

Normal Design Algebra

Walter Guttman¹ and Bernhard Möller²

¹Institut für Programmiermethodik und Compilerbau, Universität Ulm, D-89069 Ulm, Germany

²Institut für Informatik, Universität Augsburg, D-86135 Augsburg, Germany

Abstract. We generalise the designs of Unifying Theories of Programming (UTP) by defining them as matrices over semirings with ideals. This clarifies the algebraic structure of designs and considerably simplifies reasoning about them, e.g., forming a Kleene and omega algebra of designs. Moreover, we prove a generalised fixpoint theorem for isotone functions on designs. We apply our framework to investigate symmetric linear recursion and its relation to tail-recursion; this substantially involves Kleene and omega algebra as well as additional algebraic formulations of determinacy, invariants, domain, pre-image, convergence and noetherity. Due to the uncovered algebraic structure of UTP designs, all our general results also directly apply to UTP.

Keywords: UTP; semiring; Kleene algebra; omega algebra; fixpoint; linear recursion

1. Introduction

The Unifying Theories of Programming (UTP), developed in [HH98], model the termination behaviour of programs using two special variables ok and ok' that express whether a program has been started and has terminated. Specifications and programs are identified with predicates relating the initial values v of variables to their final values v' ; moreover, ok and ok' may occur freely in predicates. Using these variables, Hoare and He introduce *designs*, i.e., predicates of the form

$$P \vdash Q \Leftrightarrow_{df} ok \wedge P \Rightarrow ok' \wedge Q,$$

with ok and ok' not occurring in P or Q . The intended use is an assumption/commitment style of specification: if the assumption P holds then every computation admitted by the design will eventually terminate so that the commitment Q holds. In particular, designs reflect a total correctness view.

In the general case, UTP allows the assumption P to involve both the initial and final values of the program variables. A subclass that is interesting for a number of reasons is that of (*H3*) or *normal* designs in which P is a *condition*, i.e., is only allowed to depend on the input values of variables.

In preceding papers [GM06, Möl06] we have presented a general algebraic treatment of designs and of the more liberal predicates known as *prescriptions* [Dun01] that reflect a general correctness view. In particular, these approaches do no longer mention the “unobservable” variables ok and ok' ; in fact they are even completely variable-free and hence do not need to work with substitutions. This makes calculations not

Correspondence and offprint requests to:

Walter Guttman (walter.guttman@uni-ulm.de) and Bernhard Möller (moeller@informatik.uni-augsburg.de)

only simpler, but also safer. Truly hiding the unobservables is important, since their unchecked use can lead to inconsistencies and paradoxes such as the “dead variable paradox” [KP00].

The present paper is devoted to a simpler algebraic framework, that of *ideal semirings*, tailored to the particular case of (normal) designs. While still properly more general than the original, purely relational, UTP semantics, it exhibits the algebraic structure of designs more clearly and allows a much simpler derivation of the basic properties.

Perhaps the most important among these are that the generalised normal designs again form an ideal semiring and even a weak Kleene and omega algebra. Moreover, we show that they can be made into a test algebra and enriched by the modal operators diamond and box. For termination analysis we deal with the convergence and divergence operators that characterise the program states from which, respectively, no and at least one infinite computation is possible. Another result generalises the fixpoint theorem 3.1.6 of [HH98] in various ways.

But the power of the algebra is also demonstrated by a number of investigations on special linear but non-tail recursions. Since these are performed at the more abstract level of Kleene and omega algebras, they are not only valid for UTP but also for many other models.

The structure of this paper and its contributions (beyond the predecessor papers [GM06, Möl06]) are as follows.

In Section 2 we propose axioms for (normal) designs and show that they form a weak Kleene and omega algebra and an ideal semiring. In contrast to previous axiomatisations, the new axioms are based on the established ring-theoretic concept of ideals and perfectly suited for the calculation with designs using their matrix representation. The axioms together with the matrix representation allow a very concise derivation of the Kleene star and omega operations for normal designs. We furthermore generalise Theorem 3.1.6 of [HH98] to our setting and use it to extend star and omega to general designs.

In Section 3, we show how to apply the framework of Kleene algebra and omega algebra to relate tail-recursion to linear recursion and different kinds of linear recursions. By using the model of Section 2, our results are considerably more general than in plain UTP. We furthermore deal with both the least and the greatest fixpoints.

In Section 4, we apply our algebraic techniques to symmetric linear recursion. Our general framework, hence also UTP, is extended by algebraic notions of convergence, determinacy, invariants, domain and pre-image. We show how to implement the recursion’s least and greatest fixpoints each by two consecutive while-loops. We finally discuss axioms for symmetric linear recursion along the lines of the axioms for Kleene/omega algebra.

In Section 5, we use the convergence operator to discuss noetherity. Assuming an atomistic test algebra, we show that progressive boundedness and progressive finiteness coincide for deterministic elements.

2. Star and omega designs

In this section, we derive the Kleene star and omega operations for generalised designs. While the result already appears in [GM06], the present generalisation is based on a modified set of axioms, and the new derivation is considerably shortened by using the matrix representation of [Möl06].

Condition semirings have been introduced in [GM06] to model the essence of normal designs. They are more general than the predicates or relations used in [HH98], that is, they impose fewer axioms. We propose a modified set of axioms that reflects the traditional nomenclature from ring theory, and investigate the connection to the former definition.

We define designs and normal designs as matrices of elements governed by the new axioms and point out why the new axiomatisation is adequate for this purpose.

Since it is well-known how to lift the Kleene star operation to matrix algebras [Koz94], the matrix model also lends itself to deriving the Kleene star operation for normal designs, and, using a similar lifting, also the omega operation.

We finally prove a generalisation of Theorem 3.1.6 of [HH98] that describes the least and greatest fixpoints of functions on general designs. It is applied to derive the star and omega operations for these.

2.1. Axioms for conditions

In [GM06], normal designs have been modelled as commands over condition semirings, adapting the axioms of commands over test semirings studied in [MS06]. In the following, we base our axiomatisation on the established ring-theoretic concept of ideals (generalised to semirings [HW93]).

Definition 2.1.

1. A *weak semiring* is a structure $(S, +, 0, \cdot, 1)$ such that
 - $(S, +, 0)$ is a commutative monoid,
 - $(S, \cdot, 1)$ is a monoid,
 - operation \cdot distributes over $+$ in both arguments
 - and 0 is a left annihilator, i.e., $0 \cdot x = 0$.
2. A weak semiring is *idempotent* if $+$ is, i.e., if $x + x = x$.
3. In an idempotent weak semiring the relation $x \leq y \Leftrightarrow_{df} x + y = y$ is a partial order, called the *natural order* on S .
4. A *semiring* is a weak semiring in which 0 is also a right annihilator, i.e., $x \cdot 0 = 0$.

Mostly, we will notationally suppress the \cdot operation. Moreover, we extend \cdot elementwise to sets $A, B \subseteq S$ by $A \cdot B =_{df} \{a \cdot b \mid a \in A \wedge b \in B\}$ and by $A \cdot b =_{df} A \cdot \{b\}$ for $b \in S$.

In an idempotent weak semiring $+$ can be interpreted as (angelic) choice, with 0 modelling the most partial program with no transition possibilities at all, and \cdot as sequential composition, where 1 models the program skip. The natural order has 0 as its least element. Moreover, $+$ and \cdot are isotone with respect to \leq and $x + y$ is the least upper bound or join of x and y with respect to \leq .

To model the assumption parts of normal designs, we need special semiring elements that play the role of conditions. To this end, let us list some properties that are typical of conditions in the relational calculus:

1. The sequential composition of an arbitrary relation and a condition yields a condition again.
2. An arbitrary relation is input-restricted by conjoining it, i.e., forming its meet, with a condition.
3. This restriction distributes over union in both arguments.
4. Restriction by the universal condition is no restriction at all.
5. Conditions can be used for Boolean reasoning, since they form a Boolean algebra.
6. Conditions are right-universal relations, equivalently, are invariant under post-composition with the universal relation.

It turns out that these properties are sufficient for an abstract axiomatisation of conditions. Since for many results already properties 1–5 suffice, we first deal only with these; property 6 will be added later. Property 1 can be rephrased by saying that the set of all conditions is a left ideal of the semiring under consideration. This motivates the following definition.

Definition 2.2. A structure $(S, T, +, 0, \cdot, 1, \top, \wedge, \bar{})$ is an *ideal semiring* iff the following properties hold.

- $(S, +, 0, \cdot, 1)$ is an idempotent weak semiring with greatest element \top .
- T is a left ideal of S , i.e.,
 - * $(T, +, 0)$ is a sub-monoid of $(S, +, 0)$ and
 - * $S \cdot T \subseteq T$.
- The *restriction operation* $\wedge : T \times S \rightarrow S$ distributes over $+$, i.e.,
 - * $\forall a \in S : \forall t, u \in T : (t + u) \wedge a = (t \wedge a) + (u \wedge a)$ and
 - * $\forall a, b \in S : \forall t \in T : t \wedge (a + b) = (t \wedge a) + (t \wedge b)$.
- $\forall a \in S : \top \wedge a = a$.
- $(T, +, 0, \wedge, \top, \bar{})$ is a Boolean algebra.

In the remainder we abbreviate ideal semiring structures to (S, T) . An ideal semiring is *strict* if the underlying weak semiring is a semiring, i.e., if 0 is a right annihilator: $x \cdot 0 = 0$.

Over an ideal semiring, the assumption part of a design will be taken from the set T , and its commitment part from the set S .

The following lemma gives a few properties of restriction and shows that $t \wedge a$ is the meet of t and a .

Lemma 2.3. In an ideal semiring (S, T) ,

1. \wedge is isotone in both arguments,
2. \wedge is the greatest lower bound operation on $T \times S$, and
3. the shunting rule $t \wedge a \leq b \Leftrightarrow a \leq \bar{t} + b$ holds.

Proof. Let $a, b \in S$ and $t \in T$.

1. Immediate from the distributivity axioms of \wedge .
2. $- t \wedge a \leq t \wedge \top = t$ by part 1 and Boolean algebra.
 $- t \wedge a \leq \top \wedge a = a$ by part 1 and an axiom.
 $-$ If $b \leq t$ and $b \leq a$, then, by part 1, $b = \top \wedge b = (\bar{t} + t) \wedge b = \bar{t} \wedge b + t \wedge b \leq \bar{t} \wedge t + t \wedge a = t \wedge a$.
3. $(\Rightarrow) a = (\bar{t} + t) \wedge a = \bar{t} \wedge a + t \wedge a \leq \bar{t} + b$.
 $(\Leftarrow) t \wedge a \leq t \wedge (\bar{t} + b) = t \wedge \bar{t} + t \wedge b \leq b$. □

Consider an ideal semiring (S, T) . Since T is a left ideal of S , it follows that T is also a sub-semiring (without 1) of S . But T has another semiring structure, by virtue of being a Boolean algebra, with $+$ as addition and \wedge as composition. We can therefore relate T to the concept of a module from ring theory [HW93].

Lemma 2.4. In an ideal semiring (S, T) , S is a unitary left T -semimodule with scalar multiplication \wedge .

Proof. Let $a \in S$ and $t, u \in T$. It remains to show the associative law $(t \wedge u) \wedge a = t \wedge (u \wedge a)$, since the distributive and unitary laws are already axioms. The proof uses Lemma 2.3 several times.

- $- (t \wedge u) \wedge a \leq t \wedge u \leq t$ and $(t \wedge u) \wedge a \leq u \wedge a$, hence $(t \wedge u) \wedge a \leq t \wedge (u \wedge a)$.
- $- t \wedge (u \wedge a) \leq t \wedge u$ and $t \wedge (u \wedge a) \leq u \wedge a \leq a$, hence $t \wedge (u \wedge a) \leq (t \wedge u) \wedge a$. □

Thus, the elements T of an ideal semiring (S, T) can be seen as acting on the elements of the semiring S . At the same time the associativity law is typical of a restriction operation. As a consequence of Lemma 2.4, (S, T) may be characterised by stating that T is a Boolean algebra and a left ideal of S , and S is a unitary left T -semimodule, all with respect to the appropriate operations.

Ideal semirings are sufficient for representing designs, whereas for normal designs we need the subclass of ideal condition semirings. As stated above, conditions t are invariant under post-composition with the universal element \top , i.e., $t \cdot \top = t$. As will be shown below, it suffices to require that $t = u \cdot \top$ for some $u \in T$. This motivates the following definition.

Definition 2.5. An *ideal condition semiring* is an ideal semiring (S, T) where additionally $T \subseteq T \cdot \top$ holds. In this case the elements of T are called *conditions*. A semiring S with greatest element \top is *replete* if $S \cdot \top$ is a Boolean algebra.

It is easy to see that the set $S \cdot \top \subseteq T$ consists of all elements that are invariant under right composition with \top . If $S \cdot \top$ is a Boolean algebra, it could be smaller than T , and certainly is another candidate for the condition set of an ideal semiring over S . We show below that it actually coincides with the full set T of conditions, hence the name *replete*.

Since in the literature the elements of $S \cdot \top$ are sometimes known as right ideals, repleteness was termed ideal-closedness in [GM06], a terminology which does no longer fit with the use of the term *ideal* in the present paper.

Lemma 2.6. Let (S, T) be an ideal condition semiring.

1. $S \cdot \top = T$; hence S is replete.
2. $\forall t \in T : t \cdot \top = t$.
3. $\forall t \in T : \forall a, b \in S : (t \wedge a) \cdot b = t \wedge (a \cdot b)$.

Proof. Let $a, b \in S$ and $t \in T$.

1. $S \cdot \top \subseteq S \cdot T \subseteq T \subseteq T \cdot \top \subseteq S \cdot \top$, hence $S \cdot \top = T$ is a Boolean algebra.

2. Since $T \subseteq T \cdot \top$, there is a $u \in T$ such that $t = u \cdot \top$. As a consequence, $t \cdot \top = u \cdot \top \cdot \top = u \cdot \top = t$.
3. (\leq) By isotony, $(t \wedge a) \cdot b \leq a \cdot b$ and $(t \wedge a) \cdot b \leq t \cdot b \leq t \cdot \top = t$, using part 2.
 (\geq) By Boolean algebra and (\leq), $t \wedge (a \cdot b) = t \wedge ((t \wedge a) \cdot b + (\bar{t} \wedge a) \cdot b) \leq t \wedge ((t \wedge a) \cdot b + \bar{t} \wedge (a \cdot b)) = t \wedge ((t \wedge a) \cdot b) \leq (t \wedge a) \cdot b$. \square

Therefore, and in contrast to condition semirings, every ideal condition semiring is already replete. The cause for this restriction is the axiom $S \cdot T \subseteq T$, since the other prerequisites of Lemma 2.6 also hold in condition semirings. As we will point out in Section 2.2, however, this axiom is necessary for the representation of normal designs as matrices and, subsequently, to obtain a Kleene and omega algebra. Nevertheless, the new axioms are less restrictive than those in [Möl06], since they do not require S to be a Boolean semiring.

