Extended Conscriptions Algebraically

Walter Guttmann

Department of Computer Science and Software Engineering,
University of Canterbury, New Zealand
walter.guttmann@canterbury.ac.nz

Abstract. Conscriptions are a model of sequential computations with assumption/commitment specifications in which assumptions can refer to final states, not just to initial states. We show that they instantiate existing algebras for iteration and infinite computations. We use these algebras to derive an approximation order for conscriptions and one for extended conscriptions, which additionally represent aborting executions. We give a new computation model which generalises extended conscriptions and apply the algebraic techniques for a unified treatment.

1 Introduction

Various relational models have been proposed for sequential computations, for example, in [15, 4, 13, 9, 11]. The most precise of these models can represent finite, infinite and aborting executions independently of each other. Less precise models ignore certain kinds of executions if others are present, but have simpler descriptions. All of these models have two properties in common:

- They represent only the initial and final states of computations and disregard the intermediate states.
- Whenever they represent infinite or aborting executions, they only record if such executions are present or absent from each starting state.

The latter in particular means that there is no notion of a final state when it comes to infinite or aborting executions. For several reasons it is desirable to eliminate this restriction:

- Its removal provides a basis for more detailed models that involve time.
- Different interpretations of a model may replace infinite or aborting executions with other kinds of executions for which final states are observable.
- A final state may be observable for a blocking computation which waits for external input that never arrives.
- A final state may be observable for an infinite execution, for example, if it stabilises and continues as an endless loop that does not change the state.
- A final state may be observable for an aborting execution, namely the last state before the execution aborts.
- The restriction is a technical constraint on the model: some entries in the matrix of relations that represent a computation have to be vectors, which are special relations. It is natural to generalise them to arbitrary relations.

Two models that lift the restriction have been described in [5]. *Conscriptions* represent infinite and finite executions independently, but have no notion of aborting executions. There is no restriction on the final states in the case of infinite executions. *Extended conscriptions* represent aborting, infinite and finite executions independently. There is no restriction on the final states in the case of aborting executions, but the restriction to vectors still applies for infinite executions. For both models, basic algebraic properties of sequential composition and non-deterministic choice have been derived.

The present paper extends this investigation by considering the following questions:

- How is recursion defined for the new computation models?
- Can extended conscriptions be further generalised by lifting the restriction on infinite executions?
- What algebraic structures underlie iteration in these models?
- Can all of these models be captured by a unifying algebraic theory?

We provide the following contributions:

- Several new computation models, the most precise of which represents finite, infinite and aborting executions independently of each other with no restrictions on the final states for any kind of execution.
- Instances of previously introduced algebras for iteration and infinite computations for each of these models. Consequences are a unified description of recursion and applications including separation and refinement theorems and various program transformations for the new models.
- An approximation order for the new models, which is used to define the semantics of recursion. The approximation orders known from previous models do not generalise in a direct way. Instead, the new models turn out to satisfy axioms of algebras previously developed for non-strict computations [12]; we use these algebras to obtain the approximation order.

Section 2 gives the basic algebraic structures referred to in the remainder of this paper. Section 3 derives an approximation order for conscriptions and shows how conscriptions instantiate algebraic structures to describe iterations and infinite executions. Section 4 applies the method of Section 3 to extended conscriptions. Section 5 introduces the most precise computation model, applies the method of Section 3 to it and shows how to specialise it to obtain other models.

2 Algebraic Structures for Sequential Computations

In this section we axiomatise the operations of non-deterministic choice, conjunction and sequential composition, the infinite executions of a computation and various forms of iteration featured by many computation models.

A *lattice-ordered semiring* is an algebraic structure $(S, +, \curlywedge, \cdot, 0, 1, \top)$ such that the following axioms hold:

$$\begin{array}{rr} x + (y + z) = (x + y) + z & x \curlywedge (y \curlywedge z) = (x \curlywedge y) \curlywedge z \\ x + y = y + x & x \curlywedge y = y \curlywedge x \\ x + x = x & x \curlywedge x = x \\ 0 + x = x & \top \curlywedge x = x \\ x + (y \curlywedge z) = (x + y) \curlywedge (x + z) & x \curlywedge (y + z) = (x \curlywedge y) + (x \curlywedge z) \\ x + (x \curlywedge y) = x & x \curlywedge (x + y) = x \\ 1 \cdot x = x & x \cdot (y + z) = (x \cdot y) + (x \cdot z) \\ x \cdot 1 = x & (x + y) \cdot z = (x \cdot z) + (y \cdot z) \\ x \cdot (y \cdot z) = (x \cdot y) \cdot z & 0 \cdot x = 0 \end{array}$$

