Domain Axioms for a Family of Near-Semirings

Jules Desharnais[1] and Georg Struth[2]

[1] Département d'informatique et de génie logiciel, Université Laval, Canada
Jules.Desharnais@ift.ulaval.ca
[2] Department of Computer Science, University of Sheffield, United Kingdom
g.struth@dcs.shef.ac.uk

Abstract. Axioms for domain operations in several variants of Kleene algebras and their semiring reducts are presented. They provide abstract enabledness conditions for algebras designed for the verification and refinement of action systems, probabilistic protocols, basic processes and games. The axiomatisations are simpler, more uniform and more flexible than previous attempts; they are especially suited for automated deduction. This is further demonstrated through the automated verification of some classical refinement laws for action systems.

1 Introduction

Variants of Kleene algebras provide the basic operations for modelling the dynamics of discrete systems. Choices between actions or processes are modelled through addition, sequential composition through multiplication, finite and infinite iteration via fixed points. Additive identities capture deadlock or abortion; silent or ineffective actions correspond to multiplicative identities. A main benefit of the approach is its suitability for first-order automated deduction in applications where model checking or interactive theorem proving is usually employed.

Axiomatic variations are dictated by the semantics of application domains. Kleene algebras, for instance, capture partial program correctness under angelic choice [10] or trace models of reactive systems. Variants in which some axioms have been weakened admit predicate transformer models for total program correctness under demonic choice [16], expectation transformer models for probabilistic programs and protocols [11], and multirelational [7] or game-based models [8] for situations where angelic and demonic choices interact. Other variants of Kleene algebras provide algebraic semantics for parallel reactive systems modelled by action systems [3] or for basic process algebras [4].

A main source of variation is the interaction of choice with composition. For games and processes, for instance, a choice between the sequences xy and xz of actions x, y and z must be distinguished from a choice between y or z after execution of x, hence $x(y + z) \neq xy + xz$. Relation- or trace-based models, in contrast, require this distributivity law. Other applications might exclude that an infinite action x can be aborted after it has started, that is, $x0 \neq 0$. Again, this annihilation law certainly holds for binary relations.

In many of these applications, axiomatising a domain operator is essential. For Kleene algebras, domain has been defined as a map into an embedded Boolean

J. Meseguer and G. Roşu (Eds.): AMAST 2008, LNCS 5140, pp. 330–345, 2008.

algebra that models a state space [5]. In fact, it suffices to axiomatise domain on the semiring retract of the Kleene algebra. It turned out that this axiomatisation can essentially be reused for some weaker variants [13] and applied to demonic refinement algebras [15] and probabilistic Kleene algebras [12]. Operationally, domain provides enabledness conditions for programs or processes. Moreover, on Kleene algebras, domain induces modal operators and formalisms similar to dynamic logic. Recently, domain axioms for Kleene algebras have been provided in a one-sorted setting [6] which is simpler, more flexible and better suited for automated deduction. But is this new axiomatisation stable and robust enough to scale to weaker variants of Kleene algebras?

The present paper provides a positive answer. We adapt the new axiomatisation of domain semirings to demonic refinement algebras [16], probabilistic refinement algebras [11], basic process algebras [4] and other variants of Kleene algebras. Our first contribution is a systematic development of domain on families of near-semirings [14]. As in the semiring case, we provide axioms that make the algebras of domain elements into distributive lattices. Second, based on antidomain operations, we provide simple axiomatisations that induce Boolean domain algebras. It turns out that we can simply reuse the semiring domain axioms for demonic refinement algebras and probabilistic Kleene algebras. Third, we also consider codomain operations for all variants. The entire development and investigation is based on the automated theorem proving system (ATP system) Prover9 and the counterexample generator Mace4 [2]. We therefore do not display proofs we could automate, but provide an encoding for Prover9/Mace4 at a web site [1] from which all results in this paper can easily be reproduced. A fourth contribution is an application of our new axioms to the automated verification of some classical action system refinement laws [3,15].

2 From Near-Semirings to Semirings

To model actions or processes, we consider weak variants of semirings with different identities. A *near-semiring* [14] is a structure $(S, +, \cdot)$ such that $(S, +)$ and $(S, \cdot)$ are semigroups and all elements satisfy the *right distributivity* law

$$(x + y)z = xz + yz \ . \tag{1}$$

Here and henceforth, the multiplication symbol is omitted. A *pre-semiring* is a near-semiring in which all elements satisfy the *left pre-isotonicity* law

$$x + y = y \Rightarrow zx + zy = zy \ . \tag{2}$$

A *semiring* is a near-semiring in which all elements also satisfy the left distributivity law $x(y + z) = xy + yz$. Every semiring is also a pre-semiring. We will restrict our attention to *commutative* near-semirings without explicitly mentioning the commutativity law $x + y = y + x$. A near-semiring is *idempotent* if all elements satisfy $x + x = x$. We also consider different identities.

- δ satisfies the identity and left annihilation axioms $x + \delta = x$ and $\delta x = \delta$.
- τ satisfies the right identity axiom $x\tau = x$.
- 0 satisfies the δ-axioms and the right annihilation axiom $x0 = 0$.
- 1 satisfies the τ-axiom and the left identity axiom $1x = x$.

In the presence of both additive or both multiplicative identities, these entities coincide: $\delta = \delta 0 = 0$ and $\tau = 1\tau = 1$. We will not consider 0 in this paper.

