and applications can be based on these results.

A main area of future research are certainly the modal algebras over distributive lattices and (co-)Heyting algebras that arise from our axiomatisations. Distributive lattices and Heyting algebras with operators are relevant to many-valued logics and description logics. A comprehensive survey can be found in an article by Sofronie-Stokkermans [22]. Intuitionistic dynamic logics, which are based on Heyting algebras, have been developed by Degen and Werner [3]. Intuitionistic variants of the linear temporal logic LTL with applications in program verification, in particular for assume-guarantee reasoning and for properties that relate finite and infinite behaviour, have been studied by Maier [19]. But both areas seem to be rather underexplored. In program analysis, domain elements can model tests in control structures such as conditionals and loops. The semiring semantics of the conditional $\mathsf{if}\ p\ \mathsf{then}\ x\ \mathsf{else}\ y$, for instance, is $px + \neg py$. On Boolean domain algebras, tests can only evaluate to true or to false. In models with finer granularity, tests can also diverge or abort, which violates tertium non datur. Heyting algebras and similar lattices provide the appropriate semantics for these behaviours.

In program analysis and verification, our results provide a uniform framework from which a variety of domain-specific applications can be explored. A particular challenge lies in transforming the combination of computational algebras with ATP technology into lightweight formal methods with heavyweight automation.

Finally, our results motivate a number of concrete research questions which might contribute to a better understanding of domain semirings and Kleene algebras with domain.

1. Proposition 5.3 shows that some semirings cannot be extended to domain semirings. Which conditions guarantee the existence of a domain operation with various domain algebras?
2. When domain algebras are not unique, does the set of all domain algebras carry any algebraic structure?
3. How can the distributive lattices and (co-)Heyting algebras with domain be extended into coherent modal algebras, including modal box operators?
4. Which axioms lead to antidomain semirings with well-defined antidomain algebras?

5. Can (co-)Heyting complementation on the domain algebra of a domain semiring be axiomatised by a total function?
6. Are there further interesting domain semirings in the spectrum of domain algebras between distributive lattices and Boolean algebras?
7. What are the free algebras for various classes of domain (near-)semirings?

Acknowledgements

We are most grateful to Peter Jipsen for interesting discussions, to Viorica Sofronie-Stokkermans for important references, and to the referees for their comments.

Appendix A. Axioms for Prover9/Mace4

This appendix contains a template for proving theorems with the ATP system Prover9 and the counterexample generator Mace4. Hypotheses for these tools must be given in the `sos`-environment, the goal must be written into the `goals`-environment. Prover9 implicitly negates the goal, performs a series of transformations on the input and then tries to derive an inconsistency $F from the hypotheses and the negated goal. Mace4 tries to construct a finite model in which the hypotheses are true and the goal is false.

The templates can easily be adapted to Kleene algebras, Boolean domain semirings and other structures. Further examples can be found at our web site [23], and comprehensive documentation can be found at the tool web site [20].

```
op(500, infix, "+"). % addition
op(490, infix, ";"). % multiplication

formulas(sos).
% semiring axioms
    x+y=y+x.          % additive monoid
    x+0=x.
    x+(y+z)=(x+y)+z.
    x;1=x.            % multiplicative monoid
    1;x=x.
    x;(y;z)=(x;y);z.
    0;x=0.            % annihilation
    x;0=0.
    x;(y+z)=x;y+x;z.  % distributivity
    (x+y);z=x;z+y;z.
   %x+x=x.            % idempotency

% domain axioms
    x+d(x);x=d(x);x.
    d(x;y)=d(x;d(y)).
    d(x)+1=1.
    d(0)=0.
    d(x+y)=d(x)+d(y).

end_of_list.

formulas(goals).

    %add lemma to be proved/refuted

end_of_list.
```