2.2. Designs and normal designs as matrices

We define (normal) designs as 2×2 matrices over a weak semiring, similarly to [Möl06]. The difference is that we do not demand a Boolean semiring, but take the elements from an ideal semiring.

Let us repeat the basic motivation. The main aim with the matrix representation is to get rid of explicit uses of the special variables ok and ok' . This can be achieved by recording, for each combination of possible values of these two variables, the residual predicate that depends only on the proper program variables. To this end, a UTP predicate $R(ok, ok')$ is represented as the 2×2 matrix

$$R = \begin{pmatrix} R(false, false) & R(false, true) \\ R(true, false) & R(true, true) \end{pmatrix}.$$

The advantage of the matrix representation is that all operations on predicates can now be performed as standard matrix operations and hence reasoned about in a completely component-free manner; ok and ok' need not be mentioned and variable substitutions disappear.

In [HH98], designs are characterised by the healthiness conditions (H1) and (H2). A predicate R satisfies (H1) iff R is *true* whenever ok is *false*; this means that the top row of R 's matrix is constantly *true*. It satisfies (H2) iff the rows of its matrix are increasing in the implication order.

This motivates our definition of designs in the abstract setting, where we use 2×2 matrices with elements from an ideal semiring S as entries.

Definition 2.7. Let (S, T) be an ideal semiring. The set of *designs* over (S, T) is

$$D(S, T) =_{df} \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in S^{2 \times 2} \mid a = b = \top \wedge c \in T \wedge c \leq d \right\}.$$

For $t \in T$ and $a \in S$, we define the *design*

$$t \vdash a =_{df} \begin{pmatrix} \top & \top \\ \bar{t} & \bar{t} + a \end{pmatrix}.$$

If, additionally, (S, T) is an ideal condition semiring, we call its designs *normal* and set $ND(S, T) =_{df} D(S, T)$.

From these definitions it follows that every design can be denoted in abbreviated form, since

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in D(S, T) \Rightarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \bar{c} \vdash d.$$

Using matrix addition and multiplication, we can lift the semiring structure to normal designs as shown by the following lemma. While this is clear for the matrix semiring $S^{2 \times 2}$, we have to check the restrictions imposed by designs. Observe that the left ideal property is crucial for the totality of \cdot and that a strict ideal condition semiring is needed for the right unit law.

Lemma 2.8. Let (S, T) be a strict ideal condition semiring. Then the structure $(ND(S, T), +, \top \vdash 0, \cdot, \top \vdash 1)$ is an idempotent weak semiring.

Proof. Let $a, b \in S$ and $t, u \in T$ such that $t \leq a$ and $u \leq b$.

– $+$ is total since $t + u \in T$ and $t + u \leq a + b$ and

$$\begin{pmatrix} \top & \top \\ t & a \end{pmatrix} + \begin{pmatrix} \top & \top \\ u & b \end{pmatrix} = \begin{pmatrix} \top & \top \\ t+u & a+b \end{pmatrix}.$$

– $+$ is associative, commutative and idempotent since it is defined componentwise.

– $\top \vdash 0$ is neutral with respect to $+$ since

$$\begin{pmatrix} \top & \top \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} = \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} = \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} + \begin{pmatrix} \top & \top \\ 0 & 0 \end{pmatrix}.$$

– \cdot is total since $t\top + au = t + au \in T$ and $t + au \leq t + ab$ and

$$\begin{pmatrix} \top & \top \\ t & a \end{pmatrix} \cdot \begin{pmatrix} \top & \top \\ u & b \end{pmatrix} = \begin{pmatrix} \top\top + \top u & \top\top + \top b \\ t\top + au & t\top + ab \end{pmatrix} = \begin{pmatrix} \top & \top \\ t+au & t+ab \end{pmatrix}.$$

– \cdot is associative since matrix multiplication is associative.

– $\top \vdash 1$ is neutral with respect to \cdot since

$$\begin{pmatrix} \top & \top \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} = \begin{pmatrix} \top & \top \\ 0+1t & 0+1a \end{pmatrix} = \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} = \begin{pmatrix} \top & \top \\ t+a0 & t+a1 \end{pmatrix} = \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} \cdot \begin{pmatrix} \top & \top \\ 0 & 1 \end{pmatrix}.$$

– \cdot distributes over $+$ since it does so for matrices.

– $\top \vdash 0$ is a left annihilator of \cdot since

$$\begin{pmatrix} \top & \top \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} = \begin{pmatrix} \top & \top \\ 0+0t & 0+0a \end{pmatrix} = \begin{pmatrix} \top & \top \\ 0 & 0 \end{pmatrix}. \quad \square$$

With two modifications, Lemma 2.8 generalises to designs over ideal semirings: The composition of designs is the more verbose

$$\begin{pmatrix} \top & \top \\ t & a \end{pmatrix} \cdot \begin{pmatrix} \top & \top \\ u & b \end{pmatrix} = \begin{pmatrix} \top & \top \\ t\top + au & t\top + ab \end{pmatrix},$$

and the right unit law fails.

As a consequence of Lemma 2.8, we obtain that normal designs behave just as expected from [HH98], also in their abbreviated forms. Additionally, the natural order of normal designs also makes sense for designs.

Corollary 2.9. The natural order of (normal) designs is

$$t \vdash a \leq u \vdash b \Leftrightarrow u \leq t \wedge u \wedge a \leq b \Leftrightarrow u \leq t \wedge a \leq \bar{u} + b.$$

Moreover,

$$t \vdash a = u \vdash b \Leftrightarrow t = u \wedge t \wedge a = u \wedge b$$

and

$$t \vdash a = t \vdash \bar{t} + a = t \vdash t \wedge a.$$

The composition of designs is $(t \vdash a) \cdot (u \vdash b) = \overline{\bar{t}\top + a\bar{u}} \vdash ab$ which simplifies to $t \wedge \bar{a}\bar{u} \vdash ab$ for normal designs.

Proof. Let $a, b \in S$ and $t, u \in T$.

– By the componentwise matrix order and the shunting rule of Lemma 2.3.3,

$$\begin{aligned} t \vdash a \leq u \vdash b &\Leftrightarrow \begin{pmatrix} \top & \top \\ \bar{t} & \bar{t} + a \end{pmatrix} \leq \begin{pmatrix} \top & \top \\ \bar{u} & \bar{u} + b \end{pmatrix} \Leftrightarrow \bar{t} \leq \bar{u} \wedge \bar{t} + a \leq \bar{u} + b \\ &\Leftrightarrow \bar{t} \leq \bar{u} \wedge \bar{t} \leq \bar{u} + b \wedge a \leq \bar{u} + b \Leftrightarrow u \leq t \wedge a \leq \bar{u} + b \Leftrightarrow u \leq t \wedge u \wedge a \leq b. \end{aligned}$$

Therefore,

$$\begin{aligned} t \vdash a = u \vdash b &\Leftrightarrow t \vdash a \leq u \vdash b \wedge u \vdash b \leq t \vdash a \Leftrightarrow u \leq t \wedge u \wedge a \leq b \wedge t \leq u \wedge t \wedge b \leq a \\ &\Leftrightarrow t = u \wedge t \wedge a = u \wedge b. \end{aligned}$$

From this $t \vdash a = t \vdash \bar{t} + a = t \vdash t \wedge a$ follows immediately.

– The composition of designs is given by

$$\begin{aligned} (t \vdash a) \cdot (u \vdash b) &= \begin{pmatrix} \top & \top \\ \bar{t} & \bar{t} + a \end{pmatrix} \cdot \begin{pmatrix} \top & \top \\ \bar{u} & \bar{u} + b \end{pmatrix} = \begin{pmatrix} \top & \top \\ \bar{t}\top + (\bar{t} + a)\bar{u} & \bar{t}\top + (\bar{t} + a)(\bar{u} + b) \end{pmatrix} \\ &= \begin{pmatrix} \top & \top \\ \bar{t}\top + a\bar{u} & \bar{t}\top + a\bar{u} + ab \end{pmatrix} = \overline{\bar{t}\top + a\bar{u}} \vdash ab, \end{aligned}$$

and $\overline{\bar{t}\top + a\bar{u}} = t \wedge \overline{a\bar{u}}$ for normal designs. \square

Note that the natural order reflects the implication order on designs, not the refinement order of [HH98], which is the reverse.

For the remainder of this section and the two following ones, we restrict our attention to normal designs, assuming a strict ideal condition semiring. In Section 2.6 we return to the more general case of designs over an ideal semiring.

We can also lift the ideal condition semiring structure to normal designs, which will be useful to represent UTP-conditions as tests in Section 3. Let (S, T) be a strict ideal condition semiring. In Lemma 2.8 we have already shown that $\mathcal{S} =_{df} \text{ND}(S, T)$ forms an idempotent weak semiring and $0 \vdash 0$ clearly is its greatest element. We define its condition subset as

$$\mathcal{T} =_{df} \{t \vdash 0 \mid t \in T\} = \left\{ \begin{pmatrix} \top & \top \\ t & t \end{pmatrix} \mid t \in T \right\}.$$

It is easily calculated that \mathcal{T} is a sub-monoid of \mathcal{S} , and the left ideal property follows since

$$(t \vdash a) \cdot (u \vdash 0) = t \wedge \overline{a\bar{u}} \vdash a0 = t \wedge \overline{a\bar{u}} \vdash 0.$$

As a special case, we obtain the condition property $(t \vdash 0) \cdot (0 \vdash 0) = t \vdash 0$.

We define the restriction \wedge as the componentwise restriction on the matrix representation:

$$\begin{pmatrix} \top & \top \\ t & t \end{pmatrix} \wedge \begin{pmatrix} \top & \top \\ u & b \end{pmatrix} =_{df} \begin{pmatrix} \top & \top \\ t \wedge u & t \wedge b \end{pmatrix}.$$

Immediate consequences are distributivity over $+$ and neutrality of the universal condition. The Boolean algebra structure also follows using the complement

$$\overline{\begin{pmatrix} \top & \top \\ t & t \end{pmatrix}} =_{df} \begin{pmatrix} \top & \top \\ \bar{t} & \bar{t} \end{pmatrix}.$$

We thus obtain:

Proposition 2.10. If (S, T) is a strict ideal condition semiring then $(\mathcal{S}, \mathcal{T})$ is an ideal condition semiring.

2.3. Star designs

We show that normal designs form a weak Kleene algebra. The Kleene star operation is useful to give closed representations of finite iterations and simplifies calculations involving such iterations. It will be used in Sections 3 and 4 for this purpose. That normal designs have a star means that the results of these sections are applicable to UTP.

Definition 2.11. A *Kleene algebra* is a structure $(S, *)$ such that S is an idempotent semiring and the operation star $*$ satisfies the unfold and induction laws

$$\begin{aligned} 1 + a \cdot a^* &\leq a^* & 1 + a^* \cdot a &\leq a^* \\ b + a \cdot c \leq c &\Rightarrow a^* \cdot b \leq c & b + c \cdot a \leq c &\Rightarrow b \cdot a^* \leq c \end{aligned}$$

for $a, b, c \in S$ [Koz94]. In a *weak* Kleene algebra, S is only required to be an idempotent weak semiring.

Hence a^*b is the least fixpoint of the mapping $\lambda x. ax + b$.

The star operation can be lifted to matrices by a standard construction. We take the version presented in [ÉL05], a similar construction appears in [Koz94].

Definition 2.12. The *Kleene star* of a 2×2 matrix is given by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^* =_{df} \begin{pmatrix} f^* & f^*bd^* \\ e^*ca^* & e^* \end{pmatrix},$$

where $f = a + bd^*c$ and $e = d + ca^*b$.

The Kleene star of a normal design hence is

$$\begin{pmatrix} \top & \top \\ t & a \end{pmatrix}^* = \begin{pmatrix} \top^* & \top a^* \\ a^*t\top & a^* \end{pmatrix} = \begin{pmatrix} \top & \top \\ a^*t & a^* \end{pmatrix},$$

since $f = \top + \top a^*t = \top$ and $e = a + t\top^*\top = a + t = a$ and $\top a^* \geq \top 1 = \top$. The result is a normal design since $t \in T \Rightarrow a^*t \in T$ and $t \leq a \Rightarrow a^*t \leq a^*a \leq a^*$. Observe that the left ideal property is crucial again. We therefore have the following lemma.

Proposition 2.13. Let (S, T) be an ideal condition semiring such that S is a Kleene algebra. Then the structure $(\text{ND}(S, T), +, \top \vdash 0, \cdot, \top \vdash 1, *)$ is a weak Kleene algebra.

Proof. After Lemma 2.8, it remains to show that the star unfold and induction axioms are satisfied. But this follows, since they are valid in the encompassing full matrix algebra and $\text{ND}(S, T)$ is closed under star. \square

We finally derive the Kleene star of a normal design in the abbreviated representation used by [HH98]. As a prerequisite we prove a lemma concerning the Kleene star in an ideal condition semiring.

Lemma 2.14. Consider an ideal condition semiring (S, T) such that S is a Kleene algebra and let $a \in S$ and $t \in T$.

1. $(t + a)^* = a^*t + a^*$.
2. $(t + a)^*t = a^*t$.

Proof.

1. First, we have $1 + (t + a)(a^*t + a^*) \leq 1 + t\top + a(a^*t + a^*) = t + aa^*t + a^* = a^*t + a^*$, and therefore $(t + a)^* \leq a^*t + a^*$ by star induction. Second, $a^*t + a^* \leq (t + a)^*(t + a) + (t + a)^* \leq (t + a)^*$.
2. By part 1, we have $(t + a)^*t = a^*tt + a^*t \leq a^*t\top + a^*t = a^*t$. The converse, $a^*t \leq (t + a)^*t$, follows by isotony of star. \square

Theorem 2.15. Let (S, T) be an ideal condition semiring such that S is a Kleene algebra. Let $t \vdash a$ be a normal design over (S, T) , then $(t \vdash a)^* = (\overline{a^*t} \vdash a^*)$.