Idempotent near-semirings $(S, +, \cdot)$, possibly with δ and τ, are also called *basic process algebras*. There, δ is called *deadlock* and τ the *silent action* [4]. Idempotent pre-semirings $(S, +, \cdot, 0, 1)$ arise as reducts of probabilistic Kleene algebras [11] by forgetting an operation of finite iteration, and as game algebras [8]. Idempotent semirings $(S, +, \cdot, \delta, 1)$ arise as reducts of demonic refinement algebras [16] by forgetting an operation of strong iteration (see Section 9). We will see that a uniform treatment of the last three variants can be achieved via pre-semirings $(S, +, \cdot, \delta, 1)$.

Explicit axioms for near-semirings and pre-semirings, as input for Prover9, can be found at a web site [1]. The following fact has been verified with Prover9.

Lemma 2.1. *The relation $\leq$ defined, for all elements x and y of an idempotent near-semiring, by $x \leq y \Leftrightarrow x + y = y$ is a partial order. The identity δ or 0 is the least element with respect to that order if it exists. Addition and right multiplication are isotone with respect to the order.*

Mace4 yields a 3-element counterexample to isotonicity of left multiplication for near-semirings and a 4-element one for those with δ and 1. By Lemma 2.1, every idempotent near-semiring can be ordered and $(S, +)$ is a semilattice.

3 Domain Semirings

An operation of domain for semirings has been defined in a companion paper [6]. A *domain semiring* is a semiring $(S, +, \cdot, 0, 1)$ extended by a function $d : S \to S$ that satisfies

$$x + d(x)x = d(x)x \ , \qquad \text{(D1)} \qquad\qquad d(x) + 1 = 1 \ , \qquad \text{(D4)}$$

$$d(xy) = d(xd(y)) \ , \qquad \text{(D2)} \qquad\qquad d(0) = 0 \ . \qquad \text{(D5)}$$

$$d(x + y) = d(x) + d(y) \ , \qquad \text{(D3)}$$

We call (D1), (D2) and (D3) the *basic domain axioms* and overload (D4) and (D5) also for τ and δ. Every domain semiring is automatically idempotent [6]. Mace4 easily shows that the domain axioms are irredundant, that is, counterexamples exist for each mutual implication.

The axioms can be abstracted from relational models, where the domain $d(x)$ of a binary relation x is the binary relation consisting of all ordered pairs (a, a) with $(a, b) \in x$ for some b. (D1) says that $d(x)$ does not restrict the execution of x. (D2) says that the enabledness of a sequence xy depends on y only through its enabledness condition. (D3) says that the enabledness condition for a choice

between actions is the union of the enabledness conditions of the particular actions. (D4) says that all enabledness conditions are below 1 and (D5) says that enabling abort or deadlock yields abort or deadlock.

The set of domain elements of S is denoted by $d(S)$ and it has been shown that $d(S) = \{x \in S : d(x) = x\}$, whence domain elements are precisely the fixpoints of the domain operation. This can be used to show that the *domain algebra* $(d(S), +, \cdot, 0, 1)$ of a domain semiring S is a bounded distributive lattice. By $d(x)+d(y) = d(d(x)+d(y))$, e.g., domain elements are closed under addition.

The domain algebras of domain semirings can be turned into Boolean algebras by adding an antidomain operation $a : S \to S$ that satisfies

$$d(x) + a(x) = 1 \ , \qquad (3) \qquad\qquad d(x)a(x) = 0 \ . \qquad (4)$$

The resulting structures are called *Boolean domain semirings*. It can be shown that the antidomain of an element is precisely the Boolean complement of its domain, hence $a(x)$ models those states for which x is not enabled. Since $a^2(x) = d(x)$ holds, domain can be eliminated from all axioms and it follows that a semiring is a Boolean domain semiring if and only if it satisfies the *basic Boolean domain axioms*

$$a(x)x = 0 \ , \qquad \text{(BD1)} \qquad a^2(x) + a(x) = 1 \ . \qquad \text{(BD3)}$$
$$a(xy) + a(xa^2(y)) = a(xa^2(y)) \ , \qquad \text{(BD2)}$$

These considerations form the basis for axiomatising domain on near-semirings.

4 Domain Conditions

Before investigating domain on a family of near-semirings, we collect some natural conditions that each domain operation should satisfy. First of all, axiomatisations should respect our intuitions about domain. Second, as in the semiring case, they should induce distributive lattices or Boolean algebras.

Let S be a near-semiring. The main intuition behind domain is that a domain element $d(x)$ is a left preserver of $x \in S$ in the sense that $x \leq d(x)x$, as expressed by (D1), or even

$$x = d(x)x \ . \qquad (5)$$

Since 1 is also a left preserver of x (if it exists), $d(x)$ should even be the *least left preserver* of x. Hence, for all $x \in S$ and $p \in d(S)$,

$$x \leq px \Leftrightarrow d(x) \leq p \ . \qquad (6)$$

Similarly, antidomain elements should be *greatest left annihilators*, that is,

$$px = \delta \Leftrightarrow p \leq a(x) \ . \qquad (7)$$

All axiomatisations of domain should therefore respect (5) and (6); all axiomatisations of antidomain should respect (7).