The axioms not involving $\cdot$ make up a *bounded distributive lattice* $(S, +, \curlywedge, 0, \top)$. The axioms not involving $\curlywedge$ make up an idempotent semiring without right annihilator $(S, +, \cdot, 0, 1)$, simply called *semiring* in the remainder of this paper. In particular, $x \cdot 0 = 0$ is not an axiom. The *lattice order* $x \leq y \Leftrightarrow x + y = y \Leftrightarrow x \curlywedge y = x$ has least element 0, greatest element $\top$, least upper bound $+$ and greatest lower bound $\curlywedge$. The operations $+$, $\curlywedge$ and $\cdot$ are $\leq$-isotone. We abbreviate $x \cdot y$ as xy.

In many computation models the operation $+$ represents non-deterministic choice, the operation $\curlywedge$ conjunction, the operation $\cdot$ sequential composition, 0 the computation with no executions, 1 the computation that does not change the state, $\top$ the computation with all executions, and $\leq$ the refinement relation.

The following algebras capture various fixpoints of the function $\lambda x.yx + z$, which are useful to describe iteration.

A *Kleene algebra* $(S, +, \cdot, ^*, 0, 1)$ adds to a semiring an operation * with the following unfold and induction axioms [16]:

$$\begin{array}{ll} 1 + yy^* \leq y^* & z + yx \leq x \Rightarrow y^*z \leq x \\ 1 + y^*y \leq y^* & z + xy \leq x \Rightarrow zy^* \leq x \end{array}$$

It follows that y^*z is the $\leq$-least fixpoint of $\lambda x.yx + z$ and that zy^* is the $\leq$-least fixpoint of $\lambda x.xy + z$. The operation * is $\leq$-isotone.

An *omega algebra* $(S, +, \cdot, ^*, ^\omega, 0, 1)$ adds to a Kleene algebra an operation $^\omega$ with the following unfold and induction axioms [2, 17]:

$$yy^\omega = y^\omega \qquad x \leq yx + z \Rightarrow x \leq y^\omega + y^*z$$

It follows that $y^\omega + y^*z$ is the $\leq$-greatest fixpoint of $\lambda x.yx + z$. In particular, $\top = 1^\omega$ is the $\leq$-greatest element. The operation $^\omega$ is $\leq$-isotone.

For computation models that require different fixpoints of $\lambda x.yx + z$, we use the following generalisations of Kleene algebras.

An *extended binary itering* $(S, +, \cdot, \star, 0, 1)$ adds to a semiring a binary operation $\star$ with the following axioms [10]:

$$\begin{array}{rr} (x + y) \star z = (x \star y) \star (x \star z) & x \star (y + z) = (x \star y) + (x \star z) \\ (xy) \star z = z + x((yx) \star (yz)) & (x \star y)z \leq x \star (yz) \end{array}$$

$$zx \leq y(y \star z) + w \Rightarrow z(x \star v) \leq y \star (zv + w(x \star v))$$
$$xz \leq z(y \star 1) + w \Rightarrow x \star (zv) \leq z(y \star v) + (x \star (w(y \star v)))$$
$$w(x \star (yz)) \leq (w(x \star y)) \star (w(x \star y)z)$$

It follows that $y \star z$ is a fixpoint of $\lambda x.yx + z$. The operation $\star$ is $\leq$-isotone. The element $y \star z$ corresponds to iterating y an unspecified number of times, followed by a single occurrence of z. This may involve an infinite number of iterations of y.

In models that satisfy $(x \star y)z = x \star (yz)$, the binary itering operation specialises to a unary operation $^\circ$ with the following simpler axioms. An *itering* $(S, +, \cdot, ^\circ, 0, 1)$ adds to a semiring an operation $^\circ$ with the sumstar and productstar equations of [3] and two simulation axioms [9]:

$$(x + y)^\circ = (x^\circ y)^\circ x^\circ \qquad zx \leq yy^\circ z + w \Rightarrow zx^\circ \leq y^\circ(z + wx^\circ)$$
$$(xy)^\circ = 1 + x(yx)^\circ y \qquad xz \leq zy^\circ + w \Rightarrow x^\circ z \leq (z + x^\circ w)y^\circ$$

It follows that $y^\circ z$ is a fixpoint of $\lambda x.yx+z$ and that zy° is a fixpoint of $\lambda x.xy+z$. The operation $^\circ$ is $\leq$-isotone.