Appendix B. Axioms for Waldmeister

Waldmeister is a highly efficient theorem prover for pure equations, which we predominantly used for the equational proofs in this paper. Further examples for proofs with Waldmeister can be found at our web site [23] and the tool web site [10].

```
NAME            dsemiring

MODE            PROOF

SORTS           S

SIGNATURE           +: S S -> S
                    0:     -> S
                    *: S S -> S
                    1:     -> S
```

```
                    d:   S -> S

                  c1,c2,sk:     -> S

ORDERING         KBO
                 +=1, 0=1, *=1, 1=1, c1=1, c2=1, sk=1, d=1
                 * > + > d > 0 > 1 > c1 > c2 > sk

VARIABLES        x,y,z: S

EQUATIONS      % 1. semiring axioms

                     +(+(x,y),z) = +(x,+(y,z))
                          +(x,y) = +(y,x)
                          +(x,0) = x


                     *(*(x,y),z) = *(x,*(y,z))
                          *(x,0) = 0
                          *(0,x) = 0
                          *(x,1) = x
                          *(1,x) = x

                     *(x,+(y,z)) = +(*(x,y),*(x,z))
                     *(+(x,y),z) = +(*(x,z),*(y,z))

               % 2. domain axioms

                   +(x,*(d(x),x)) = *(d(x),x)
                        d(*(x,y)) = d(*(x,d(y)))
                        +(d(x),1) = 1
                             d(0) = 0
                        d(+(x,y)) = +(d(x),d(y))

CONCLUSION   % Equations are implicitly existentially quantified.
             % Universally quantified identities must be skolemised.

                      % add lemma to be proved
```

Appendix C. Proof of Proposition 4.1 by Waldmeister

The following axioms imply the following theorem:

Axiom 1: $+(x1, x2) = +(x2, x1)$
Axiom 2: $*(x1, 1) = x1$
Axiom 3: $+(x1, *(d(x1), x1)) = *(d(x1), x1)$
Axiom 4: $+(d(x1), 1) = 1$
Theorem 1: $+(1, 1) = 1$.

Proof. Lemma 1: $d(1) = +(1, d(1))$

$d(1)$
= by Axiom 2 RL
$*(d(1), 1)$
= by Axiom 3 RL
$+(1, *(d(1), 1))$
= by Axiom 2 LR
$+(1, d(1))$

Lemma 2: $+(1, d(x1)) = 1$

$+(1, d(x1))$
= by Axiom 1 RL
$+(d(x1), 1)$
= by Axiom 4 LR
1

Lemma 3: $1 = +(1, 1)$

$$\begin{aligned} & 1 \\ = \quad & \text{by Lemma 2 RL} \\ & +(1, d(1)) \\ = \quad & \text{by Lemma 1 LR} \\ & +(1, +(1, d(1))) \\ = \quad & \text{by Lemma 2 LR} \\ & +(1, 1) \end{aligned}$$

Theorem 1: $+(1, 1) = 1$

$$\begin{aligned} & +(1, 1) \\ = \quad & \text{by Lemma 3 RL} \\ & 1 \end{aligned}$$

These lemmas encode the direct equational proof

$$1 + 1 = 1 + 1 + d(1) = 1 + 1 + d(1)1 = 1 + d(1)1 = 1 + d(1) = 1.$$

Unfortunately, Waldmeister's decomposition of more complex theorems with longer proofs into lemmas often does not have the right granularity for humans, and even the reconstruction of readable proofs can be extremely tedious. Presenting such proofs is therefore much less helpful than providing the templates by which a proof can be reproduced. Prover9 outputs are resolution proofs and therefore even less readable, as the examples on our web site [23] show. □

Appendix D. Proof of Proposition 5.1(xi) by Prover9

The goal of the template from Appendix A must now be instantiated as follows:

```
formulas(goals).

  d(y)=y -> (x<=d(y);x <-> d(x)<=d(y)).

end_of_list.
```