To induce a lattice as a domain algebra, it is necessary and sufficient that each domain element satisfies, besides the basic domain axioms, the *lattice conditions*

$$d^2(x) = d(x) \ , \tag{8}$$
$$d(d(x) + d(y)) = d(x) + d(y) \ , \tag{9}$$
$$d(d(x)d(y)) = d(x)d(y) \ , \tag{10}$$
$$(d(x))^2 = d(x) \ , \tag{11}$$
$$d(x)d(y) = d(y)d(x) \ , \tag{12}$$
$$d(x) = d(x) + d(x)d(y) \ , \tag{13}$$
$$d(x) = d(x)(d(x) + d(y)) \ . \tag{14}$$

The first three identities are closure conditions, and, more precisely, they are necessary if the fixpoint characterisation of domain elements holds. The other conditions correspond to lattice axioms. In the presence of τ or 1, $d(\tau) = \tau$ or $d(1) = 1$ should hold as well. The condition $d(\delta) = \delta$ holds by axiom (D5).

In the presence of (3) and (4), the *Boolean conditions*

$$d(a(x)) = a(x) \ , \tag{15}$$
$$a(x)d(x) = \delta \tag{16}$$

should also be checked, but condition (16) follows from (4) and (12). The first of the following conditions is needed for d-elimination; the second one is dual to (D3) and again very natural.

$$a^2(x) = d(x) \ , \tag{17}$$
$$a(x + y) = a(x)a(y) \ . \tag{18}$$

Finally, for non-idempotent near-semirings, it must be checked that the resulting domain weak semiring is idempotent, otherwise addition does not model choice.

We call a domain near-semirings or Boolean domain near-semiring *healthy* if it satisfies the relevant conditions among (5) to (18). Further natural properties may arise in particular applications and these can be added as axioms if needed. Also, we always make sure that axiomatisations are *irredundant* in the sense that no axiom is entailed by the remaining ones. This can usually (but not necessarily) be established by Mace4 through finite counterexamples. In the case of pre-semirings, additional domain conditions have a substantial impact on domain algebras. These will be investigated in Section 6.

5 A Family of Domain Near-Semirings

We now consider domain or enabledness axioms for near-semirings with and without δ, τ and 1. The general recipe is as follows: Start with the basic domain axioms plus the domain axioms for the respective identities. Then add domain conditions until the axiomatisation is healthy and induces a distributive lattice. Finally, remove redundancies. Prover9 and Mace4 allowed us to considerably simplify this analysis and we do not display any proofs that could be automated.

We first axiomatise various domain near-semirings. We do not investigate all possible combinations of identities, but restrict ourselves to structures that have previously been considered in applications. Pre-semirings with 1, for instance, form the basis for probabilistic Kleene algebras, game algebras and demonic refinement algebras, but we do not investigate pre-semirings with τ.

A *domain near-semiring* is a near-semiring $(S, +, \cdot)$ extended by a domain function $d : S \to S$ that satisfies (5), (D2), (D3), (12) and (14).

Table 1. A Family of Domain Near-Semirings

		NS	NS_δ	NS^τ	NS^τ_δ	NS^1	NS^1_δ	PS^1	PS^1_δ
(D1)	$x \leq d(x)x$					√	√	√	√
(D2)	$d(xy) = d(xd(y))$	√	√	√	√	√	√	√	√
(D3)	$d(x+y) = d(x) + d(y)$	√	√	√	√	√	√	√	√
(D4)	$d(x) \leq \tau$			√	√				
(D4)	$d(x) \leq 1$					√	√	√	√
(D5)	$d(\delta) = \delta$		√		√		√		√
(5)	$x = d(x)x$	√	√	√	√				
(12)	$d(x)d(y) = d(y)d(x)$	√	√	√	√	√	√		
(14)	$d(x) = d(x)(d(x) + d(y))$	√	√						

NS: near-semiring, PS: pre-semiring.

A *domain near-semiring with* τ is a near-semiring $(S, +, \cdot, \tau)$ extended by a function $d : S \to S$ that satisfies (5), (D2), (D3), (D4) and (12).

A *domain near-semiring with* 1 is a near-semiring $(S, +, \cdot, 1)$ extended by a function $d : S \to S$ that satisfies the domain axioms (D1)-(D4) and (12).

A *domain pre-semiring* (with 1) is a pre-semiring $(S, +, \cdot, 1)$ extended by a function $d : S \to S$ that satisfies (D1)-(D4).

In each case a variant with δ is obtained by adding (D5). The explicit domain axioms for these structures are displayed in Table 1.

The following fact has been verified by Prover9 and Mace4.

Lemma 5.1. *All domain axiomatisations are healthy and irredundant.*

For all domain near-semirings without 1, (5) cannot be replaced by (D1), since in that case, (10) or idempotency would not be entailed. For near-semirings with 1, (D1) can be used. Healthiness implies the following facts.

Lemma 5.2. *Domain near-semirings are idempotent and can be ordered.*

Hence the approach applies to basic process algebras, probabilistic Kleene algebras, game algebras and demonic refinement algebras, which are all idempotent.

Lemma 5.3. *Domain elements of near-semirings are least left preservers.*

So all axiomatisations respect our basic intuitions about domain and enabledness. Mace4 can easily show that all classes considered are indeed distinct.

We now investigate the impact of healthiness on the domain algebra. First, we can characterise domain elements within the language of domain weak semirings.