Proof. By Lemma 2.14,

$$(t \vdash a)^* = \begin{pmatrix} \top & \top \\ \bar{t} & \bar{t} + a \end{pmatrix}^* = \begin{pmatrix} \top & \top \\ (\bar{t} + a)^*\bar{t} & (\bar{t} + a)^* \end{pmatrix} = \begin{pmatrix} \top & \top \\ a^*\bar{t} & a^*\bar{t} + a^* \end{pmatrix} = \overline{a^*t} \vdash a^*. \quad \square$$

2.4. Omega designs

We show that normal designs form a weak omega algebra. The omega operation together with the Kleene star allows a closed representation of infinite iterations and of certain greatest fixpoints. Corresponding results of Sections 3 and 4 are applicable to UTP due to the fact that normal designs have an omega.

Definition 2.16. A *weak omega algebra* is a structure (S, ω) such that S is a weak Kleene algebra and the omega ω satisfies the unfold and co-induction laws

$$\begin{aligned} a^\omega &= a \cdot a^\omega \\ c \leq a \cdot c + b &\Rightarrow c \leq a^\omega + a^* \cdot b \end{aligned}$$

for $a, b, c \in S$ [Möl04].

It follows that $a^\omega + a^*b$ is the greatest fixpoint of the mapping $\lambda x.ax + b$.

In contrast to this definition, an *omega algebra* requires S to be a Kleene algebra but weakens the unfold axiom to $a^\omega \leq a \cdot a^\omega$ [Coh00]. The reverse inequality need not hold in absence of the right annihilation axiom [Möl04].

The omega operation can be lifted to matrices by another construction, presented in [MD06].

Definition 2.17. The *omega* of a 2×2 matrix is given by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^\omega =_{df} \begin{pmatrix} f^\omega + a^*be^\omega & f^\omega + a^*be^\omega \\ d^*cf^\omega + e^\omega & d^*cf^\omega + e^\omega \end{pmatrix},$$

where $f = a + bd^*c$ and $e = d + ca^*b$.

The omega of a normal design hence is

$$\begin{pmatrix} \top & \top \\ t & a \end{pmatrix}^\omega = \begin{pmatrix} \top^\omega + \top a^\omega & \top^\omega + \top a^\omega \\ a^*t\top^\omega + a^\omega & a^*t\top^\omega + a^\omega \end{pmatrix} = \begin{pmatrix} \top & \top \\ a^\omega + a^*t & a^\omega + a^*t \end{pmatrix},$$

since, as before, $f = \top$ and $e = a$. The result is a normal design since $t \in T \Rightarrow a^\omega + a^*t = a^\omega\top + a^*t \in T$. Observe that the left ideal property is crucial again. We therefore have the following lemma.

Proposition 2.18. Let (S, T) be an ideal condition semiring such that S is an omega algebra. Then the structure $(\text{ND}(S, T), +, \top \vdash 0, \cdot, \top \vdash 1, *, \omega)$ is a weak omega algebra.

Proof. After Proposition 2.13, it remains to show that the omega co-induction and unfold axioms are satisfied. But this follows, since they are valid in the encompassing full matrix algebra and $\text{ND}(S, T)$ is closed under omega. \square

We finally derive the omega of a normal design in the abbreviated representation used by [HH98]. As a prerequisite we prove a lemma concerning the omega operation in an ideal condition semiring.

Lemma 2.19. Consider an ideal condition semiring (S, T) such that S is an omega algebra and let $a \in S$ and $t \in T$. Then, $(t + a)^\omega + (t + a)^*t = a^\omega + a^*t$.

Proof. (\geq) is immediate by isotony. For (\leq) , after application of Lemma 2.14.2 it suffices to show $(t + a)^\omega \leq a^\omega + a^*t$. But this follows by omega co-induction from $(t + a)^\omega \leq (t + a)(t + a)^\omega \leq t\top + a(t + a)^\omega = t + a(t + a)^\omega$. \square

Theorem 2.20. Let (S, T) be an ideal condition semiring such that S is an omega algebra. Let $t \vdash a$ be a normal design over (S, T) , then $(t \vdash a)^\omega = \overline{(a^\omega + a^*t \vdash 0)}$.

Proof. By Lemma 2.19,

$$\begin{pmatrix} \top & \top \\ \bar{t} & \bar{t} + a \end{pmatrix}^\omega = \begin{pmatrix} \top & \top \\ (\bar{t} + a)^\omega + (\bar{t} + a)^*\bar{t} & (\bar{t} + a)^\omega + (\bar{t} + a)^*\bar{t} \end{pmatrix} = \begin{pmatrix} \top & \top \\ a^\omega + a^*\bar{t} & a^\omega + a^*\bar{t} \end{pmatrix}. \quad \square$$

2.5. UTP algebras

As we have seen in the previous sections, normal designs form an idempotent weak semiring, a weak Kleene algebra and a weak omega algebra. The qualifier “weak” states that the right annihilation axiom $\forall x : x \cdot 0 = 0$ is not required to hold. In a semiring with greatest element \top this axiom may be restated as $\top \cdot 0 = 0$ by isotony.

As stated just after Corollary 2.9, (normal) designs have the greatest element $0 \vdash 0$, also called *true* in [HH98]. It is easily verified that the right annihilation law does not hold for (normal) designs; indeed $\top_{\text{ND}(S, T)} \cdot 0_{\text{ND}(S, T)} = \top_{\text{ND}(S, T)}$, since

$$(0 \vdash 0) \cdot (\top \vdash 0) = \overline{0 \cdot \top + 0 \cdot \top} \vdash 0 \cdot 0 = 0 \vdash 0.$$

Omitting the right annihilation axiom gives us the freedom to impose this left annihilation axiom instead.

Definition 2.21. A *UTP semiring/Kleene algebra/omega algebra* is a weak semiring/Kleene algebra/omega algebra with greatest element \top such that

$$\top \cdot 0 = \top \tag{1}$$

or, equivalently, $\forall x : \top \cdot x = \top$.

An immediate consequence is that normal designs form a UTP omega algebra.

The axiom $\top \cdot 0 = \top$ is typical of total correctness frameworks, not only of UTP (see, e.g., also the demonic refinement algebra of [Wri04]).

Once \top is a left annihilator, further elements are, too.

Lemma 2.22.

1. Let a be an element of a UTP omega algebra, then a^ω is a left annihilator.
2. Let (S, T) be a strict ideal condition semiring and $a \in S$ and $t \in T$ such that $a \leq \bar{t}$. Then $t \vdash a$ is a left annihilator. In particular, $t \vdash \bar{t}$ is a left annihilator.

Proof.

1. $a^\omega x = a^\omega \top x = a^\omega \top = a^\omega$.
2. By Corollary 2.9, $t \vdash a = t \vdash t \wedge a = t \vdash 0$, and $(t \vdash 0) \cdot (u \vdash b) = t \wedge \overline{0u} \vdash 0b = t \vdash 0$. □

2.6. Fixpoints and designs

In Sections 2.3 and 2.4 we have shown how to calculate the least and the greatest fixpoints of the iteration function $\lambda x. ax + b$ on normal designs. We now use our algebraic techniques to extend this by considering all designs instead of just normal designs, and by investigating fixpoints of the more general function

$$H(P \vdash Q) =_{df} F(P \vdash Q) \vdash G(P \vdash Q)$$

on designs. Its greatest fixpoint is described by Theorem 3.1.6 of [HH98]. We generalise that result in two respects:

1. We do not assume that F is isotone in P and antitone in Q and the other way around for G . Instead, we only require that H is isotone as a whole.

Let us discuss why this is more general. First, we show that Hoare and He's requirement implies that H is isotone. Let F be isotone in P and antitone in Q and let G be antitone in P and isotone in Q . Assume that $P_1 \vdash Q_1 \leq P_2 \vdash Q_2$, i.e., by Corollary 2.9, $P_2 \leq P_1$ and $Q_1 \leq \overline{P_2} + Q_2$. By anti/isotony of F/G this implies

$$\begin{aligned} F(P_2 \vdash Q_2) &= F(P_2 \vdash \overline{P_2} + Q_2) \leq F(P_1 \vdash Q_1), \\ G(P_1 \vdash Q_1) &\leq G(P_2 \vdash \overline{P_2} + Q_2) = G(P_2 \vdash Q_2). \end{aligned}$$

Hence, a fortiori, $F(P_2 \vdash Q_2) \wedge G(P_1 \vdash Q_1) \leq G(P_2 \vdash Q_2)$. Now Corollary 2.9 shows $H(P_1 \vdash Q_1) \leq H(P_2 \vdash Q_2)$. It follows that H is isotone.

Second, we show that the converse does not hold. Let $F(P \vdash Q) = 0$ and $G(P \vdash Q) = \overline{(P + Q)\top}$, then

$$H(P \vdash Q) = F(P \vdash Q) \vdash G(P \vdash Q) = 0 \vdash G(P \vdash Q) = 0 \vdash 0.$$

Being a constant function, H is isotone. However, G is neither antitone in P nor isotone in Q ; indeed it is the other way around.

We will shortly see in Lemma 2.24 that the requirement on F cannot be dispensed with.

2. We do not assume that P and Q are relations. Instead, we only require that $P \vdash Q$ is a design over an ideal semiring and that certain fixpoints exist.

In our treatment we assume the following axioms for fixpoints (see, e.g., [DMT06]).

Definition 2.23. Let f be a function on a partial order. An element a is a *fixpoint* of f if it satisfies the fixpoint law $f(a) = a$. The element μf is the *least prefixpoint* of f if the following unfold and induction laws hold:

$$f(\mu f) \leq \mu f, \quad \forall x : f(x) \leq x \Rightarrow \mu f \leq x.$$

The element νf is the *greatest postfixpoint* of f if the following unfold and co-induction laws hold:

$$\nu f \leq f(\nu f), \quad \forall x : x \leq f(x) \Rightarrow x \leq \nu f.$$

If f is isotone it follows that μf is the least fixpoint of f , and νf the greatest. If f and g are isotone and $f \leq g$ it follows that $\mu f \leq \mu g$ and $\nu f \leq \nu g$. We abbreviate $\mu(\lambda x.f(x))$ by $\mu x.f(x)$. If the partial order is a Boolean algebra, $\mu f = \neg \nu x.\neg f(\neg x)$ and $\nu f = \neg \mu x.\neg f(\neg x)$.

2.6.1. The greatest fixpoint

In the following, let $H(t \vdash a) =_{df} F(t \vdash a) \vdash G(t \vdash a)$ be an isotone mapping of designs over an ideal semiring (S, T) such that $F(t \vdash a) \in T$ for all $t \in T$ and $a \in S$.

Lemma 2.24. Let $a, b \in S$ and $t, u \in T$ such that $a \leq b$ and $u \leq t$. Then $F(u \vdash b) \leq F(t \vdash a)$ and $F(u \vdash b) \wedge G(t \vdash a) \leq G(u \vdash b)$. In particular, $\lambda t.F(t \vdash a)$ is isotone and $\lambda a.F(t \vdash a)$ is antitone.

Proof. $a \leq b$ implies $u \wedge a \leq b$. Hence the assumptions entail $t \vdash a \leq u \vdash b$. Since H is isotone, we conclude $F(t \vdash a) \vdash G(t \vdash a) \leq F(u \vdash b) \vdash G(u \vdash b)$ which, by Corollary 2.9, is equivalent to the claim. \square

The following definitions of P , R and Q are based on [HH98] and assume that certain fixpoints exist. Note that μ and ν are swapped relative to [HH98], since we use the implication order and not the refinement order. We first prove some further isotony statements.

Definition 2.25. Define $P : S \rightarrow T$ by $P(a) =_{df} \mu t.F(t \vdash a)$. We assume that $\mu t.F(t \vdash a)$ exists; if T is complete, this is guaranteed by Lemma 2.24.

Lemma 2.26. P is antitone.

Proof. Assume $a \leq b$. By Lemma 2.24, $\lambda t.F(t \vdash a) \geq \lambda t.F(t \vdash b)$, hence $\mu t.F(t \vdash a) \geq \mu t.F(t \vdash b)$ by the remark following Definition 2.23. \square

Definition 2.27. Define $R : S \rightarrow S$ by $R(a) =_{df} \overline{P(a)} + G(P(a) \vdash a)$.

Lemma 2.28. R is isotone.

Proof. Let $a, b \in S$ such that $a \leq b$. By Lemma 2.26, we have $P(b) \leq P(a)$. Now Lemma 2.24 shows $F(P(b) \vdash b) \wedge G(P(a) \vdash a) \leq G(P(b) \vdash b)$. By Definition 2.25, $F(P(b) \vdash b) = P(b)$ and shunting shows $G(P(a) \vdash a) \leq \overline{P(b)} + G(P(b) \vdash b) = R(b)$. Since $\overline{P(a)} \leq \overline{P(b)} \leq R(b)$, we have $R(a) \leq R(b)$. \square

Definition 2.29. Define $Q =_{df} \nu R$. We assume that νR exists; if S is complete, this is guaranteed by Lemma 2.28.

We are now ready to generalise Theorem 3.1.6 of [HH98] in our setting.

Theorem 2.30. $\nu H = P(Q) \vdash Q$.

Proof. First, we prove that $P(Q) \vdash Q$ is a fixpoint of H . By Definition 2.25, $P(Q) = F(P(Q) \vdash Q)$. Hence,

$$\begin{aligned} H(P(Q) \vdash Q) &= F(P(Q) \vdash Q) \vdash G(P(Q) \vdash Q) = P(Q) \vdash G(P(Q) \vdash Q) \\ &= P(Q) \vdash \overline{P(Q)} + G(P(Q) \vdash Q) = P(Q) \vdash R(Q) = P(Q) \vdash Q. \end{aligned}$$

Second, we prove that $P(Q) \vdash Q$ is the greatest postfixpoint of H . Assume $t \vdash a \leq H(t \vdash a)$ which by Corollary 2.9 is equivalent to

$$F(t \vdash a) \leq t \text{ and } F(t \vdash a) \wedge a \leq G(t \vdash a). \quad (\star)$$

Hence, by Definition 2.25, $P(a) \leq t$ and therefore also $P(a) = F(P(a) \vdash a) \leq F(t \vdash a)$ by Lemma 2.24. This and (\star) imply $P(a) \wedge a \leq G(t \vdash a)$, and therefore $P(a) \wedge a \leq F(P(a) \vdash a) \wedge G(t \vdash a) \leq G(P(a) \vdash a)$ by Lemma 2.24. By shunting, $a \leq \overline{P(a)} + G(P(a) \vdash a) = R(a)$, hence $a \leq Q$ by Definition 2.29. Therefore, $P(Q) \wedge a \leq Q$, and, by Lemma 2.26, $P(Q) \leq P(a) \leq t$. Altogether $t \vdash a \leq P(Q) \vdash Q$. \square

In Section 2.4 we have derived the omega operation on normal designs. As an example of using Theorem 2.30, let us now characterise the omega operation on designs.