Prover9 then produced the following output:

```
% Proof 1 at 10.35 (+ 0.45) seconds.
% Length of proof is 40.
% Level of proof is 8.
% Maximum clause weight is 13.
% Given clauses 1339.

1 x <= y <-> x + y = y # label(non_clause).  [assumption].
2 d(y) = y -> (x <= d(y) ; x <-> d(x) <= d(y)) # label(non_clause) # label(goal).  [goal].
5 x + y = y + x.  [assumption].
9 x ; 1 = x.  [assumption].
10 1 ; x = x.  [assumption].
13 x ; (y + z) = x ; y + x ; z.  [assumption].
14 x ; y + x ; z = x ; (y + z).  [copy(13),flip(a)].
15 (x + y) ; z = x ; z + y ; z.  [assumption].
16 x ; y + z ; y = (x + z) ; y.  [copy(15),flip(a)].
17 -(x <= y) | x + y = y.  [clausify(1)].
18 x <= y | x + y != y.  [clausify(1)].
19 d(x) ; x = x.  [assumption].
20 d(x ; y) = d(x ; d(y)).  [assumption].
21 d(x ; d(y)) = d(x ; y).  [copy(20),flip(a)].
22 d(x) + 1 = 1.  [assumption].
23 1 + d(x) = 1.  [copy(22),rewrite([5(3)])].
25 d(x + y) = d(x) + d(y).  [assumption].
26 d(c1) = c1.  [deny(2)].
27 c2 <= d(c1) ; c2 | d(c2) <= d(c1).  [deny(2)].
28 c2 <= c1 ; c2 | d(c2) <= c1.  [copy(27),rewrite([26(3),26(9)])].
29 -(c2 <= d(c1) ; c2) | -(d(c2) <= d(c1)).  [deny(2)].
30 -(c2 <= c1 ; c2) | -(d(c2) <= c1).  [copy(29),rewrite([26(3),26(9)])].
36 x ; (1 + y) = x + x ; y.  [para(9(a,1),14(a,1,1)),flip(a)].
42 (x + 1) ; y = y + x ; y.  [para(10(a,1),16(a,1,2)),rewrite([5(2)]),flip(a)].
46 x <= y | y + x != y.  [para(5(a,1),18(b,1))].
51 d(1) = 1.  [para(19(a,1),9(a,1)),flip(a)].
55 (x + d(y)) ; y = y + x ; y.  [para(19(a,1),16(a,1,2)),rewrite([5(2)]),flip(a)].
66 1 + c1 = 1.  [para(26(a,1),23(a,1,2))].
67 c2 <= c1 ; c2 | c1 + d(c2) = c1.  [resolve(28,b,17,a),rewrite([5(9)])].
```

```
72 1 + 1 = 1.  [para(51(a,1),23(a,1,2))].
89 x + c1 ; x = x.  [para(66(a,1),16(a,2,1)),rewrite([10(2),10(5)])].
96 x + x = x.  [para(72(a,1),14(a,2,2)),rewrite([9(2),9(2),9(3)])].
109 d(x) + d(x ; y) = d(x).  [para(36(a,1),21(a,2,1)),rewrite([25(3),51(2),23(3),9(2),25(4)]),flip(a)].
113 x <= x.  [resolve(96,a,18,b)].
681 (x + d(y)) ; y = (x + 1) ; y.  [para(55(a,2),42(a,2)),flip(a)].
898 c1 + d(c2) = c1 | c1 ; c2 = c2.  [resolve(67,a,17,a),rewrite([89(11)]),flip(b)].
32234 c1 ; c2 = c2.  [para(898(a,1),681(a,1,1)),rewrite([5(11),66(11),10(11)]),merge(b)].
32292 -(d(c2) <= c1).  [back_rewrite(30),rewrite([32234(4)]),unit_del(a,113)].
32293 c1 + d(c2) != c1.  [ur(46,a,32292,a)].
32316 $F.  [para(32234(a,1),109(a,1,2,1)),rewrite([26(2),26(6)]),unit_del(a,32293)].
```

It should be evident from this example that Prover9 outputs are not intended to be readable for humans. Again, presenting a proof is therefore less helpful than providing the templates for reproducing it.