Lemma 5.4. *An element of a domain near-semiring is a domain element if and only it is a fixpoint of the domain operation.*

Proof. Let S be a near-semiring with a healthy mapping d, whence in particular $d^2(x) = d(x)$ holds for all $x \in S$. We show that $x \in d(S)$ if and only if $x = d(x)$. First, every $x \in d(S)$ is the image of some $y \in S$, that is, $x = d(y)$. Therefore,

$d(x) = d(d(y)) = d(y) = x$ holds by healthiness. Second, $x = d(x)$ trivially implies that $x \in d(S)$. □

So (9) and (10) are indeed closure conditions for domain elements, and the domain algebras can easily be characterised.

Proposition 5.5. *Let S be a domain near-semiring. Then $(d(S), +, \cdot)$ is a distributive lattice. If the near-semiring has an additive (multiplicative) identity, it is the least (greatest) element of the lattice.*

Proof. The lattice conditions imply that $d(S)$ forms a lattice. The right distributivity axiom of near-semirings holds in particular for domain elements. By standard lattice theory, $d(S)$ is therefore a distributive lattice. The bound conditions could readily be checked with Prover9. □

Let us further discuss these results. We have seen that all near-semirings considered can be endowed with simple equational domain axioms that induce an order on the near-semiring and a domain algebra which is a distributive lattice. These axioms support our basic intuitions about domain and enabledness. In the case of pre-semirings with 1, which form the basis for probabilistic Kleene algebras, game algebras and demonic refinement algebras, the basic domain axioms of domain semirings [6] can entirely be reused.

There is, however, a crucial difference between domain semirings and domain for the weaker variants considered. Forward modal operators can be defined on a domain semiring S as $|x\rangle p = d(xp)$, for all $x \in S$ and $p \in d(S)$. The name "modal operator" is justified since $\lambda p.|x\rangle p$ is a strict and additive mapping, that is, it satisfies $|x\rangle 0 = 0$ and $|x\rangle(p+q) = |x\rangle p + |x\rangle q$. For weaker variants, $\lambda p.d(xp)$ need be neither strict nor additive. Prover9 and Mace4 could show that strictness holds only in the presence of the right annihilation law and additivity holds only in the presence of the left distributivity law. Therefore, none of the weak variants considered gives rise to a *modal* near-semiring; we do not obtain basic process algebras, probabilistic Kleene algebras, game algebras or demonic refinement algebras with modal operators from the domain axioms. This is an important negative result. The situation is different for backward diamonds which are based on an axiomatisation of codomain (cf. Section 8).

6 Boolean Domain Conditions

In domain semirings, domain algebras are strongly linked with maximal Boolean subalgebras [6]. Prover9 could show that this link still exists for domain pre-semirings with δ and 1, but not for near-semirings.

Proposition 6.1. *Let S be a domain pre-semiring with δ and 1. An element $x \in S$ is a domain element if some $y \in S$ satisfies $x + y = 1$ and $yx = \delta$.*

This statement does not hold in domain near-semirings with δ and 1; Mace4 presented a 5-element counterexample.

Corollary 6.2. *Elements x and y of a domain pre-semiring with δ and 1 are domain elements if $x + y = 1$, $xy = \delta$ and $yx = \delta$.*

Again, Mace4 presented a 5-element counterexample for near-semirings.

We say that an element x of a *weak semiring*near-semiring S is *complemented* if there exists some $y \in S$ such that $x + y = 1$, $xy = \delta$ and $yx = \delta$. We denote the set of all complemented elements in S by B_S.

Lemma 6.3. *Let $(S, +, \cdot, \delta, 1)$ be a domain pre-semiring. Then $(B_S, +, \cdot, \delta, 1)$ is a Boolean algebra.*

Proof. First, if x is complemented, then x is idempotent. Second, if x and y are complemented, then $xy = yx$. Third, if x and y are complemented, then so are $x + y$ and xy. The second fact has a 280-step proof, the third one a 212-step proof with Prover9. The first fact requires almost no time. □

Lemma 6.3 has considerable impact on the structure of domain algebras.

Theorem 6.4. *Let $(S, +, \cdot, \delta, 1)$ be a domain pre-semiring. Then $d(S)$ contains the greatest Boolean subalgebra of S bounded by δ and 1.*

Again, Mace4 showed that Lemma 6.3 and Theorem 6.4 do not generalise to near-semirings. Also, the domain algebra of a domain pre-semiring with δ and 1 need not itself be Boolean.

7 A Family of Boolean Domain Near-Semirings

We now provide axioms for near-semirings with δ and τ or 1 which induce Boolean domain algebras. This situation corresponds perhaps most closely to the state spaces or propositional structures underlying practical applications, but as for semirings, Heyting domain algebras should also be possible [6].

A *Boolean domain pre-semiring* is a domain pre-semiring $(S, +, \cdot, \delta, 1)$ that satisfies the domain axioms (D1)-(D5) and that is extended by an *antidomain operation* $a : S \to S$ that satisfies (3) and (4).

Lemma 7.1. *Boolean domain pre-semirings are healthy.*

The proof of (18) with Prover9 has 168 steps. Corollary 6.2 and Theorem 6.4 imply the following fact.

Proposition 7.2. *The domain algebra of a Boolean domain pre-semiring is the maximal Boolean subalgebra of the pre-semiring of subidentities.*

Healthiness also implies that $a^2(x) = d(x)$, whence, as in the semiring case, domain can be eliminated from the axiomatisation and the following theorem could be shown automatically by Prover9.

Theorem 7.3. *A pre-semiring S is a Boolean domain pre-semiring if and only if it can be extended by an antidomain operation $a : S \to S$ that satisfies the basic Boolean domain axioms (BD1), (BD2) and (BD3).*

Moreover, Mace4 easily showed the following fact.