Every Kleene algebra is an itering using $x^\circ = x^*$. Every omega algebra is an itering using $x^\circ = x^\omega 0 + x^*$. Every itering is an extended binary itering using $x \star y = x^\circ y$. Further instances and consequences of iterings are given in [9, 10].

We finally describe the set of states $n(x)$ from which a computation x has infinite executions. Sets are represented as tests, that is, as elements ≤ 1. The axioms have been developed in [12] for a unified treatment of strict and non-strict computations; the latter can produce defined outputs from undefined inputs. An axiomatisation for strict computations has been given in [9].

An *n-algebra* $(S, +, \curlywedge, \cdot, n, 0, 1, \mathsf{L}, \top)$ adds to a lattice-ordered semiring an operation $n : S \to S$ and a constant L with the following axioms:

$$(n1)\quad n(x) + n(y) = n(n(x)\top + y) \qquad (n6)\quad n(x) \leq n(\mathsf{L}) \curlywedge 1$$
$$(n2)\quad n(x)n(y) = n(n(x)y) \qquad (n7)\quad n(x)\mathsf{L} \leq x$$
$$(n3)\quad n(x)n(x + y) = n(x) \qquad (n8)\quad n(\mathsf{L})x \leq xn(\mathsf{L})$$
$$(n4)\quad n(\mathsf{L})x = (x \curlywedge \mathsf{L}) + n(\mathsf{L}0)x \qquad (n9)\quad xn(y)\top \leq x0 + n(xy)\top$$
$$(n5)\quad x\mathsf{L} = x0 + n(x\mathsf{L})\mathsf{L} \qquad (n10)\quad x\top y \curlywedge \mathsf{L} \leq x\mathsf{L}y$$

An *n-omega algebra* $(S, +, \curlywedge, \cdot, n, ^*, ^\omega, 0, 1, \mathsf{L}, \top)$ adds the following axioms to an n-algebra $(S, +, \curlywedge, \cdot, n, 0, 1, \mathsf{L}, \top)$ and an omega algebra $(S, +, \cdot, ^*, ^\omega, 0, 1)$:

$$(n11)\quad n(\mathsf{L})x^\omega \leq x^* n(x^\omega)\top \qquad (n12)\quad x\mathsf{L} \leq x\mathsf{L}x\mathsf{L}$$

The constant L represents the endless loop, that is, the computation with all infinite executions. A constant for the computation with all aborting executions is not provided.

3 Conscriptions

The state space A of a sequential computation is given by the values of the program variables. A computation is thus represented as a relation R on A, that

is, as a subset of the Cartesian product $A \times A$. A pair $(x, x') \in R$ signifies that there is a finite execution of the program which starts in state x and ends in state x'; in other words, x' is a possible output for input x. Several outputs for the same input indicate non-determinism.

This simple model is sufficient for partial correctness, but does not provide means to represent infinite executions. For the latter, extended models can be used which represent computations as assumption/commitment pairs. The assumption part specifies the conditions under which termination is guaranteed and the commitment part specifies the effect if the program terminates. The conditions of termination traditionally refer only to the pre-state x of the computation, not to its post-state x'. As discussed in the introduction, conscriptions are introduced in [5] to eliminate this restriction.

A *conscription* is a 2×2 matrix whose entries are relations over A. The matrix has the following form:

$$\begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}$$

The entries in the top row are the identity relation 1 and the empty relation 0, respectively, for each conscription. Only the entries in the bottom row can vary: the relation Q represents the complement of the assumption (the infinite executions) and the relation R represents the commitment (the finite executions). No restrictions are placed on Q and R. This is in contrast to other models such as the designs of [15], the prescriptions of [4] and the extended designs of [13]. They require Q to be a vector, that is, in those models Q relates every state x either to all states x' or to no state.

Sequential composition and non-deterministic choice of conscriptions are given by matrix product and componentwise union, respectively. The refinement order on conscriptions is the componentwise set inclusion order. The computation which does not change the state and the endless loop are represented by the conscriptions

$$\mathsf{skip} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \mathsf{L} = \begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix}$$

where $\top$ is the universal relation. The conscription skip is a neutral element of sequential composition and L is a left annihilator.

We wish to define the semantics of recursion by least fixpoints in a suitable approximation order $\sqsubseteq$. Because the endless loop L has to be the $\sqsubseteq$-least element, the refinement order cannot be used for approximation.