Appendix E. Proof of Proposition 11.3 by Prover9

The following input file shows an extreme example of hypothesis learning where almost all Kleene algebra axioms have been commented out and various hypotheses – all of them previously verified – have been added.

```
op(500, infix, "+").
op(490, infix, ";").
op(480, postfix, "*").
op(500, infix, "-").

formulas(sos).
% Kleene algebra axioms  %%%%%%%%%%%%%%%%%%%%%%
    %x+y = y+x.
    %x+0 = x.
    %x+(y+z) = (x+y)+z.
    %x;1 = x.
    %1;x = x.
    x;(y;z) = (x;y);z.
    %0;x = 0.
    %x;0 = 0.
    %x;(y+z) = x;y+x;z.
    (x+y);z = x;z+y;z.
    %x+x = x.
    x <= y <-> x+y = y.
    1+x;x* = x*.
    %1+x*;x = x*.
    %z+x;y <= y -> x*;z <= y.
    %z+y;x <= y -> z;x* <= y.

% Boolean domain axioms %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %a(x);x = 0.
    %a(x;y) = a(x;a(a(y))).
    %a(a(x))+a(x)=1.

% syntactic sugar %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %d(x)=a(a(x)).
    %d(x)-d(y)=d(x);a(y).
    %om(x,d(y))=d(y)-d(x;d(y)).

% added Lemmas (all verified) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    x <= x.
    x <= y & y <= z -> x <= z.
    d(1;d(y))=d(y).
    d(x;y) = d(x;d(y)).
    d(om(x,d(y))) = om(x,d(y)).
    d(d(x)-d(y)) = d(x)-d(y).
    d(x+y)=d(x)+d(y).
    d(y)-om(x,d(y)) <= d(x;d(y)).
    d(x)-(d(y)+d(z)) = (d(x)-d(y))-d(z).
    d(x) <= d(y) -> d(x)-d(z) <= d(y)-d(z).
    d(x;d(y))-d(x;d(z)) <= d(x;(d(y)-d(z))).

end_of_list.

formulas(goals).
    (all y (d(y) <= d(x;d(y)) -> d(y) = 0)) -> (all y (d(y)-d(x*;om(x,d(y))) = 0)).

end_of_list.
```

Proof search is almost entirely based on domain-specific additional hypotheses. Mace4 has been substantial for axiom deletion: when hypotheses are too weak to entail the goal, counterexamples could be found. Here, hypothesis learning could to some extent exploit our manual proof, but additional experiments were needed. An automation of this procedure could drastically enhance the entire approach. A proof with less hypothesis learning could perhaps have been found with more patience.