Lemma 7.4. *The axioms (BD1)-(BD3) are irredundant.*

Therefore, the basic Boolean domain axioms for semirings can be reused for probabilistic Kleene algebras, game algebras and demonic refinement algebras.

We now consider domain near-semirings. First, we turn to the case with δ and 1. A *Boolean domain near-semiring with δ and* 1 is a domain near-semiring $(S, +, \cdot, \delta, 1)$ that satisfies the domain axioms (D1)-(D5) and (12) and that is extended by an *antidomain operation* $a : S \to S$ that satisfies (3), (4) and (15).

Lemma 7.5. *Boolean domain near-semirings with δ and* 1 *are healthy and irredundant.*

This could be shown by Prover9. Also, by Mace4, the Boolean domain semiring axioms alone are too weak. The following fact is an immediate consequence.

Proposition 7.6. *Boolean domain near-semirings with δ and* 1 *have Boolean domain algebras.*

However, this Boolean algebra need not always be maximal. Mace4 presented a 5-element counterexample to Corollary 6.2. Healthiness again implies that $a^2(x) = d(x)$, whence domain can be eliminated from the axiomatisation and the following theorem could be shown by Prover9.

Theorem 7.7. *A near-semiring $(S, +, \cdot, \delta, 1)$ is a Boolean domain near-semiring if and only if it can be extended by an antidomain operation $a : S \to S$ that satisfies the axioms (BD1)-(BD3) and (18).*

Moreover, Mace4 easily showed that these antidomain axioms are irredundant.

A *Boolean domain near-semiring with δ and τ* is a domain near-semiring $(S, +, \cdot, \delta, \tau)$ that satisfies (5), (D2), (D3), (D5) and (12) and that is extended by an *antidomain operation* $a : S \to S$ that satisfies (3), (4) and (15).

Lemma 7.8. *Boolean domain near-semirings with δ and τ are healthy and irredundant.*

This could be proved by Prover9, too. The following fact follows immediately.

Proposition 7.9. *Boolean domain near-semirings with δ and τ have Boolean domain algebras.*

Again, this Boolean algebra need not be maximal; there is a 5-element counterexample. Since healthiness implies that $a^2(x) = d(x)$, domain can be eliminated from the axiomatisation and the axioms can somewhat be simplified. However, the compaction obtained is not comparable to the stronger near-semirings and we therefore do not provide a deeper discussion.

The axioms for our family of Boolean domain near-semirings are summed up in Table 2. The ordering can be used because of Lemma 5.2. The first column is relevant for basic process algebras; the last column for probabilistic Kleene algebras, game algebras and demonic refinement algebras. The axiomatisation

Table 2. A Family of Boolean Domain Near-Semirings

		$\mathsf{NS}^{\tau}_{\delta}$	NS^{1}_{δ}	PS^{1}_{δ}
(BD1)	$a(x)x = \delta$		$\surd$	$\surd$
(BD2)	$a(xy) \leq a(xa^2(y))$		$\surd$	$\surd$
(BD3)	$a^2(x) + a(x) = 1$		$\surd$	$\surd$
(18)	$a(x+y) = a(x)a(y)$		$\surd$	
(5)	$x = d(x)x$	$\surd$		
(D2)	$d(xy) = d(xd(y))$	$\surd$		
(D3)	$d(x+y) = d(x) + d(y)$	$\surd$		
(D5)	$d(\delta) = \delta$	$\surd$		
(3)	$d(x) + a(x) = 1$	$\surd$		
(4)	$d(x)a(x) = 0$	$\surd$		
(12)	$d(x)d(y) = d(y)d(x)$	$\surd$		
(15)	$d(a(x)) = a(x)$	$\surd$		

NS: near-semiring, PS: pre-semiring.

in that case is precisely that of Boolean domain semirings [6] and the three basic Boolean domain axioms that need to be added to the semiring axioms are simpler and better suited for automated deduction than those of previous approaches [12,13,15], in which the Boolean algebra of states had to be axiomatised and embedded explicitly in a two-sorted setting.

8 Codomain

In the semiring case, domain and codomain are duals with respect to semiring opposition, which swaps the order of multiplication. Weaker variants break this symmetry and codomain therefore deserves special attention. Codomain is of independent interest because it induces an image operation which is useful, for instance, in the context of Hoare-style logics and for reachability analysis.

A *codomain semiring* is a semiring $(S, +, \cdot, 0, 1)$ extended by a function $d^\circ : S \to S$ that satisfies the basic codomain axioms

$$x + xd^\circ(x) = xd^\circ(x) \ , \qquad d^\circ(x) + 1 = 1 \ ,$$
$$d^\circ(xy) = d^\circ(d^\circ(x)y) \ , \qquad d^\circ(0) = 0 \ .$$
$$d^\circ(x+y) = d^\circ(x) + d^\circ(y) \ ,$$

We call an expression in the language of codomain near-semirings *dual* to an expression in the language of domain near-semirings if it is dual with respect to opposition, each term $d(x)$ is replaced by $d^\circ(x)$, and each term $a(x)$ is replaced by $a^\circ(x)$. Here, a° denotes the anticodomain operation. We refer to antidomain axioms as the duals of domain axioms. For instance, we write (D1°) for the dual of (D1), and likewise for the lattice and healthiness conditions.