3.1 An Approximation Order for Conscriptions: Two Attempts

In the following we discuss two attempts to define an approximation order for conscriptions and the reasons why they fail. The first attempt is to take the approximation order of prescriptions [7, 6]. Prescriptions correspond to a subset of conscriptions in which the assumption component Q is a vector. It is therefore natural to assume that the approximation order for conscriptions specialises to

the approximation order for prescriptions when restricted to this subset. The approximation order $\sqsubseteq_1$ on conscriptions would accordingly be defined as

$$\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} \sqsubseteq_1 \begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix} \;\Leftrightarrow\; Q_2 \subseteq Q_1 \wedge R_1 \subseteq R_2 \subseteq R_1 \cup Q_1$$

The intuition underlying $\sqsubseteq_1$ is that in states with infinite executions, finite executions can be added but only such that have the same output. A problem with $\sqsubseteq_1$ is that sequential composition from the right is not $\sqsubseteq_1$-isotone. Namely,

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \sqsubseteq_1 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

but

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \not\sqsubseteq_1 \begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix}$$

The second attempt converts conscriptions to prescriptions and takes their order:

$$\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} \sqsubseteq_2 \begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix} \;\Leftrightarrow\; \begin{pmatrix} 1 & 0 \\ Q_1\top & R_1 \end{pmatrix} \sqsubseteq_1 \begin{pmatrix} 1 & 0 \\ Q_2\top & R_2 \end{pmatrix}$$

The assumptions are converted to vectors by composing them with $\top$. Hence the resulting prescription has an infinite execution from state x if the original conscription has any infinite execution starting in x. The intuition underlying $\sqsubseteq_2$ is that in states with any infinite executions, any finite execution can be added. A problem with $\sqsubseteq_2$ is that it is not antisymmetric. Namely,

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \sqsubseteq_2 \begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} \sqsubseteq_2 \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$$

More generally, it is inconsistent to assume all of the following four properties of an approximation relation $\sqsubseteq$ for conscriptions:

- $\sqsubseteq$ is a partial order,
- sequential composition from the right is $\sqsubseteq$-isotone,
- $\begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} \sqsubseteq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$,
- $\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \sqsubseteq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

This is because they would imply

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} \sqsubseteq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix}$$

and

$$\begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \sqsubseteq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$$

and therefore

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix}$$

The approximation relation we subsequently derive satisfies the first three properties of the above list, but not the last one.

3.2 An Approximation Order for Conscriptions

To obtain a suitable approximation order for conscriptions, we use the algebraic method developed in [9, 12]. We first observe the following basic structure.

Theorem 1. *Conscriptions form a lattice-ordered semiring with the operations*

$$\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix} + \begin{pmatrix}1 & 0\\ Q_2 & R_2\end{pmatrix} = \begin{pmatrix}1 & 0\\ Q_1 \cup Q_2 & R_1 \cup R_2\end{pmatrix} \qquad 0 = \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$$
$$\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix} \curlywedge \begin{pmatrix}1 & 0\\ Q_2 & R_2\end{pmatrix} = \begin{pmatrix}1 & 0\\ Q_1 \cap Q_2 & R_1 \cap R_2\end{pmatrix} \qquad \top = \begin{pmatrix}1 & 0\\ \top & \top\end{pmatrix}$$
$$\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix} \cdot \begin{pmatrix}1 & 0\\ Q_2 & R_2\end{pmatrix} = \begin{pmatrix}1 & 0\\ Q_1 \cup R_1 Q_2 & R_1 R_2\end{pmatrix} \qquad 1 = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

Proof. The claims follow by simple calculations since $+$ and $\curlywedge$ are defined componentwise and $\cdot$ is the matrix product. □

The lattice order $\leq$ on conscriptions therefore amounts to componentwise inclusion. The algebraic approach to approximation is based on the operation n that represents the infinite executions of a computation as a test, that is, as an element ≤ 1. For conscriptions, tests take the form

$$\begin{pmatrix}1 & 0\\ 0 & R\end{pmatrix}$$

where $R \subseteq 1$. The operation n that maps semiring elements to tests is characterised by the Galois connection

$$n(x)\mathsf{L} \leq y \;\Leftrightarrow\; n(x) \leq n(y)$$

Hence $n(y)$ is the greatest test whose composition with L is below y. We use this Galois connection to obtain a definition of n for conscriptions. Because the result of n is a test, assume that n is given by the general form

$$n\begin{pmatrix}1 & 0\\ Q & R\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & f(Q,R)\end{pmatrix}$$

using a function f that maps its argument relations Q and R to a relation below the identity 1, that is, $f(Q,R) \subseteq 1$. By the Galois connection,