Corollary 2.31. Let $t \vdash a$ be given and set $H(u \vdash b) =_{df} (t \vdash a)(u \vdash b)$. Then $\nu H = \overline{a^\omega + a^* \bar{t} \top} \vdash 0$.

Proof. Observe that H is isotone and, by Corollary 2.9,

$$H(u \vdash b) = F(u \vdash b) \vdash G(u \vdash b),$$

where $F(u \vdash b) =_{df} \overline{\bar{t} \top + a\bar{u}}$ and $G(u \vdash b) =_{df} \bar{t} \top + a\bar{u} + ab$. By Definition 2.25,

$$P(b) = \mu u. F(u \vdash b) = \mu u. \overline{\bar{t} \top + a\bar{u}} = \overline{\nu u. \bar{t} \top + au} = \overline{a^\omega + a^* \bar{t} \top}.$$

Since $P(b)$ is constant, let $P = P(b)$. By Definitions 2.27 and 2.29, as well as omega and star properties,

$$\begin{aligned} Q &= \nu b. \overline{P(b)} + G(P(b) \vdash b) = \nu b. \bar{P} + \bar{t} \top + a\bar{P} + ab = a^\omega + a^* (\bar{P} + \bar{t} \top + a\bar{P}) = a^\omega + a^* \bar{P} + a^* \bar{t} \top \\ &= a^\omega + a^* (a^\omega + a^* \bar{t} \top) + a^* \bar{t} \top = a^\omega + a^* \bar{t} \top = \bar{P}. \end{aligned}$$

By Theorem 2.30, $\nu H = P(Q) \vdash Q = P \vdash \bar{P} = P \vdash 0$. \square

2.6.2. The least fixpoint

The least fixpoint of a function on designs can be calculated in a similar way. To this end, we swap μ and ν in the definitions of P and Q , so that $P(a) =_{df} \nu t. F(t \vdash a)$ and $Q =_{df} \mu R$. Lemmas 2.24, 2.26 and 2.28 and their proofs remain unchanged. We only need to restate the main theorem:

Theorem 2.32. $\mu H = P(Q) \vdash Q$.

Proof. The proof that $P(Q) \vdash Q$ is a fixpoint of H proceeds exactly as for Theorem 2.30. We now prove that $P(Q) \vdash Q$ is the least prefixpoint of H . To this end, assume $H(t \vdash a) = F(t \vdash a) \vdash G(t \vdash a) \leq t \vdash a$, i.e.,

$$t \leq F(t \vdash a) \text{ and } t \wedge G(t \vdash a) \leq a.$$

Since $t \leq F(t \vdash a) = F(t \vdash \bar{t} + a)$, we have $t \leq P(\bar{t} + a)$ by definition of P as greatest fixpoint. Moreover,

$$t \wedge G(P(\bar{t} + a) \vdash \bar{t} + a) = t \wedge F(t \vdash \bar{t} + a) \wedge G(P(\bar{t} + a) \vdash \bar{t} + a) \leq t \wedge G(t \vdash \bar{t} + a) = t \wedge G(t \vdash a) \leq a$$

by Lemma 2.24. By shunting, $R(\bar{t} + a) = \overline{P(\bar{t} + a)} + G(P(\bar{t} + a) \vdash \bar{t} + a) \leq \bar{t} + a$, hence $Q = \mu R \leq \bar{t} + a$. This implies $t \wedge Q \leq a$, and $t \leq P(\bar{t} + a) \leq P(Q)$ by antitony of P . Hence, $P(Q) \vdash Q \leq t \vdash a$. \square

In Section 2.3 we have derived the star operation on normal designs. As an example of using Theorem 2.32, let us now characterise the Kleene star on designs.

Corollary 2.33. Let $t \vdash a$ be given and set $H(u \vdash b) =_{df} (t \vdash a)(u \vdash b) + (\top \vdash 1)$. Then $\mu H = \overline{a^* \bar{t} \top} \vdash a^*$.

Proof. Observe that H is isotone and, by Corollary 2.9,

$$H(u \vdash b) = (t \vdash a)(u \vdash b) + (\top \vdash 1) = \overline{\bar{t} \top + a\bar{u}} \vdash (ab + 1) = F(u \vdash b) \vdash G(u \vdash b),$$

where $F(u \vdash b) =_{df} \overline{\bar{t} \top + a\bar{u}}$ and $G(u \vdash b) =_{df} \bar{t} \top + a\bar{u} + ab + 1$. By the definition of P ,

$$P(b) = \nu u. F(u \vdash b) = \nu u. \overline{\bar{t} \top + a\bar{u}} = \overline{\mu u. \bar{t} \top + au} = \overline{a^* \bar{t} \top}.$$

Since $P(b)$ is constant, let $P = P(b)$. By the definitions of R and Q , as well as star properties,

$$\begin{aligned} Q &= \mu b. \overline{P(b)} + G(P(b) \vdash b) = \mu b. \bar{P} + \bar{t} \top + a\bar{P} + ab + 1 = a^* (\bar{P} + \bar{t} \top + a\bar{P} + 1) \\ &= a^* \bar{P} + a^* \bar{t} \top + a^* = a^* a^* \bar{t} \top + a^* \bar{t} \top + a^* = a^* \bar{t} \top + a^* = \bar{P} + a^*. \end{aligned}$$

By Theorem 2.32 and Corollary 2.9, $\mu H = P(Q) \vdash Q = P \vdash \bar{P} + a^* = P \vdash a^*$. \square

3. Relating recursive definitions

As our first application of the general results of the previous section, we investigate the relation between different kinds of linear recursions. We also show how to replace the conditions of UTP by tests [Koz97] which enable a more convenient notation.

As a motivating example, we derive three variants of the computation of the factorial. Only one of the implementations is tail-recursive, which leads to considerable difficulties when trying to show their equivalence. Let us start with the tail-recursive variant. Unless stated otherwise, we assume that the variables x and y denote natural numbers.

Example 3.1. We start with the specification $P_1 =_{df} x, y := 0, yx!$ and derive, using the notations [HH98] $\cdot \triangleleft \cdot \triangleright \cdot$ for the conditional and \mathbb{I} for **skip**,

$$\begin{aligned}
P_1 &= x, y := 0, yx! \\
&= x, y := 0, yx! \triangleleft x = 0 \triangleright x, y := 0, yx! \\
&= y := y \cdot 1 \triangleleft x = 0 \triangleright x, y := 0, yx(x-1)! \\
&= y := y \triangleleft x = 0 \triangleright y := yx; x, y := 0, y(x-1)! \\
&= \mathbb{I} \triangleleft x = 0 \triangleright y := yx; x := x-1; x, y := 0, yx! \\
&= \mathbb{I} \triangleleft x = 0 \triangleright y := yx; x := x-1; P_1.
\end{aligned}$$

The calculation of the factorial is thus realised by successively multiplying the numbers $x, x-1, \dots, 1$ in decreasing order. In each recursive step, the variable x is decremented after the multiplication but before the recursive call. The recursion terminates when $x = 0$; the starting value of x is lost after this procedure.

In UTP, the solution to such a recursive equation is defined as a least fixpoint, so we obtain

$$\begin{aligned}
P_1 &= \mu X \bullet \mathbb{I} \triangleleft x = 0 \triangleright y := yx; x := x-1; X \\
&= (x \neq 0) * (y := yx; x := x-1).
\end{aligned}$$

using the notation of [HH98]. However, in UTP the least fixpoint is taken with respect to the refinement order, which is the reverse of the implication order we are using in our model of UTP designs. So we shall have to investigate the greatest fixpoint with respect to the natural order.

To represent the conditional algebraically, we use tests (see, e.g., [Koz97]). They are similar to conditions, but work more symmetrically and hence can express pre- and post-restrictions in a uniform way.

Definition 3.2. The set of *tests* of an ideal condition semiring (S, T) is $\text{test}(S, T) =_{df} \{t \wedge 1 \mid t \in T\}$. The *negation* of $p \in \text{test}(S, T)$ is $\neg p =_{df} \overline{p \top} \wedge 1$.

For designs, we obtain as tests the matrices

$$\begin{pmatrix} \top & \top \\ t & t \end{pmatrix} \wedge \begin{pmatrix} \top & \top \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \top & \top \\ 0 & t \wedge 1 \end{pmatrix} = \top \vdash (t \wedge 1).$$

Tests are isomorphic to conditions as stated in the following lemma.

Lemma 3.3. Let (S, T) be an ideal condition semiring and $t, u \in T$ and $p, q \in \text{test}(S, T)$.

1. $t = (t \wedge 1) \top$.
2. $p = p \top \wedge 1$.
3. $\neg(t \wedge 1) = \bar{t} \wedge 1$.
4. $\overline{p \top} = \neg p \top$.
5. $t \wedge 1 \leq u \wedge 1 \Leftrightarrow t \leq u$.
6. $(\text{test}(S, T), +, 0, \cdot, 1, \neg)$ is a Boolean algebra.

Proof.

1. This is a special case of Lemma 2.6.3.
2. Let $p = t \wedge 1$. Then, $p \top \wedge 1 = (t \wedge 1) \top \wedge 1 = t \wedge 1 = p$ using part 1.
3. By part 1, $\neg(t \wedge 1) = \overline{(t \wedge 1) \top} \wedge 1 = \bar{t} \wedge 1$.
4. Using again Lemma 2.6.3, $\neg p \top = \overline{(p \top) \top} \wedge 1 = \overline{p \top}$.
5. (\Leftarrow) follows by isotony of \wedge .
 (\Rightarrow) By part 1, the assumption and isotony, and part 1 again, $t = (t \wedge 1) \top \leq (u \wedge 1) \top = u$.

6. By the axioms, $(t \wedge 1) + (u \wedge 1) = (t + u) \wedge 1$. Moreover, by Lemmas 2.6.3 and 2.4, $(t \wedge 1)(u \wedge 1) = t \wedge (u \wedge 1) = (t \wedge u) \wedge 1$. Now the claim follows from the order isomorphism expressed by part 5. \square

Using a test p we can form the *input* and *output restrictions* of an element a by p as pa and ap , respectively. Hence we can define the conditional as

$$a \triangleleft p \triangleright b =_{df} pa + \neg pb.$$

Using

$$m =_{df} (y := yx) \quad d =_{df} (x := x - 1) \quad p =_{df} (x \neq 0)$$

we thus obtain for our previous example

$$P_1^\nu = \nu x.pmdx + \neg p.$$

We furthermore investigate the least fixpoint $P_1^\mu = \mu x.pmdx + \neg p$ that may be of interest in other theories. If we assume that the underlying semiring is a Kleene algebra or even an omega algebra (as are UTP designs), the fixpoints can be represented as $P_1^\mu = (pmd)^* \neg p$ and $P_1^\nu = (pmd)^\omega + (pmd)^* \neg p$.

Example 3.4. We now start with the specification $P_1 =_{df} y := yx!$ and derive

$$\begin{aligned} P_2 &= y := yx! \\ &= y := yx! \triangleleft x = 0 \triangleright y := yx! \\ &= y := y \cdot 1 \triangleleft x = 0 \triangleright y := yx(x - 1)! \\ &= y := y \triangleleft x = 0 \triangleright y := yx; y := y(x - 1)! \\ &= \mathbb{I} \triangleleft x = 0 \triangleright y := yx; x := x - 1; y := yx!; x := x + 1 \\ &= \mathbb{I} \triangleleft x = 0 \triangleright y := yx; x := x - 1; P; x := x + 1. \end{aligned}$$

The calculation proceeds as in Example 3.1, but the variable x is incremented after returning from the recursive call. Therefore, the value of x after this procedure is the same as its starting value.

We reason similarly as in the first example, and using $i =_{df} (x := x + 1)$ we thus obtain

$$P_2^\nu = \nu x.pmdxi + \neg p$$

and $P_2^\mu = \mu x.pmdxi + \neg p$. Note that this implementation is no longer tail-recursive. Hence, there is no obvious representation of the fixpoints using the Kleene star or the omega operation.

Both specifications can be related as follows:

$$P_2; x := 0 = y := yx!; x := 0 = x, y := 0, yx! = P_1.$$

To extend this relation to the implementations, both derivations above have to be performed independently. We could reduce the amount of work, if we were able to transform one implementation into the other. Our objective in the following is, therefore, to relate the implementations. Using $z =_{df} (x := 0)$ we would like to obtain

$$P_2^\nu z = (\nu x.pmdxi + \neg p)z = (\nu x.pmdx + \neg p) = P_1^\nu$$

and similarly $P_2^\mu z = P_1^\mu$.

3.1. Relating tail-recursion and linear recursion

We are now ready to relate the implementations P_1^μ and P_2^μ by the following lemma.

Lemma 3.5. Let a, b, c, d be elements of a Kleene algebra such that $bd = d$ and $cd = c$, then $(\mu x.axb + c)d = a^*c$.

Proof. Let $g(x) =_{df} axb + c$.

If the Kleene algebra is complete and composition distributes over arbitrary sums, we can apply μ -fusion [ABB⁺95]. Let $f(x) =_{df} xd$ and $h(x) =_{df} ax + c$, then

$$f(g(x)) = (axb + c)d = axbd + cd = axd + c = h(f(x)),$$

from which the claim $f(\mu g) = \mu h$ follows.

Without these additional assumptions, we can prove the claim as follows. For (\geq) note first that $c = cd = (\mu x.c)d \leq (\mu x.axb + c)d = (\mu g)d$. Second, using the fixpoint law in the last step,

$$a(\mu g)d = a(\mu g)bd \leq a(\mu g)bd + cd = (a(\mu g)b + c)d = (\mu g)d.$$

Therefore, $c + a(\mu g)d \leq (\mu g)d$, which implies $a^*c \leq (\mu g)d$ by star induction.

For (\leq) , from $a(a^*cb^*)b + c \leq a^*cb^*$ we infer $\mu g \leq a^*cb^*$ by the fixpoint induction law. Therefore,

$$(\mu g)d \leq a^*cb^*d = a^*cd = a^*c,$$

since $bd = d \Rightarrow b^*d = d$. □

Instantiating $d = 1$ in Lemma 3.5, and therefore also $b = 1$, we obtain the special case $(\mu x.ax + c) = a^*c$, the least fixpoint representation of a^*c .

Corollary 3.6. $P_2^\mu z = P_1^\mu$.