Table 3. A Family of Codomain Near-Semirings

		NS	NS_δ	NS^τ	NS^τ_δ	NS^1	NS^1_δ	PS^1	PS^1_δ
	$x + x = x$	√	√	√	√				
(D1°)	$x \leq xd^\circ(x)$							√	√
(D2°)	$d^\circ(xy) = d^\circ(d^\circ(x)y)$	√	√	√	√	√	√	√	√
(D3°)	$d^\circ(x + y) = d^\circ(x) + d^\circ(y)$	√	√	√	√	√	√	√	√
(D4°)	$d^\circ(x) \leq \tau$			√	√				
(D4°)	$d^\circ(x) \leq 1$					√	√	√	√
(D5°)	$d^\circ(\delta) = \delta$		√		√		√		√
(5°)	$x = xd^\circ(x)$	√	√	√	√	√	√		
(12°)	$d^\circ(x)d^\circ(y) = d^\circ(y)d^\circ(x)$	√	√	√	√	√	√		
(14°)	$d^\circ(x) = d^\circ(x)(d^\circ(x) + d^\circ(y))$	√	√						
	$d^\circ(\tau) = \tau$			√	√				

NS: near-semiring, PS: pre-semiring.

Because of the lack of duality, the codomain axioms for our family of near-semirings differ from the domain axioms if healthiness is to be preserved. In particular, idempotency must sometimes be assumed.

A *codomain near-semiring* is an idempotent near-semiring $(S, +, \cdot)$ extended by a function $d^\circ : S \to S$ that satisfies (5°), (D2°), (D3°), (12°) and (14°).

A *codomain near-semiring with* τ is an idempotent near-semiring $(S, +, \cdot, \tau)$ extended by $d^\circ : S \to S$ that satisfies (5°), (D2°)-(D4°), (12°) and $d^\circ(\tau) = \tau$.

A *codomain near-semiring with* 1 is a near-semiring $(S, +, \cdot, 1)$ extended by $d^\circ : S \to S$ that satisfies (5°), (D2°)-(D4°) and (12°).

A *codomain pre-semiring* is a pre-semiring $(S, +, \cdot, 1)$ extended by $d^\circ : S \to S$ that satisfies (D1°)-(D4°).

Variants with δ are obtained by adding (D5°). Table 3 shows all axiomatisations.

The following statements could be proved by Prover9.

Lemma 8.1

(i) All axiomatisations are healthy and irredundant.
(ii) All codomain near-semirings are idempotent and can be ordered.
(iii) Codomain elements of codomain near-semirings are least right preservers.
(iv) An element of a codomain near-semiring is a codomain element if and only it is a fixpoint of the codomain operation.

Proposition 8.2. *Let S be a codomain near-semiring. Then $(d^\circ(S), +, \cdot)$ is a distributive lattice. If the near-semiring has an additive (multiplicative) identity, it is the least (greatest) element of the lattice.*

In Section 5 we saw that the domain operations on our family of near-semirings did not induce modal operators. Here the situation is different.

Proposition 8.3. *For every codomain near-semiring S, all $x \in S$ and all $p, q \in d^\circ(S)$ satisfy $d^\circ((p+q)x) = d^\circ(px)+d^\circ(qx)$, and $d^\circ(\delta) = \delta$ if this identity exists.*

Table 4. A Family of Boolean Codomain Near-Semirings

		$\mathsf{NS}^{\tau}_{\delta}$	NS^{1}_{δ}	PS^{1}_{δ}
	$x \leq xa^\circ(a^\circ(x))$			$\surd$
(BD2°)	$a^\circ(xy) \leq a^\circ(a^\circ(a^\circ(x))y)$			$\surd$
(BD3°)	$a^\circ(a^\circ(x)) + a^\circ(x) = 1$			$\surd$
	$a^\circ(x)a^\circ(a^\circ(x)) = \delta$			$\surd$
(5°)	$x = xd^\circ(x)$	$\surd$	$\surd$	
(D2°)	$d^\circ(xy) = d^\circ(d^\circ(x)y)$	$\surd$	$\surd$	
(D3°)	$d^\circ(x + y) = d^\circ(x) + d^\circ(y)$	$\surd$	$\surd$	
(D5°)	$d^\circ(\delta) = \delta$	$\surd$	$\surd$	
(12°)	$d^\circ(x)d^\circ(y) = d^\circ(y)d^\circ(x)$	$\surd$	$\surd$	
(3°)	$d^\circ(x) + a^\circ(x) = 1$		$\surd$	
(3°)	$d^\circ(x) + a^\circ(x) = \tau$	$\surd$		
(4°)	$a^\circ(x)d^\circ(x) = \delta$	$\surd$	$\surd$	

NS: near-semiring, PS: pre-semiring.

So codomain on near-semirings is strict and additive, and it induces backward diamond operators $\langle x|p = d^\circ(px)$ defined via images.

In analogy to Boolean domain near-semirings, we now axiomatise an anti-codomain operation in order to obtain Boolean codomain algebras.

A *Boolean codomain near-semiring with δ and τ* is a codomain near-semiring $(S, +, \cdot, \delta, \tau)$ that satisfies (5°), (D2°), (D3°), (D5°) and (12) and that is extended by an *anticodomain operation* $a^\circ : S \to S$ satisfying (3°) and (4°). In particular, the near-semiring is idempotent. The definition of *Boolean codomain near-semiring with δ and* 1 is analogous; both axiom sets are shown in Table 4.