References

[1] G. Birkhoff, Lattice Theory, in: Colloquium Publications, vol. 25, American Mathematical Society, 1984, reprint.
[2] J.W. Blok, D. Pigozzi, Algebraizable logics, Memoirs of the American Mathematical Society 77 (396) (1989).
[3] W. Degen, J.M. Werner, Towards intuitionistic dynamic logic, in: Proceedings of Studia Logica 2006, in: Logic and Logical Philosophy, vol. 15, Nicolaus Copernicus University Press, 2007, pp. 305–324.
[4] J. Desharnais, B. Möller, G. Struth, Termination in modal Kleene algebra, in: J.-J. Lévy, E.W. Mayr, J.C. Mitchell (Eds.), IFIP TCS2004, Kluwer, 2004, pp. 647–660. revised version: Algebraic Notions of Termination, Technical Report 2006–23, Institut für Informatik, Universität Augsburg, 2006.
[5] J. Desharnais, B. Möller, G. Struth, Kleene algebra with domain, ACM Transactions on Computational Logic 7 (4) (2006) 798–833.
[6] J. Desharnais, G. Struth, Domain axioms for a family of near-semirings, in: J. Meseguer, G. Roşu (Eds.), Algebraic Methodology and Software Technology, 12th International Conference (AMAST 2008), in: Lect. Notes in Comp. Sci., vol. 5140, Springer, 2008, pp. 330–345.
[7] J. Desharnais, G. Struth, Modal semirings revisited, in: P. Audebaud, C. Paulin-Mohring (Eds.), Mathematics of Program Construction, 9th International Conference, MPC 2008, in: Lect. Notes in Comp. Sci., vol. 5133, Springer, 2008, pp. 360–387.
[8] Z. É sik, W. Kuich, A semiring-semimodule generalization of ω-context-free languages, in: J. Karhumäki, H. Maurer, G. Păun, G. Rozenberg (Eds.), Theory is Forever, in: Lect. Notes in Comp. Sci., vol. 3113, Springer, 2004, pp. 68–80.
[9] J. Gunawardena (Ed.), Idempotency, Cambridge University Press, 1998.
[10] T. Hillenbrand, Waldmeister, http://www.waldmeister.org.
[11] P. Höfner, G. Struth, Automated reasoning in Kleene algebra, in: P. Pfenning (Ed.), Automated Deduction—CADE-21, in: Lect. Notes in Art. Intelligence, vol. 4603, Springer, 2007, pp. 279–294.
[12] P. Höfner, G. Struth, G. Sutcliffe, Automated verification of refinement laws, Annals of Mathematics and Artificial Intelligence 55 (1–2) (2009) 35–62.
[13] P. Jipsen, G. Struth, The structure of the one-generated free domain semiring, in: R. Berghammer, B. Möller, G. Struth (Eds.), Relations and Kleene Algebras in Computer Science, in: LNCS, vol. 4988, Springer, 2008, pp. 234–242.
[14] P.J. Johnstone, Stone Spaces, Cambridge University Press, 1982.
[15] B. Jónsson, A. Tarski, Boolean algebras with operators, Part I, American Journal of Mathematics 73 (1951) 891–939.
[16] D. Kozen, A completeness theorem for Kleene algebras and the algebra of regular events, Information and Computation 110 (2) (1994) 366–390.
[17] D. Kozen, Kleene algebra with tests, ACM Transactions on Programming Languages and Systems 19 (3) (1997) 427–443.
[18] H. Leiß, Kleene modules and linear languages, Journal of Logic and Algebraic Programming 66 (2) (2006) 185–194.
[19] P. Maier, Intuitionistic LTL and a new characterization of safety and liveness, in: A. Tarlecki, J. Marcinkowski (Eds.), Computer Science Logic, 18th International Workshop, in: Lect. Notes in Comp. Sci., vol. 3210, Springer, 2004, pp. 295–309.
[20] W. McCune, Prover9 and Mace4, http://www.cs.unm.edu/~mccune/prover9.
[21] B. Möller, G. Struth, Algebras of modal operators and partial correctness, Theoretical Computer Science 351 (2) (2006) 221–239.
[22] V. Sofronie-Stokkermans, Automated theorem proving by resolution in non-classical logics, Annals of Mathematics and Artificial Intelligence 49 (2007) 221–252.
[23] G. Struth, Proof database, http://www.dcs.shef.ac.uk/georg/ka.
[24] G. Struth, Modal tools for separation and refinement, Electronic Notes in Theoretical Computer Science 214 (2008) 81–101.
[25] G. Sutcliffe, C. Suttner, The CADE ATP system competition, www.cs.miami.edu/~tptp/CASC/.
[26] G. Sutcliffe, C. Suttner, The TPTP problem library for automated theorem proving, www.tptp.org.
[27] J. von Wright, Towards a refinement algebra, Science of Computer Programming 51 (1–2) (2004) 23–45.