Proof. Observe that $iz = z$ since incrementing x before setting it to 0 is superfluous. Moreover, $\bar{p}z = \bar{p}$ since setting x to 0 can be omitted if it already is 0. Therefore, the assumptions of Lemma 3.5 are satisfied and we conclude

$$P_2^\mu z = (\mu x.pmdxi + \neg p)z = (pmd)^* \neg p = (\mu x.pmdx + \neg p) = P_1^\mu. \quad \square$$

To relate the implementations P_1^ν and P_2^ν , we have to restrict ourselves to UTP algebras. Note that ν -fusion cannot be directly applied since composition does not distribute over meets.

Lemma 3.7. Let a, b, c, d be elements of a UTP omega algebra such that $bd = d$ and $cd = c$. Then

1. $\nu x.axb = a^\omega$, and
2. $(\nu x.axb + c)d = a^\omega + a^*c$.

Proof.

1. Let $e(x) = axb$. By Lemma 2.22, $aa^\omega b = aa^\omega = a^\omega$, which shows that a^ω is a fixpoint of e and hence $a^\omega \leq \nu e$. For the converse inequation observe that for an arbitrary fixpoint e° of e , $e^\circ \top = ae^\circ b \top \leq ae^\circ \top$, so that $e^\circ \top \leq a^\omega$ by omega co-induction. But $e^\circ \leq e^\circ \top$ and we are done.
2. Let $g(x) =_{df} axb + c$. Using the fixpoint law in the first step,

$$(\nu g)d = (a(\nu g)b + c)d = a(\nu g)bd + cd = a(\nu g)d + c.$$

This implies the direction (\leq) by omega co-induction. By star induction, this also implies $a^*c \leq (\nu g)d$. Hence, it remains to show $a^\omega \leq (\nu g)d$ for the direction (\geq) . But this holds, since $a^\omega = a^\omega d = (\nu x.axb)d \leq (\nu g)d$ by Lemma 2.22 and part 1. □

Corollary 3.8. In a UTP algebra, $P_2^\nu z = P_1^\nu$.

This kind of reasoning is generalised in Section 4.

3.2. Relating linear recursions

Let us derive a further implementation of the factorial, again not tail-recursive.

Example 3.9. We now start with the specification $P_3 =_{df} y := x!$ and derive

$$\begin{aligned} P_3 &= y := x! \\ &= y := x! \triangleleft x = 0 \triangleright y := x! \\ &= y := 1 \triangleleft x = 0 \triangleright y := x(x-1)! \\ &= y := 1 \triangleleft x = 0 \triangleright y := (x-1)!; y := xy \\ &= y := 1 \triangleleft x = 0 \triangleright x := x-1; y := x!; x := x+1; y := yx \\ &= y := 1 \triangleleft x = 0 \triangleright x := x-1; P; x := x+1; y := yx. \end{aligned}$$

This calculation successively multiplies the numbers $1, 2, \dots, x$ in ascending order. The reversed order is

achieved by accumulating the multiplications after returning from the recursive calls instead of before. As a consequence, the variable y has to be initialised in the base case. Again, the value of x at the start and the end of the procedure are the same.

We reason similarly as in the first example, and using $o =_{df} (y := 1)$ we thus obtain

$$P_3^\nu = \nu x.pdxim + \neg po$$

and $P_3^\mu = \mu x.pdxim + \neg po$.

The specifications P_2 and P_3 can be related as follows:

$$y := 1; P_2 = y := 1; y := yx! = y := 1; y := x! = y := x! = P_1.$$

Again, our objective is to relate the implementations, i.e., we would like to obtain

$$oP_2^\nu = o(\nu x.pmdxi + \neg p) = (\nu x.pdxim + \neg po) = P_3^\nu$$

and similarly $oP_2^\mu = P_3^\mu$.

In this case, not even μ -fusion can be applied directly, since as a prerequisite we would need $pmdxi = pdxim$ for arbitrary x , which is not true.

We will, however, show that under certain side conditions the finite parts of the following two recursions are equivalent:

$$\mu x.pmdxi + \neg p \qquad \mu x.pdxim + \neg p$$

One central assumption is $di = 1 = id$ which implies $d^k i^k = 1 = i^k d^k$ for all $k \in \mathbb{N}$. Therefore, deviating slightly from the previous examples, for the remainder of this section we assume the program variable x to be an integer rather than a natural number.

We first ignore the tests and show that the finite approximations then coincide; these are, respectively, $(md)^k i^k$ and $d^k (im)^k$ if termination occurs after k steps.

As an abbreviation, for arbitrary a and $n \in \mathbb{N}$ we set $a^{(n)} =_{df} d^n a i^n$. This corresponds to executing a in a “future” state after n operations of type d and restoring the initial state afterwards using i repeatedly.

In the concrete case where $m \hat{=} y := yx$, $d \hat{=} x := x - 1$ and $i \hat{=} x := x + 1$ we have $m^{(n)} \hat{=} y := y(x - n)$. An assumption $m^{(j)} m^{(k)} = m^{(k)} m^{(j)}$ then expresses a special case of the (right) commutativity of multiplication.

Lemma 3.10.

1. $(md)^k = m^{(0)} \dots m^{(k-1)} d^k$.
2. $(im)^k = i^k m^{(k-1)} \dots m^{(0)}$.
3. If the $m^{(j)}$ in the formulas above commute, then $(md)^k i^k = d^k (im)^k$.

Proof.

1. The proof is by induction on k . The base case $k = 0$ is obvious. For the induction step let $e =_{df} m^{(0)} \dots m^{(k-1)}$. Then

$$(md)^{k+1} = (md)^k md \stackrel{\text{IH}}{=} ed^k md = ed^k m i^k d^k d = em^{(k)} d^{k+1} = m^{(0)} \dots m^{(k)} d^{k+1}.$$

2. Symmetrically to part 1.
3. By the commutativity assumption, $e =_{df} m^{(0)} \dots m^{(k-1)} = m^{(k-1)} \dots m^{(0)}$. Hence, using parts 1 and 2,

$$(md)^k i^k = ed^k i^k = e = d^k i^k e = d^k (im)^k. \quad \square$$

We now include tests into our considerations. We will assume that for a test p also all $p^{(j)}$ and $(\neg p)^{(j)}$ (which are below 1) are tests.

The informal idea behind the next lemma is to move in an iteration $(pmd)^k$ all occurrences of p to the left so that on the right a pure iteration of md remains and we can apply the previous lemma. Consider a sequence mdp in which p is tested *after* md . Suppose now that m does not influence p (which holds for the concrete case above when $p \hat{=} x > 0$, so that we have again programs that compute the factorial). Then we can also first change the state according to d , test p and restore the original state using i . After that we execute m and d and can omit the test of p , since it has already been tested “beforehand”. In formulas,

$mdp = dpimd$ or, using our above abbreviation, $mdp^{(0)} = p^{(1)}md$, where the required independence of p from m is expressed as the commutativity requirement $pm = mp$.

If we have that property then

$$(pmd)^2 = pmdpmd = p^{(0)}mdp^{(0)}md = p^{(0)}p^{(1)}mdmd = p^{(0)}p^{(1)}(md)^2$$

and we have achieved our goal in this special case. The general case is covered by the following lemma.

Lemma 3.11. Let q be a test and a an element that commutes with all tests $q^{(j)}$ and $(\neg q)^{(j)}$. Denote by $r =_{df} \prod_{j=0}^{k-1} q^{(j)}$ the conjunction of the tests q as performed in the states reached from the initial one in at most $k-1$ steps of type d .

1. $adq^{(j)} = q^{(j+1)}ad$.
2. $(ad)^k q^{(j)} = q^{(j+k)}(ad)^k$.
3. $(qad)^k = r(ad)^k$.
4. $(qad)^k \neg q = r(\neg q)^{(k)}(ad)^k$.
5. $(qd)^k \neg q = r(\neg q)^{(k)}d^k$.

Proof.

1. $q^{(j+1)}ad = aq^{(j+1)}d = ad^{j+1}qi^{j+1}d = add^jqi^j = adq^{(j)}$.
2. Induction on k using 1.
3. Induction on k using 1.
4. Follows from 2. and 3.
5. This is the special case $a = 1$ of 4. □

Now we are ready for the main result which implies that the finite parts of oP_2^μ and P_3^μ , respectively oP_2^ν and P_3^ν coincide (since o commutes with p and d). An investigation of the infinite parts is postponed to the next section.

Theorem 3.12. Assume that m commutes with all tests $p^{(j)}$ and $(\neg p)^{(j)}$ and that the $m^{(j)}$ involved in the formulas below commute. Then

$$(pmd)^k \neg pi^k = (pd)^k \neg p(im)^k.$$

Proof. Set $r =_{df} \prod_{j=0}^{k-1} p^{(j)}$. By Lemmas 3.11.4, 3.10.3 and 3.11.5,

$$(pmd)^k \neg pi^k = r(\neg p)^{(k)}(md)^k i^k = r(\neg p)^{(k)}d^k(im)^k = (pd)^k \neg p(im)^k. \quad \square$$

4. Symmetric linear recursion

We further investigate fixpoints of the function $f(x) = axb + c$ using now modal Kleene algebras, i.e., Kleene algebras with domain, diamond and box operators. The investigation proceeds by separately considering the finite and the infinite parts of the fixpoints. One goal is to implement the recursion described by f by two consecutive while-loops.

The elements a, b, c in the definition of the function f can be instantiated to normal designs due to the results of Section 2. The results in this section therefore also apply to UTP.

Since certain results hold only if a is deterministic, we need to characterise determinacy algebraically. For this, we can again employ tests, together with the domain operation on which we can base the modal operators. We also use tests for the algebraic representation of (co-)invariants that will simplify subsequent arguments.

The domain of a semiring element a characterises the starting states of a , i.e., the states from which corresponding output states may be reached under a . We use the equational axiomatisation of [DMS06b].

Definition 4.1. Assume an ideal condition semiring (S, T) . The domain operation $\Gamma : S \rightarrow \text{test}(S, T)$ is

characterised by the axioms

$$a \leq \lceil aa \quad (\text{d1})$$

$$\lceil pa \leq p \quad (\text{d2})$$

$$\lceil a\lceil b \leq \lceil ab \quad (\text{d3})$$

Let us explain these axioms. Since $\lceil a \leq 1$ by $\lceil a \in \text{test}(S, T)$, isotony of multiplication shows that the first axiom can be strengthened to an equality expressing that restriction to the full domain is no restriction at all. The second axiom means that after restriction the remaining domain must satisfy the restricting test. Finally, (d3) says that in the interaction of a and b only their “boundary” matters and not their inner structure.

According to [DMS06b] (d1) \wedge (d2) is equivalent to

$$\lceil a \leq p \Leftrightarrow a \leq p \cdot a. \quad (2)$$

Since an ideal condition semiring has a greatest element \top , there is also an equivalent characterisation in the form of a Galois connection (see, e.g., [Aar92] and again [DMS06b]):

$$\lceil a \leq p \Leftrightarrow a \leq p\top. \quad (3)$$

Hence, the domain operation is unique if it exists. Moreover, it preserves arbitrary existing suprema. Further properties can be found in [DMS06b].

With the help of domain we define the forward modal operators diamond and box as test transformers:

$$\langle a \rangle p =_{df} \lceil (a \cdot p), \quad [a]p =_{df} \neg \langle a \rangle \neg p.$$

Thus $\langle a \rangle p$ characterises those states for which *some* a -successor state satisfies p , whereas $[a]p$ characterises those states for which *all* a -successor states satisfy p . The box operator is the abstract counterpart of the wlp operator [Dij76]. These definitions imply many useful algebraic properties [DMS06b].

It turns out that the designs over an ideal semiring with domain can be equipped with a domain operation, too [Möl06]. We use the characterisation (2) to find the proper definition. Since test designs take the form $\top \vdash p$ with a test p , we can calculate as follows:

$$\begin{aligned} \lceil \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} \leq \begin{pmatrix} \top & \top \\ 0 & p \end{pmatrix} &\Leftrightarrow \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} \leq \begin{pmatrix} \top & \top \\ 0 & p \end{pmatrix} \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} \Leftrightarrow \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} \leq \begin{pmatrix} \top & \top \\ pt & pa \end{pmatrix} \Leftrightarrow \\ t \leq pt \wedge a \leq pa &\Leftrightarrow \lceil t \leq p \wedge \lceil a \leq p \Leftrightarrow \lceil t + \lceil a \leq p \Leftrightarrow \lceil (t + a) \leq p \Leftrightarrow \lceil a \leq p, \end{aligned}$$

since for designs we have $t \leq a$. Hence for

$$\lceil \begin{pmatrix} \top & \top \\ t & a \end{pmatrix} =_{df} \begin{pmatrix} \top & \top \\ 0 & \lceil a \end{pmatrix}$$

axioms (d1) and (d2) hold by construction and a straightforward calculation shows that (d3) is satisfied as well.

Moreover, we can use diamond and box to characterise determinacy.

Definition 4.2. An element a is *modally deterministic* if $\langle a \rangle \leq [a]$.

Equivalently [DM01], a is modally deterministic if for all tests p_1, p_2 with $p_1 p_2 = 0$ also $\langle a \rangle p_1 \cdot \langle a \rangle p_2 = 0$. In the remainder we omit the qualifier “modally” and only speak of deterministic elements.

To reason about the interaction of an element and a test, we introduce the notion of a (co-)invariant. To prepare it, we first note that the characterising property (2) of domain entails the following characterisations of diamond and box as well as equivalent test propagation properties:

$$\langle a \rangle p \leq q \Leftrightarrow \lceil ap \leq q \Leftrightarrow ap \leq qap \Leftrightarrow ap \leq qa, \quad (4)$$

$$p \leq [a]q \Leftrightarrow pa \neg q \leq 0 \Rightarrow pa \leq paq \Leftrightarrow pa \leq aq. \quad (5)$$

In a right-strict setting the implication in (5) becomes an equivalence. In the particular case where $q = p$, that test propagates backwards/forwards through a , respectively, i.e., is a (co-)invariant of a :

Definition 4.3. A test p is an *invariant* of a if $pa \leq ap$, and a *co-invariant* of a if $ap \leq pa$.

Lemma 4.4. Let p be a test and a, b elements of a domain semiring.

1. If p is a co-invariant of a , then $\neg p$ is an invariant of a .
2. In a right-strict setting, p is a co-invariant of a iff $\neg p$ is an invariant of a .
3. Let a and b commute, then $\neg \bar{b}$ is an invariant of a .

Proof.