A *Boolean codomain pre-semiring* is a codomain pre-semiring $(S, +, \cdot, \delta, 1)$ that satisfies (D1°)-(D3°), (D5°) and that is extended by an *anticodomain operation* $a^\circ : S \to S$ satisfying (3°) and (4°).

Lemma 8.4. *All axiomatisations satisfy the Boolean conditions (3°) and (4°); their axioms are irredundant.*

However, Boolean codomain near-semirings can be unhealthy. Mace4 showed that each class contains models that do not satisfy (7°) or (18°). This remains true for semirings with δ; the identity $x0 = 0$ is needed for healthiness. These counterexamples formally support a previous remark in the two-sorted setting for pre-semirings with δ and 1 [13]. Still we obtain the following result.

Proposition 8.5. *Boolean codomain near-semirings have Boolean domain algebras. Those of pre-semirings are maximal in the pre-semirings of subidentities.*

In the case of Boolean codomain near-semirings (with δ and 1), Mace4 presents a 16-element counterexample to Corollary 6.2, hence to maximality.

Proposition 8.6. *A pre-semiring S is a Boolean codomain pre-semiring if and only if it can be extended by an antidomain operation $a^\circ : S \to S$ that satisfies $x \leq xa^\circ(a^\circ(x))$, the axioms (BD2°),(BD3°) and $a^\circ(x)a^\circ(a^\circ(x)) = 0$*

These axioms are also displayed in Table 4. In sum, the development of codomain near-semirings is similar to that of domain near-semirings, but, due to the lack of duality with respect to opposition, slightly different and less compact axiomatisations arise. A significant difference is that—unlike for domain semirings—modal operators are induced by the codomain operations.

9 Automated Action System Refinement

To demonstrate the power of our axiomatisations for formal software development, we automatically verified some well-known action system refinement laws [3], the proofs of which have already been replayed manually with demonic refinement algebras [15] and the two-sorted domain axiomatisation [5].

Formally, a *demonic refinement algebra* [15] is a structure $(S, +, \cdot, \delta, 1, ^\omega)$ such that $(S, +, \cdot)$ is an idempotent semiring and the *strong iteration* operation $^\omega : S \to S$ satisfies the unfold and the coinduction axiom

$$1 + xx^\omega = x^\omega \qquad \text{and} \qquad y \leq z + xy \Rightarrow y \leq x^\omega z \ .$$

von Wright's original axiomatisation also uses an operation of finite iteration that interacts with the strong variant [16]. Demonic refinement algebras model positively conjunctive predicate transformers over some state space, which themselves model demonically nondeterministic programs according to a weakest precondition semantics [16]. The law $1 + x^\omega x = x^\omega$ follows from the demonic refinement algebra axioms. Intuitively, strong iteration models a loop which is possessed by a demon, that is, which may be finite or infinite.

We define the *normaliser* $\mathsf{n}(x)$ of an element x as

$$\mathsf{n}(x) = x^\omega a(x) \ .$$

Intuitively, $\mathsf{n}(x)$ relates the states in the domain $d(x)$ of an action x with states from which no further iteration is possible, hence with x-normal forms.

It is stipulated that an action x *excludes* an action y if $x = a(y)x$ [15]. But there is a more appealing equivalent definition for near-semirings $(S, +, \cdot, 0, 1)$: x excludes y iff $d(x)d(y) = 0$, that is, if they are not jointly enabled. It immediately follows that exclusion is commutative.

Action systems formalise parallel reactive systems as loops containing demonic choices between individual actions which terminate when no more action is enabled. In the algebraic semantics of demonic refinement algebras,

$$\mathsf{do}\ x_0 \ [\!]\ \ldots\ [\!]\ x_{n-1}\ \mathsf{od} = \mathsf{n}(\sum_{i=0}^{n-1} x_i) = \sum_{i=0}^{n-1} x_i^\omega \prod_{i=0}^{n-1} a(x_i) \ .$$

We first automatically verified the *action system leapfrog* law [3]

$$\mathsf{do}\ xy\ \mathsf{od}\ x \leq x\ \mathsf{do}\ yx\ \mathsf{od}$$

for a loop without choice. In demonic refinement algebra it corresponds to

$$\mathsf{n}(xy)x \leq x\mathsf{n}(yx) \ .$$

Statements of comparable complexity usually require hypothesis learning and also here we could not prove the theorem in one full sweep within reasonable time. Therefore we started with a set of hypotheses from which explosive axioms like commutativity of addition have been discarded. We added further axioms or lemmas until Mace4 failed to detect a counterexample, that is, until we could expect that the hypotheses entail the goal. Then we ran the ATP system and, when this failed within reasonable time, tried another hypothesis set.

For proving the action system leapfrog we used the additional hypotheses $a(xy)x = a(xy)xa(y)$, which itself could be proved by Prover9 in 168 steps, left-isotonicity of multiplication and the sliding rule $x(yx)^\omega = (xy)^\omega x$, which has automatically been verified before [9]. Then Prover9 needed 52 steps and the equational proof extracted is simpler than that from the literature [15].

Second, we automatically verified the *action system decomposition law* [3]

$$\mathsf{do}\ x\ [\!]\ y\ \mathsf{od} = \mathsf{do}\ y\ \mathsf{od}\ \mathsf{do}\ x\ \mathsf{do}\ y\ \mathsf{od}\ \mathsf{od} \ ,$$

which holds if x excludes y. In demonic refinement algebra we must prove that

$$x = a(y)x \Rightarrow \mathsf{n}(x+y) = \mathsf{n}(y)\mathsf{n}(x\mathsf{n}(y)) \ .$$

Following Solin and von Wright [15], we added $d(x)\top = x\top$ as a further hypothesis, where $\top = 1^\omega$ is the maximal element of the algebra.