1. Assuming $ap \leq pa$, we have $\neg pa = \neg pap + \neg pa\neg p \leq \neg ppa + \neg pa\neg p = \neg pa\neg p \leq a\neg p$.
2. Assuming $pa \leq ap$, we have $a\neg p = pa\neg p + \neg pa\neg p \leq ap\neg p + \neg pa\neg p = \neg pa\neg p \leq \neg pa$. The claim follows together with part 1.
3. $\langle a \rangle \bar{b} = \ulcorner(a\bar{b}) = \ulcorner(ab) = \ulcorner(ba) \leq \bar{b}$, hence \bar{b} is a co-invariant of a . The claim follows by part 1. \square

We will freely use the above equivalent characterisations of co-invariants. In a Kleene algebra, by the standard star semi-commutation laws

$$ab \leq ba \Rightarrow a^*b \leq ba^* \wedge ab^* \leq b^*a, \quad (6)$$

a (co-)invariant of a is also one of a^* .

Several sufficient criteria for co-invariance under a deterministic a are given by the following lemma.

Lemma 4.5. Let a be deterministic.

1. If p is contracted by $[a]$, i.e., $[a]p \leq p$, then p is a co-invariant of a .
2. If p is expanded by $\langle a \rangle$, i.e., $p \leq \langle a \rangle p$, then $\neg p$ is a co-invariant of a .

Let us explain the intuition underlying part 2: p being expanded by $\langle a \rangle$ means pointwise that every point in p has an a -successor in p . Dually, $\neg p$ being a co-invariant of a means that all a -predecessors of points in $\neg p$ are in $\neg p$ again. Now assume that a is deterministic and some point x in $\neg p$ has an a -predecessor y in p . Then y would have an a -successor in p . But by determinacy of a the only a -successor of y is x , which is in $\neg p$; we thus obtain a contradiction. Part 1 can be discussed in a similar manner. The formal proofs, however, are much shorter:

Proof.

1. $[a]p \leq p \Rightarrow \langle a \rangle p \leq p$.
2. $p \leq \langle a \rangle p \Leftrightarrow \neg \langle a \rangle p \leq \neg p \Leftrightarrow [a]\neg p \leq \neg p$. Now apply part 1. \square

We now return to our function $f(x) =_{df} axb + c$. We say that *the choice in f is deterministic* if $\ulcorner a \ulcorner c = 0$. Our main goal in the following is to derive an implementation of f by two consecutive while-loops. To prove the correctness of the implementation, we look at the finite part and at the infinite part of the recursion, in turn. The separate correctness results are then combined.

4.1. The finite part

We use elements t from the underlying semiring to describe the left context in which the computation modelled by a terminates after at most (or exactly) n recursive steps. In the special case of t being a test, it describes the states from which such a termination occurs.

Definition 4.6. Let $n \in \mathbb{N}$ and t be a semiring element. Then t *terminates a after at most n steps* if

$$ta^n = ta^n \neg \ulcorner a$$

and *terminates a after exactly n steps* if, additionally,

$$\forall k < n : ta^k = ta^k \ulcorner a.$$

As a consequence, $\forall k > n : ta^k = ta^n aa^{k-n-1} = ta^n \neg \ulcorner a \ulcorner aa^{k-n-1} = ta^n 0$. The following lemma simplifies the recursion for such terminating states.

Lemma 4.7. Let f° be a fixpoint of f .

1. If t terminates a after at most n steps, $tf^\circ = \sum_{k=0}^n ta^k cb^k$.
2. If the choice in f is deterministic and t' terminates a after exactly n steps, $t'f^\circ = t'a^n cb^n$.

Proof. We first show by induction that $\forall n \in \mathbb{N} : f^\circ = a^n f^\circ b^n + \sum_{k < n} a^k c b^k$. For $n = 0$, this is clear, and for $n \geq 0$,

$$\begin{aligned} a^{n+1} f^\circ b^{n+1} + \sum_{k < n+1} a^k c b^k &= a^n (a f^\circ b) b^n + a^n c b^n + \sum_{k < n} a^k c b^k \\ &= a^n (a f^\circ b + c) b^n + \sum_{k < n} a^k c b^k = a^n f^\circ b^n + \sum_{k < n} a^k c b^k = f^\circ. \end{aligned}$$

1. Since $ta^{n+1} = ta^n a = ta^{n-\lceil a \rceil} a a = ta^n 0$,

$$t f^\circ = t(a^{n+1} f^\circ b^{n+1} + \sum_{k < n+1} a^k c b^k) = ta^n 0 + ta^n c b^n + t \sum_{k < n} a^k c b^k = \sum_{k=0}^n ta^k c b^k.$$

2. Since the choice in f is deterministic, $\forall k < n : t' a^k c = t' a^k \lceil a \rceil c c = t' a^k 0$. Therefore, continuing the proof of part 1,

$$t' f^\circ = \sum_{k=0}^n t' a^k c b^k = t' a^n c b^n + \sum_{k < n} t' a^k c b^k = t' a^n c b^n + \sum_{k < n} t' a^k 0 = t' a^n c b^n,$$

since $t' a^k 0 \leq t' a^n c b^n$ for $k < n$. □

Under additional assumptions, we can represent the finite part of a fixpoint of f by two consecutive while-loops. This may be compared to the construct `dojustasoften` of [Bau76].

The assumptions concern four special elements o, i, d, z of the semiring which together implement an abstract counter. Intuitively,

- o initialises a (new) counter to 0,
- i increments that counter,
- d decrements the counter if its value is greater than 0, and
- z tests whether the counter is 0.

These elements count the number of iterations, provided they do not interfere with the loop constituents which is ensured by commutativity conditions.

The idea now is to represent the recursion given by f as two consecutive while-loops. The first one performs the a operations and increases the counter as many times as recursive calls occur. After that it performs the termination action c . The second loop performs the b actions and decreases the counter till it becomes zero again.

As usual, the while-loops will be represented by fixpoints of tail-recursive functions. So let $g(x) =_{df} a i x + c$ and $h(x) =_{df} d b x + z$. Their least fixpoints are the loops $(a i)^* c$ and $(d b)^* z$. However, we will reason about arbitrary fixpoints of these functions.

Lemma 4.8. Let o, i, d, z be such that they commute with a, b, c , and

$$oz = id = 1 \wedge od = iz = 0.$$

Let f°, g° and h° be fixpoints of f, g and h , respectively, and let t terminate a after at most n steps. Then, $t f^\circ = t o g^\circ h^\circ$.

Proof. The proof proceeds in three parts.

- We show that the left loop correctly realises the iterations of a while accumulating a counting right context, i.e., that $t o g^\circ = \sum_{k=0}^n ta^k c o i^k$. Observe that $t o$ terminates $a i$ after at most n steps, since

$$t o (a i)^n = ta^n o i^n = ta^{n-\lceil a o i \rceil} a o i^n \leq ta^n o i^{n-\lceil a \rceil} = t o (a i)^{n-\lceil a \rceil} \leq t o (a i)^{n-\lceil a i \rceil}$$

by Lemma 4.4.3. Hence, by Lemma 4.7.1, $t o g^\circ = \sum_{k=0}^n t o (a i)^k c = \sum_{k=0}^n ta^k c o i^k$.

- We show that in the accumulated left context $o i^n$ the right loop correctly realises the iterations of b , i.e., that $o i^n h^\circ = b^n$. Observe that $o i^k$ terminates $d b$ after at most k steps, because $o \lceil d \rceil = 0$ and, by Lemma 4.4.3,

$$o i^k (d b)^k = o i^k d^k b^k = o \lceil d b \rceil^k \leq o b^k \lceil d \rceil = o i^k d^k b^k \lceil d \rceil \leq o i^k (d b)^k \lceil d b \rceil.$$

Hence, by Lemma 4.7.1,

$$oi^n h^\circ = \sum_{k=0}^n oi^n (db)^k z = oi^n d^n b^n z + \sum_{k=0}^{n-1} oi^{n-k-1} iz b^k = oi^n d^n b^n z + \sum_{k=0}^{n-1} oi^{n-k-1} 0 = oi^n d^n b^n z = oz b^n = b^n.$$

– We combine both facts obtained above to conclude

$$tog^\circ h^\circ = \sum_{k=0}^n ta^k coi^k h^\circ = \sum_{k=0}^n ta^k cb^k = f^\circ$$

by Lemma 4.7.1. □

Let us remark that not all commutativity conditions concerning the counter are necessary for Lemma 4.8. The proof above actually only uses that a, c commute with o, i and that b commutes with d, z .

The following lemma makes it clear that tests satisfying Definition 4.6 do exist.

Lemma 4.9. Let $n \in \mathbb{N}$ and let $q_n =_{df} \ulcorner(a^n)\urcorner\lrcorner(a^{n+1})$ and $q'_n =_{df} \ulcorner(a^n)\urcorner\lrcorner a$. Note that $q'_n = \langle a^n \rangle\lrcorner a = \lrcorner[a^n]\ulcorner a$.

1. $q_n \leq q'_n$, and if a is deterministic, $q_n = q'_n$.
2. The test q_n terminates a after at most n steps.
3. If a is deterministic, the test q'_n terminates a after exactly n steps.

Proof. The following, introductory remark is used in parts 1 and 3 of the proof, whenever a is deterministic. Observe that determinacy is closed under composition and the element 1 is deterministic [DM01]. It follows by induction that a^j is deterministic for every $j \in \mathbb{N}$.

1. (\leq) follows from $\ulcorner(a^n) = \ulcorner(a^n \ulcorner a + a^n \lrcorner a) = \ulcorner(a^{n+1})\urcorner + \ulcorner(a^n \lrcorner a)$ by shunting. For (\geq) observe $\ulcorner(a^n \lrcorner a) \leq \ulcorner(a^n)$ and $\ulcorner(a^n \lrcorner a) = \langle a^n \rangle\lrcorner a \leq [a^n]\lrcorner a = \lrcorner \langle a^n \rangle \ulcorner a = \lrcorner \ulcorner(a^{n+1})$.
2. Since $\ulcorner(q_n a^n \ulcorner a) = q_n \ulcorner(a^n \ulcorner a) = \ulcorner(a^n)\urcorner\lrcorner(a^{n+1})\ulcorner(a^{n+1}) = 0$, we have $q_n a^n \ulcorner a = 0$, hence $q_n a^n \leq q_n a^n \lrcorner a$.
3. By parts 1 and 2, $q'_n = q_n$ terminates a after at most n steps. It remains to show that termination does not occur before n steps. Let $k < n$, then a^k is deterministic, and we have

$$q'_n = \langle a^n \rangle\lrcorner a = \langle a^k \rangle \langle a \rangle \langle a^{n-k-1} \rangle\lrcorner a \leq \langle a^k \rangle \langle a \rangle 1 = \langle a^k \rangle \ulcorner a \leq [a^k]\ulcorner a.$$

Therefore, by (5), $q'_n a^k = q'_n q'_n a^k \leq q'_n a^k \ulcorner a$. The converse inequation is trivial. □

4.2. The infinite part: convergence and iteration

For the results in this section we need to characterise the starting states of an element a from which no infinite transition paths emerge. This set is represented as the test Δa [MS06], which is axiomatised as follows.

Definition 4.10. Assume an ideal condition semiring (S, T) . The *convergence operation* $\Delta : S \rightarrow \text{test}(S, T)$ satisfies the unfold and co-induction laws

$$\begin{aligned} [a](\Delta a) &= \Delta a \\ p \cdot [a]q \leq q &\Rightarrow \Delta a \cdot [a^*]p \leq q \end{aligned}$$

for $a \in S$ and $p, q \in \text{test}(S, T)$.

Hence $\Delta a \cdot [a^*]p$ is the least (pre)fixpoint of $\lambda q. p \cdot [a]q$ and Δa is the least (pre)fixpoint of $[a]$. Moreover, $\lrcorner a \leq \Delta a \leq \lrcorner(a^\omega)$ and hence $\Delta a \cdot a^\omega = 0$. Finally, Δ is antitone and $[a^*](\Delta a) = [a \cdot a^*](\Delta a) = [a](\Delta a) = \Delta a$. For proofs of these properties see [GM06].

In addition to the convergence of an element a we consider its *divergence* $\nabla a =_{df} \lrcorner \Delta a$. It abstracts the set of points from which infinite a -transition paths start. By standard fixpoint theory and the de Morgan duality between box and diamond we have $\nabla a = \nu \langle a \rangle$.

We now use (co-)invariants to show the interaction between an element and its convergence and divergence.

Lemma 4.11.

1. Δa is an invariant of a .
2. ∇a is a co-invariant of a .
3. If a is deterministic then Δa and ∇a commute with a .

Speaking pointwise, a transition that leads to a divergent path is the beginning of a divergent path itself.

Proof.

1. Immediate from the definition of Δa and (5).
2. Immediate from the definition of ∇a and (4).
3. The commutativity of Δa with a is immediate from Lemma 4.5.1 and part 1. The commutativity of ∇a with a now follows by Lemma 4.4.1 and part 2. \square

The following lemma relates divergence to finite iteration.

Lemma 4.12. Let a be deterministic.

1. If $p \leq \langle a \rangle p$, then $pa^* \neg \ulcorner a = 0$.
2. $\nabla aa^* \neg \ulcorner a = 0$.
3. If $p \leq \nabla a$, then $pa^* \neg \ulcorner a = 0$.

Part 1 again has a nicely intuitive, pointwise interpretation: Since every point in p has an a -successor in p , and that is the only a -successor, we can never reach a dead end of a when starting in p .

Proof. 1. First, $p \leq \langle a \rangle p = \ulcorner (ap) \leq \ulcorner a$. Second, $\neg p$ is a co-invariant of a by Lemma 4.5.2. Hence, by (6),

$$pa^* \neg \ulcorner a \leq pa^* \neg p \leq p \neg pa^* = 0.$$

2. By definition, $p = \nabla a$ satisfies the assumption of part 1.
3. Immediate from part 2. \square

We now apply this lemma to both representations of our non-tail-recursion f , the fixpoint and the while-loop. The result parallels Lemmas 4.7 and 4.8.

Lemma 4.13. Let a and the choice in f be deterministic and assume $p \leq \nabla a$.

1. $pa^* c = 0$.
2. $p(\mu f) = 0$.
3. Under the assumptions of Lemma 4.8, $po(\mu g) = 0 = po(\mu g)(\mu h)$.
4. In a UTP omega algebra, $p(\nu f) = pa^\omega$.
5. In a UTP omega algebra, under the assumptions of Lemma 4.8, $po(\nu g)(\nu h) \leq p(\nu f)$.

Proof. 1. By determinacy of the choice and Lemma 4.12.3,

$$pa^* c = pa^* \ulcorner cc \leq pa^* \neg \ulcorner ac = 0.$$

2. Observe that $a^* cb^*$ is contracted by f , and therefore $\mu f \leq a^* cb^*$. Hence, $p(\mu f) \leq pa^* cb^* = 0$ by part 1.
3. By the counter assumptions and (6),

$$po(\mu g) = po(ai)^* c \leq poa^* i^* c = pa^* coi^* = 0$$

again by part 1. Hence, $po(\mu g)(\mu h) = 0(\mu h) = 0$.

4. Let $e(x) = axb$. We show $p(\nu f) = p(\nu e)$ from which the claim follows by Lemma 3.7.1. (\geq) is obvious by isotony, and for (\leq) note that by Lemma 4.11.3,

$$\nabla a(\nu f) = \nabla a(a(\nu f)b + c) = \nabla aa(\nu f)b + \nabla ac = a\nabla a(\nu f)b$$

since $\nabla a \leq \ulcorner a \leq \neg \ulcorner c$. By the greatest fixpoint property, $\nabla a(\nu f) \leq (\nu e)$. Hence, also $p(\nu f) = p\nabla a(\nu f) \leq p(\nu e)$.

5. $po(\nu g)(\nu h) = po(\mu g + (ai)^\omega)(\nu h) = po(\mu g)(\nu h) + po(ai)^\omega(\nu h) = po(ai)^\omega$ by part 3 and Lemma 2.22. By Lemma 4.14 below, $po(ai)^\omega \leq poa^\omega \leq pa^\omega$. \square

Lemma 4.14. Let x and y be elements of an omega algebra that commute. Then $xy^\omega \leq y^\omega$ and $(xy)^\omega \leq x^\omega$.

Proof. By omega unfold and commutativity, $xy^\omega \leq xyy^\omega = yxy^\omega$. The first claim $xy^\omega \leq y^\omega$ now follows by omega co-induction.

By omega unfold, $y(xy)^\omega \leq yxy(xy)^\omega$, hence by omega co-induction $y(xy)^\omega \leq (yx)^\omega$. Therefore, by omega unfold and isotony, $(xy)^\omega \leq xy(xy)^\omega \leq x(yx)^\omega$.

By commutativity, $(xy)^\omega \leq x(xy)^\omega$. The second claim $(xy)^\omega \leq x^\omega$ now follows by omega co-induction. \square

Note that Lemma 4.13.5 only claims \leq and not equality for the greatest fixpoints. In terms of refinement [HH98] this means that the recursion specified by f may be implemented by the while-loops.

The assumption of Lemma 4.13 is, in particular, satisfied by $p = \lceil a^\omega$. The assumptions concerning determinism may be weakened by noting that the essential $pa^*c \leq 0$ is equivalent to $p \leq \neg \langle a^* \rangle^c = [a^*]^{-c}$. Intuitively, this characterises the states from which no a -transition paths into the domain of c exist. These are just the states where proper termination does not occur.

We conclude with the following statement, similar to Lemma 7.6 of [DMS06a].

Remark 4.15. If $\lceil a \leq a\top$ (equivalently, $\lceil a\top = a\top$) and a is deterministic then $\nabla a = \lceil a^\omega$.

By Lemma 4.11.3, ∇a commutes with a . Hence, $\nabla a\top = \nabla a\lceil a\top = \nabla aa\top = a\nabla a\top$. By omega co-induction, $\nabla a\top \leq a^\omega$. Therefore, $\nabla a = \lceil \nabla a\top \leq \lceil a^\omega$. The reverse direction follows from the greatest fixpoint property of ∇a .

4.3. Putting the finite and infinite parts together

We can now combine the results for the finite and infinite parts obtained in the previous sections. The first lemma shows how to split up the starting states into these two parts.

In the following we assume that certain countable sums of tests exist. This is the case, e.g., when the Boolean test algebra is complete. Distribution of arbitrary elements over these sums is also assumed.

Lemma 4.16.

1. Let $q_n = \lceil a^n \rceil \neg \lceil a^{n+1} \rceil$ as in Lemma 4.9 and assume that $r =_{df} \sum_{n \in \mathbb{N}} q_n$ exists. Then, $a^\infty =_{df} \neg \sum_{n \in \mathbb{N}} \neg \lceil a^n \rceil$ exists and $a^\infty + r = 1$.
2. Let $q'_n = \langle a^n \rangle \neg \lceil a \rceil$ as in Lemma 4.9 and assume that $r' =_{df} \sum_{n \in \mathbb{N}} q'_n$ exists and a distributes over this sum. Then, $r' = \langle a^* \rangle \neg \lceil a \rceil$ and $\nabla a + r' = 1$.

Proof.

1. We show that $\sum_{n \in \mathbb{N}} \neg \lceil a^n \rceil = r$. Let x be an upper bound of the tests in the sum, i.e., $\forall n \in \mathbb{N} : \neg \lceil a^n \rceil \leq x$. Since $q_n \leq \neg \lceil a^{n+1} \rceil$, we have $\forall n \in \mathbb{N} : q_n \leq x$. Therefore, x is also an upper bound of r .

It remains to show that $\neg \lceil a^n \rceil \leq r$ for all $n \in \mathbb{N}$. We prove by induction that, more generally, $\sum_{i=0}^{n-1} q_i = \sum_{i=0}^n \neg \lceil a^i \rceil$. For the induction base $n = 0$ this is clear since $0 = \neg 1 = \neg \lceil 1 \rceil = \neg \lceil a^0 \rceil$. For the induction step $n \geq 0$ we use $\neg p + pq = \neg p + q$ to calculate

$$\begin{aligned} \sum_{i=0}^n q_i &= \sum_{i=0}^{n-1} q_i + q_n \stackrel{\text{IH}}{=} \sum_{i=0}^n \neg \lceil a^i \rceil + q_n = \sum_{i=0}^{n-1} \neg \lceil a^i \rceil + \neg \lceil a^n \rceil + \lceil a^n \rceil \neg \lceil a^{n+1} \rceil \\ &= \sum_{i=0}^{n-1} \neg \lceil a^i \rceil + \neg \lceil a^n \rceil + \neg \lceil a^{n+1} \rceil = \sum_{i=0}^{n+1} \neg \lceil a^i \rceil. \end{aligned}$$

2. Since $q'_n \leq \langle a^* \rangle \neg \lceil a \rceil$ for each $n \in \mathbb{N}$, we have $r' \leq \langle a^* \rangle \neg \lceil a \rceil$.

The reverse inequation reduces by diamond star induction to $\langle a \rangle r' + \neg \lceil a \rceil \leq r'$, equivalently, $\neg \lceil a \rceil \leq r'$ and $\langle a \rangle r' \leq r'$. The first property holds by definition of r' , since $\neg \lceil a \rceil = q'_0$. The second one is equivalent to r' being a co-invariant of a . To show it, observe that $\langle a \rangle q'_n = \langle a \rangle \langle a^n \rangle \neg \lceil a \rceil = \langle a^{n+1} \rangle \neg \lceil a \rceil = q'_{n+1}$, hence

$aq'_n \leq q'_{n+1}a$ by (4). Therefore, and since $q'_0a = 0$,

$$ar' = \sum_{n \in \mathbb{N}} aq'_n \leq \sum_{n \in \mathbb{N}} q'_{n+1}a = \sum_{n \in \mathbb{N}^+} q'_n a = \sum_{n \in \mathbb{N}} q'_n a = r'a.$$

The existence of the intermediate sums is guaranteed by the existence of r' and the distributivity of a . For the second claim, observe that $1 = \ulcorner a + \lrcorner a = \langle a \rangle 1 + \lrcorner a$, from which $1 \leq \nabla a + \langle a^* \rangle \lrcorner a = \nabla a + r'$ follows by divergence co-induction. \square

We may partition the states according to Lemma 4.16.2 into the finite and infinite parts and apply the results obtained in Sections 4.1 and 4.2. As the result, we obtain the equivalence of the recursion specified by f and the corresponding implementation by two while-loops.

Theorem 4.17. Let a and the choice in f be deterministic. Under the assumptions of Lemma 4.8, $\mu f = o(\mu g)(\mu h)$ and $\nu f \geq o(\nu g)(\nu h)$.

Proof. By Lemma 4.16.2, distributivity, Lemmas 4.13 and 4.8, again distributivity, and again Lemma 4.16.2,

$$\begin{aligned} \mu f &= (\nabla a + \sum_{n \in \mathbb{N}} q'_n)(\mu f) = \nabla a(\mu f) + \sum_{n \in \mathbb{N}} q'_n(\mu f) \\ &= \nabla a o(\mu g)(\mu h) + \sum_{n \in \mathbb{N}} q'_n o(\mu g)(\mu h) = (\nabla a + \sum_{n \in \mathbb{N}} q'_n) o(\mu g)(\mu h) = o(\mu g)(\mu h). \end{aligned}$$

The calculation for ν proceeds similarly, except for using \geq in the third step. \square

If the underlying partial order is complete, we can generalise the μ -part of Theorem 4.17 using μ -fusion.

Theorem 4.18. Let S be a Kleene algebra in which arbitrary sums of elements exist and composition distributes over such sums. Under the assumptions of Lemma 4.8, $\mu f = o(\mu g)(\mu h)$.

Proof. For reasons that will become clear later, we restrict the domain of g . Let $C \subseteq S$ contain the elements that commute with i , i.e., $C = \{x \in S \mid xi = ix\}$. Note that C is complete by the assumptions. Let $g' : C \rightarrow S$ be the restriction of g to C . By the commutativity assumptions,

$$g'(x)i = (aix + c)i = aixi + ci = iaix + ic = i(aix + c) = ig'(x),$$

hence the type of g' is even $g' : C \rightarrow C$. Closing our introductory remark, observe that $\mu g' = \mu g \in C$, since

$$(\mu g)i = (ai)^* ci = (ia)^* ic = i(ai)^* c = i(\mu g).$$

Let $e : C \rightarrow S$ be given by $e(x) = ox(\mu h)$. If we can prove $e \circ g' = f \circ e$, the claim follows by μ -fusion. Observe that

$$\begin{aligned} f(e(x)) &= ae(x)b + c = aox(\mu h)b + c, \text{ and} \\ e(g'(x)) &= og'(x)(\mu h) = o(aix + c)(\mu h) = oaix(\mu h) + oc(\mu h). \end{aligned}$$

But the latter equals $aoxi(\mu h) + co(\mu h)$ by the commutativity assumptions and the restriction to C . It therefore suffices to show $i(\mu h) = (\mu h)b$ and $o(\mu h) = 1$, which hold by

$$\begin{aligned} i(\mu h) &= i(db)^* z = i(1 + d(bd)^* b)z = iz + id(bd)^* bz = 0 + (db)^* zb = (\mu h)b, \\ o(\mu h) &= o(db)^* z = o(1 + d(bd)^* b)z = oz + od(bd)^* bz = 1 + 0 = 1, \end{aligned}$$

using star properties and the counter assumptions. \square

Note that Lemma 4.16.1 gives a statement about q_n , which has been used in Section 4.1 to generalise the results about the finite part to non-deterministic a and non-deterministic choice in f . Lemma 4.13, however, does not carry on this generalisation to the infinite part. If a is deterministic, we can show that both parts of Lemma 4.16 coincide.

Lemma 4.19. Let a be deterministic and $a^\infty = \lrcorner \sum_{n \in \mathbb{N}} \lrcorner (a^n)$ as given by Lemma 4.16. Then, $a^\infty = \nabla a$.

Proof. The (\geq) direction holds even unconditionally, since

$$\nabla a \leq \langle a^n \rangle \nabla a \leq \ulcorner (a^n) \Rightarrow \lrcorner (a^n) \leq \lrcorner \nabla a \Rightarrow \sum_{n \in \mathbb{N}} \lrcorner (a^n) \leq \lrcorner \nabla a \Rightarrow \nabla a \leq a^\infty.$$

For the (\leq) part, let q_n, q'_n, r and r' be given as in Lemma 4.16. Observe that determinacy of a implies $q_n = q'_n$ by Lemma 4.9.3, hence $r = r'$. As shown in the proof of Lemma 4.16.1, even $r = \neg a^\infty$, hence $\nabla a + \neg a^\infty = 1$, from which $a^\infty \leq \nabla a$ follows. \square

This result has the following interpretation in terms of [SS89]: $\neg \nabla a = \Delta a$ describes the *progressively finite* states, i.e., states from which no infinite a -transition paths emerge (also known as the *initial part* of a). $\neg a^\infty$ describes the *progressively bounded* states, i.e., which have an upper bound on the lengths of the emerging a -transition paths. Every progressively bounded state is progressively finite. As detailed in Section 5, with deterministic a the progressively bounded and the progressively finite states are the same. The result can be seen as a special case of König's Infinity Lemma.

4.4. Axiomatisation of Symmetric Linear Recursion

Although we have now obtained quite a number of results on the function $f(x) =_{df} axb + c$, we still have no closed representations of its extremal fixpoints. This is no surprise, since it abstracts the context-free grammar $x ::= axb|c$. In the semiring of formal languages over an alphabet, with operations union and concatenation as $+$ and \cdot , its least fixpoint is the prototypical non-regular language $\{a^n cb^n \mid n \in \mathbb{N}\}$. Hence we cannot hope to express this least fixpoint using the star operation; we need something else.

One possibility is to axiomatise the fixpoints of f directly. We follow the pattern of Kleene and omega algebras; the recursions there are the special cases $a = 1$ or $b = 1$. An axiomatic treatment of least fixpoints of general context-free recursions can be found in [ÉL05].

To have a simple notation we denote the intended least fixpoint of f by $(a|c|b)$ and axiomatise it by

$$a(a|c|b)b + c \leq (a|c|b), \quad axb + c \leq x \Rightarrow (a|c|b) \leq x.$$

Putting $b = 1$ and $a^*c = (a|c|1)$ we obtain the axioms of a weak Kleene algebra.

Lemma 4.20. If the underlying weak semiring admits countable sums and composition distributes over them then $(a|c|b) = \sum_{n \in \mathbb{N}} a^n cb^n$.

Proof. Clearly, $\sum_{n \in \mathbb{N}} a^n cb^n$ is contracted by f , which shows \leq . The converse inequation follows, since for an arbitrary fixpoint f° of f a straightforward induction shows $a^n cb^n \leq f^\circ$ for all $n \in \mathbb{N}$. \square

For the special case $c = 0$ we can prove a least fixpoint result without any assumptions on countable sums.

Lemma 4.21. Assume a weak Kleene algebra and set $e(x) =_{df} axb$. Then $\mu e = a^*0$.