Irredundancy of this identity could easily be established through a 5-element counterexample by Mace4, which answers a question by Solin and von Wright.

Now this additional hypothesis implies that $\top = \mathsf{n}(x)\top$, which could be shown by Prover9 in 56 steps. The equational proof is

$$\top = a(x)\top + d(x)\top = a(x)\top + x\top \leq \mathsf{n}(x)\top \leq \top \ ,$$

where the third step uses coinduction. Using this fact as a hypothesis together with the standard law $(x+y)^\omega = y^\omega(xy^\omega)^\omega$, which has already been automatically verified [9], again the sliding rule and commutativity of antidomain elements allowed Prover9 to show our claim in 56 steps. Surprisingly, the assumption $x = a(y)x$ instead of $d(x)d(y) = 0$ turned out to be beneficial here.

The property $d(x) \leq d(x\mathsf{n}(y))$ has been used in the previous more complex manual proof [15]. Using $\top = n(x)\top$ again, we could find an instantaneous automated proof which yields a simpler equational argument:

$$a(x) = a(xd(\top)) = a(x\top) = a(x\mathsf{n}(y)\top) = a(x\mathsf{n}(y)d(\top)) = a(x\mathsf{n}(y)) \ .$$

In all these examples, using an ATP system therefore led to particularly simple proofs. Similar results for other domain near-semirings can be expected.

10 Conclusion

We have axiomatised domain operations that serve as enabledness conditions for variants of Kleene algebras with applications in program refinement, the analysis of probabilistic protocols, game theory and process algebra. The axioms obtained are simpler, more flexible and better suited for automation than previous approaches. They provide a basis from which further constraints imposed by the semantics of applications can be included. In the case of process algebras, for instance, the interaction of enabledness with parallel composition needs further investigation. We have also shown that the approach yields efficient automated proof support for applications in the refinement of parallel reactive systems.

The study of domain in weak Kleene algebras was strongly based on the ATP system Prover9 and the counterexample generator Mace4. These tools allowed us to drastically speed up the analysis, condense the presentation and dispense with routine technical proofs while even gaining in trustworthiness. The automated game of conjectures and refutations, the search for proofs and counterexamples, often took only a few seconds where humans would easily have spent several hours, and hardly more than a few minutes on a standard PC.

Beyond, that, the integration of algebraic methodology into off-the-shelf ATP technology could contribute towards bridging the gap between higher-order proof checking and model checking in software verification, and yield light-weight formal methods with heavy-weight automation.

The next step is to link the abstract algebraic level with concrete data (types) and their manipulation through assignment or communication. To achieve this as far as possible within ATP systems and to integrate appropriate decision procedures remains a challenge both for program analysis and theorem proving.

References

1. http://www.dcs.shef.ac.uk/~georg/ka
2. Prover9 and Mace4, http://www.cs.unm.edu/~mccune/prover9
3. Back, R.J.R., von Wright, J.: Reasoning algebraically about loops. Acta Informatica 36(4), 295–334 (1999)
4. Bergstra, J.A., Fokkink, W.J., Ponse, A.: Process algebra with recursive operations. In: Bergstra, J.A., Ponse, A., Smolka, S.A. (eds.) Handbook of Process Algebra, pp. 333–389. Elsevier, Amsterdam (2001)
5. Desharnais, J., Möller, B., Struth, G.: Kleene algebra with domain. ACM TOCL 7(4), 798–833 (2006)
6. Desharnais, J., Struth, G.: Domain semirings revisited. Technical Report CS-08-01, Department of Computer Science, University of Sheffield (2008); Accepted for Mathematics of Program Construction (MPC) (2008)
7. Furusawa, H., Tsumagari, N., Nishizawa, K.: A non-probabilistic model of probabilistic Kleene algebra. In: Berghammer, R., Möller, B., Struth, G. (eds.) Relations and Kleene Algebra in Computer Science. LNCS, vol. 4988, pp. 110–122. Springer, Heidelberg (2008)
8. Goranko, V.: The basic algebra of game equivalences. Studia Logica 75, 221–238 (2003)

9. Höfner, P., Struth, G.: Can refinement be automated? ENTCS 201, 197–222 (2007)
10. Kozen, D.: On Hoare logic and Kleene algebra with tests. ACM TOCL 1(1), 60–76 (2000)
11. McIver, A.K., Gonzalia, C., Cohen, E., Morgan, C.C.: Using probabilistic Kleene algebra pKA for protocol verification. J. Logic and Algebraic Programming 76(1), 90–111 (2008)
12. Meinicke, L., Solin, K.: Refinement algebra for probabilistic programs. ENTCS 201, 177–195 (2007)
13. Möller, B.: Kleene getting lazy. Sc. Computer Programming 65(2), 195–214 (2007)
14. Pilz, G.: Near-Rings: The Theory and Its Application. North-Holland, Amsterdam (1983)
15. Solin, K., von Wright, J.: Refinement algebra with operators for enabledness and termination. In: Uustalu, T. (ed.) MPC 2006. LNCS, vol. 4014, pp. 397–415. Springer, Heidelberg (2006)
16. von Wright, J.: Towards a refinement algebra. Sc. Computer Programming 51(1-2), 23–45 (2004)