Proof. We first show $(\mu e)0 = a^*(\mu e)0$. The direction (\leq) holds by $1 \leq a^*$ and (\geq) reduces by star induction to $(\mu e)0 + a(\mu e)0 \leq (\mu e)0$, i.e., $a(\mu e)0 \leq (\mu e)0$. But this holds, since $(\mu e)0 = a(\mu e)b0 \geq a(\mu e)0$.

Now we have $a^*0 \leq a^*(\mu e)0 = (\mu e)0 \leq \mu e$. The reverse inequation holds, since $e(a^*0) = aa^*0b \leq a^*0$. \square

The greatest fixpoint of f can be axiomatised together with a^ω . For a demonic setting, suitable axioms are the following:

$$a^\omega = aa^\omega b, \quad x \leq axb + c \Rightarrow x \leq a^\omega + (a|c|b).$$

Putting again $b = 1$ we obtain the axioms of a weak omega algebra.

By the omega unfold axiom all elements a^ω and hence, in particular, $\top = 1^\omega$ are left zeros; hence the above characterises demonic settings such as UTP or demonic refinement algebra [HMS06].

5. Noetherity and deterministic termination

We have already employed the convergence and divergence operators to good advantage. In this section we use them to discuss noetherian elements, i.e., elements that do not admit infinite transition paths. Subsequently this is used in the termination analysis of deterministic programs; in particular, we show that for these the difference between progressive finiteness and progressive boundedness does not arise.

5.1. Noetherian elements

The absence of infinite transition paths can be characterised as follows.

Definition 5.1. An element a of a convergence semiring is *noetherian* if $\Delta a = 1$.

It is known [DMS06b] that a is noetherian iff 0 is the only test expanded by $\langle a \rangle$, i.e., iff

$$\forall p : p \leq \langle a \rangle p \Rightarrow p \leq 0. \quad (7)$$

We now develop some useful further properties of noetherity.

Lemma 5.2. a is noetherian iff $\lceil a \leq \Delta a$.

Proof. $\lceil a \leq \Delta a \Leftrightarrow 1 \leq \neg \lceil a + \Delta a = \lceil a \rceil \Delta a = \lceil a \rceil [a] \Delta a = \lceil a a \rceil \Delta a = [a] \Delta a = \Delta a$. \square

Corollary 5.3.

1. qa is noetherian iff $q \lceil a \leq \Delta(qa)$.
2. If $q \leq \Delta a$ then qa is noetherian.

Proof. Part 1 is immediate from Lemma 5.2. Part 2 follows from part 1, since $q \lceil a \leq q \leq \Delta a \leq \Delta(qa)$ by antitony of Δ . \square

We conclude with some properties of the convergence operator.

Lemma 5.4.

1. $\Delta(qa) \geq [q](\Delta a) = \neg q + \Delta a$.
2. $[qa] \geq [aq] \Rightarrow \Delta(qa) \leq [q](\Delta a)$.

Proof.

1. By the rolling rule of fixpoint calculus and antitony of Δ ,

$$\Delta(qa) = [q](\Delta(aq)) \geq [q](\Delta a) = \neg \lceil (q \neg \Delta a) = \neg q + \Delta a.$$

2. We show that $[q](\Delta a)$ is contracted by $[qa]$: By convergence unfold, the idempotence of tests, box composition, and the assumption,

$$[q](\Delta a) = [q][a](\Delta a) = [q][qa](\Delta a) \geq [q][aq](\Delta a) = [qa][q](\Delta a). \quad \square$$

The premise of Lemma 5.4.2 reads more nicely in diamond form, viz. $\langle qa \rangle \leq \langle aq \rangle$, meaning that extensionally q is an invariant of a .

5.2. Atomic tests and their images

To prepare our result about deterministic termination we need to consider atomic tests; they abstract singleton sets of states.

Definition 5.5. Consider a partial order with least element 0 . Then a is an *atom* if it is a minimal non-zero element, i.e.,

$$\forall b : b \leq a \Rightarrow b = 0 \vee b = a.$$

An element b is a *subatom* if it is below an atom, i.e., if it is 0 or an atom.

In an ideal semiring (S, T) an *atomic test* is an atom in the Boolean algebra $\text{test}(S, T)$. We call S *test-atomistic* if $\text{test}(S, T)$ is an atomistic Boolean algebra, i.e., if arbitrary sums of atomic tests exist, every test is the sum of the atomic tests below it, and composition distributes through arbitrary sums of atomic tests.

For the remainder of this section we assume an ideal semiring with domain and codomain.

Definition 5.6. The *image* and *inverse image* of a test p under an element a are, respectively,

$$p : a =_{df} (pa)^\lceil, \quad a : p =_{df} \lceil (ap).$$

Note that $a : p = \langle a \rangle p$.

Lemma 5.7. Consider a test-atomistic semiring. Let q be an atomic test and a be deterministic. Then $q : a$ is a subatom.

Proof. If $q : a = 0$ the claim is trivial. So assume $p =_{df} q : a \neq 0$, hence $qa \neq 0$. Let $\text{At}(p)$ be the set of atomic tests below p , hence $p = \sum_{r \in \text{At}(p)} r$. We have

$$qa = qa(qa)^\top = qap = qa \sum_{r \in \text{At}(p)} r = \sum_{r \in \text{At}(p)} qar.$$

Since determinacy is downward closed (see Lemma 17 of [DM01]), also qa is deterministic. Hence for all $r_1, r_2 \in \text{At}(p)$ with $r_1 \neq r_2$ we have $\langle qa \rangle r_1 \cdot \langle qa \rangle r_2 = 0$. By atomicity of q and domain axiom (d2) we have $\langle qa \rangle r_i \in \{0, q\}$. Therefore for at most one $r \in \text{At}(p)$ we have $\langle qa \rangle r \neq 0$, i.e., $qar \neq 0$. In this case, $0 \neq (qar)^\top = (qa)^\top r \leq r$ and $qa = qar$, hence, using atomicity of r ,

$$q : a = (qa)^\top = (qar)^\top = r. \quad \square$$

Lemma 5.8. If $q \leq \lceil a$ then $q \leq a : (q : a)$.

Proof. $a : (q : a) = \lceil a(qa)^\top \rceil \geq \lceil qa(qa)^\top \rceil = \lceil qa \rceil = q^\top a = q. \quad \square$

Corollary 5.9. Let q be an atomic test with $qa \neq 0$. Then $q \leq a : (q : a)$.

Proof. Since $qa \neq 0$, we have $0 \neq \lceil qa \rceil \leq q$ and therefore $q = \lceil qa \rceil = q^\top a$, i.e., $q \leq \lceil a$, and Lemma 5.8 applies. \square

5.3. Deterministic termination

Using atoms we can sharpen our statements about noetherity and restriction.

Lemma 5.10. Let qa be noetherian and q an atom. Then $qaq = 0$ and hence $(qa)^n = 0$ for all $n \geq 2$.

Proof. Suppose $qaq \neq 0$, hence $\lceil qaq \rceil \neq 0$. By domain axiom (d2), $\lceil qaq \rceil \leq q$. By atomicity of q therefore $\lceil qaq \rceil = q$. Hence $q = \langle qa \rangle q$ and $q \neq 0$, a contradiction to noetherity of qa . \square

The final lemma shows that for deterministic elements the difference between progressive boundedness and progressive finiteness does not arise. Extending Lemma 4.19 we can now characterise single states as atoms and thus, in the noetherian case, show the existence of an upper bound on the number of iterations starting from such a state.

Lemma 5.11. Consider a test-atomistic convergence semiring. If a is deterministic and noetherian and q is an atomic test then there is an $n \in \mathbb{N}$ with $qa^n = 0$.

Proof. Assume that $qa^n \neq 0$ for all $n \in \mathbb{N}$ and set $r_n =_{df} q : a^n$. Since determinacy is closed under composition (see Lemma 22 of [DM01]), all a^n are deterministic, too, and hence by Lemma 5.7 all r_n are atoms. Moreover, a straightforward calculation shows $r_{n+1} = r_n : a$. Hence, by Corollary 5.9, we have

$$r_n \leq a : r_{n+1}. \quad (\star)$$

We now show that $p =_{df} \sum_{n \geq 0} r_n$ is expanded by $\langle a \rangle$. By universal distributivity of test multiplication and of domain, an index shift and (\star) ,

$$a : p = \sum_{n \geq 0} a : r_n \geq \sum_{n > 0} a : r_n = \sum_{n \geq 0} a : r_{n+1} \geq \sum_{n \geq 0} r_n \geq p.$$

Moreover, $p \neq 0$ since $q = r_0 \leq p$. By (7) this contradicts noetherity of a . \square

6. Conclusion

The treatment has shown that almost all of the standard theory of normal designs carries over to our more general algebraic setting. Moreover, we have presented a generalisation of the fixpoint theorem 3.1.6

of [HH98] that allows an alternative derivation of the omega operator on designs. It should be noted that the operations of complement and meet are not required for all semiring elements but only on the conditions.

The combination of the approach using ideal semirings with the matrix calculus of [Möl06] has led to considerably simpler reasoning, since well-known results about the star and omega iterations of matrices can be re-used. Recently it has also been shown [HMS06] that designs and prescriptions form a demonic refinement algebra in the sense of von Wright [Wri04]; thus that framework can be re-used, too.

Finally, we have shown that the normal designs can be equipped with box and diamond operators. While the box on the underlying semiring is the abstract counterpart of the wlp operator, the one on designs corresponds to wp. Hence the general soundness and completeness proof for the associated Hoare logic, originally developed for a partial correctness framework, can directly be applied to normal designs (see [MS06] for details).

It is to be hoped that the generalised results will also be of use for handling trace semantics and other semantical models, thus dealing with healthiness conditions such as (R1)–(R3) of UTP in a purely algebraic fashion. The presented method could also serve as a model for the extension by parameters that describe further observations as proposed in [HH98].

Acknowledgement. We are grateful to Peter Höfner for valuable remarks.

References

- [Aar92] C. J. Aarts. Galois connections presented calculationally. Master’s thesis, Department of Mathematics and Computing Science, Eindhoven University of Technology, 1992.
- [ABB⁺95] C. J. Aarts, R. C. Backhouse, E. A. Boiten, H. Doornbos, N. van Gasteren, R. van Geldrop, P. F. Hoogendijk, E. Voermans, and J. van der Woude. Fixed-point calculus. *Information Processing Letters*, 53(3):131–136, 10 February 1995.
- [Bau76] F. L. Bauer. Programming as an evolutionary process. In *Proceedings of the 2nd International Conference on Software Engineering*, pages 223–234. IEEE Computer Society Press, 1976.
- [Coh00] E. Cohen. Separation and reduction. In R. Backhouse and J. N. Oliveira, editors, *Mathematics of Program Construction*, number 1837 in Lecture Notes in Computer Science, pages 45–59. Springer-Verlag, 2000.
- [Dij76] E. W. Dijkstra. *A Discipline of Programming*. Prentice Hall International, 1976.
- [DM01] J. Desharnais and B. Möller. Characterizing determinacy in Kleene algebras. *Information Sciences*, 139(3):253–273, December 2001.
- [DMS06a] J. Desharnais, B. Möller, and G. Struth. Algebraic notions of termination. Report 2006-23, Institut für Informatik, Universität Augsburg, October 2006.
- [DMS06b] J. Desharnais, B. Möller, and G. Struth. Kleene algebra with domain. *ACM Transactions on Computational Logic*, 7(4):798–833, October 2006.
- [DMT06] J. Desharnais, B. Möller, and F. Tchier. Kleene under a modal demonic star. *Journal of Logic and Algebraic Programming (Special Issue on Relation Algebra and Kleene Algebra)*, 66(2):127–160, February–March 2006.
- [Dun01] S. Dunne. Recasting Hoare and He’s unifying theory of programs in the context of general correctness. In A. Butterfield, G. Strong, and C. Pahl, editors, *5th Irish Workshop on Formal Methods*, EWiC. The British Computer Society, 2001.
- [ÉL05] Z. Ésik and H. Leiß. Algebraically complete semirings and Greibach normal form. *Annals of Pure and Applied Logic*, 3(1–3):173–203, May 2005.
- [GM06] W. Guttmann and B. Möller. Modal design algebra. In S. Dunne and W. Stoddart, editors, *Unifying Theories of Programming*, number 4010 in Lecture Notes in Computer Science, pages 236–256. Springer-Verlag, 2006.
- [HH98] C. A. R. Hoare and J. He. *Unifying theories of programming*. Prentice Hall Europe, 1998.
- [HMS06] P. Höfner, B. Möller, and K. Solin. Omega algebra, demonic refinement algebra and commands. In R. Schmidt, editor, *Relations and Kleene Algebra in Computer Science*, number 4136 in Lecture Notes in Computer Science, pages 222–234. Springer-Verlag, 2006.
- [HW93] U. Hebisch and H. J. Weinert. *Halbringe*. Teubner, 1993.
- [Koz94] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, May 1994.
- [Koz97] D. Kozen. Kleene algebra with tests. *ACM Transactions on Programming Languages and Systems*, 19(3):427–443, May 1997.
- [KP00] D. Kozen and M.-C. Patron. Certification of compiler optimizations using Kleene algebra with tests. In J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. M. Pereira, Y. Sagiv, and P. J. Stuckey, editors, *Computational Logic – CL 2000*, number 1861 in Lecture Notes in Artificial Intelligence, pages 568–582. Springer-Verlag, 2000.
- [MD06] V. Mathieu and J. Desharnais. Verification of pushdown systems using omega algebra with domain. In W. MacCaull, M. Winter, and I. Düntsch, editors, *Relational Methods in Computer Science 2005*, number 3929 in Lecture Notes in Computer Science, pages 188–199. Springer-Verlag, 2006.

- [Möl04] B. Möller. Lazy Kleene algebra. In D. Kozen, editor, *Mathematics of Program Construction*, number 3125 in Lecture Notes in Computer Science, pages 252–273. Springer-Verlag, 2004.
- [Möl06] B. Möller. The linear algebra of UTP. In T. Uustalu, editor, *Mathematics of Program Construction*, number 4014 in Lecture Notes in Computer Science, pages 338–358. Springer-Verlag, 2006.
- [MS06] B. Möller and G. Struth. WP is WLP. In W. MacCaull, M. Winter, and I. Düntsch, editors, *Relational Methods in Computer Science 2005*, number 3929 in Lecture Notes in Computer Science, pages 200–211. Springer-Verlag, 2006.
- [SS89] G. Schmidt and T. Ströhlein. *Relationen und Graphen*. Springer-Verlag, 1989.
- [Wri04] J. von Wright. Towards a refinement algebra. *Science of Computer Programming*, 51(1–2):23–45, May 2004.