$$\begin{aligned}
& f(Q_1,R_1) \subseteq f(Q_2,R_2) \;\Leftrightarrow\; \begin{pmatrix}1 & 0\\ 0 & f(Q_1,R_1)\end{pmatrix} \leq \begin{pmatrix}1 & 0\\ 0 & f(Q_2,R_2)\end{pmatrix}\\
\Leftrightarrow\;& n\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix} \leq n\begin{pmatrix}1 & 0\\ Q_2 & R_2\end{pmatrix} \;\Leftrightarrow\; n\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix}\mathsf{L} \leq \begin{pmatrix}1 & 0\\ Q_2 & R_2\end{pmatrix}\\
\Leftrightarrow\;& \begin{pmatrix}1 & 0\\ 0 & f(Q_1,R_1)\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & 0\end{pmatrix} \leq \begin{pmatrix}1 & 0\\ Q_2 & R_2\end{pmatrix} \;\Leftrightarrow\; \begin{pmatrix}1 & 0\\ f(Q_1,R_1)\top & 0\end{pmatrix} \leq \begin{pmatrix}1 & 0\\ Q_2 & R_2\end{pmatrix}\\
\Leftrightarrow\;& f(Q_1,R_1)\top \subseteq Q_2 \;\Leftrightarrow\; f(Q_1,R_1) \subseteq \overline{\overline{Q_2}\top} \;\Leftrightarrow\; f(Q_1,R_1) \subseteq \overline{\overline{Q_2}\top} \cap 1
\end{aligned}$$

The operation $^-$ is the relational complement. The next-to-last step is a consequence of a Schröder equivalence.

The above calculation suggests the definition $f(Q,R) = \overline{\overline{Q}\top} \cap 1$. A simple rearrangement of the calculation shows that this satisfies the Galois connection. We therefore define

$$n\begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q}\top} \cap 1 \end{pmatrix}$$

In [9] we give an approximation order in terms of n that covers a range of models of strict computations including prescriptions and extended designs. The present definition of n for conscriptions does not satisfy the axioms $n(x+y) = n(x)+n(y)$ and $xn(y)\mathsf{L} = x0 + n(xy)\mathsf{L}$ used there. However, n satisfies the weaker axioms given in [12], which have been developed to uniformly describe strict and non-strict computations.

Theorem 2. *Conscriptions form an n-algebra with the operations*

$$n\begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q}\top} \cap 1 \end{pmatrix} \qquad \mathsf{L} = \begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix}$$

Proof. Observe that $n(\mathsf{L}) = 1$ and $\mathsf{L}x = \mathsf{L}$; this implies axioms $(n4)$, $(n6)$ and $(n8)$. The remaining axioms are shown as follows. See [20] for properties of relations used in the calculations.

$(n1)$

$$\begin{aligned}
&n\left(n\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix}\right) \\
&= n\left(\begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q_1}\top} \cap 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix}\right) = n\begin{pmatrix} 1 & 0 \\ \overline{\overline{Q_1}\top} \cup Q_2 & \overline{\overline{Q_1}\top} \cup R_2 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{\overline{Q_1}\top \cup Q_2\top} \cap 1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q_1}\top \cap \overline{Q_2}\top} \cap 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & (\overline{\overline{Q_1}\top} \cup \overline{\overline{Q_2}\top}) \cap 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q_1}\top} \cap 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q_2}\top} \cap 1 \end{pmatrix} = n\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} + n\begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix}
\end{aligned}$$

$(n2)$

$$\begin{aligned}
&n\left(n\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix}\right) = n\left(\begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q_1}\top} \cap 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix}\right) \\
&= n\begin{pmatrix} 1 & 0 \\ (\overline{\overline{Q_1}\top} \cap 1)Q_2 & (\overline{\overline{Q_1}\top} \cap 1)R_2 \end{pmatrix} = n\begin{pmatrix} 1 & 0 \\ \overline{\overline{Q_1}\top} \cap Q_2 & \overline{\overline{Q_1}\top} \cap R_2 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{\overline{Q_1}\top \cap Q_2\top} \cap 1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q_1}\top} \cap \overline{\overline{Q_2}\top} \cap 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & (\overline{\overline{Q_1}\top} \cap 1)(\overline{\overline{Q_2}\top} \cap 1) \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q_1}\top} \cap 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q_2}\top} \cap 1 \end{pmatrix} = n\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} n\begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix}
\end{aligned}$$

($n3$) The calculation uses that $\overline{\overline{Q_1}\top} \subseteq \overline{\overline{Q_1 \cup Q_2}\top}$:

$$
\begin{aligned}
&n\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix} n\left(\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix} + \begin{pmatrix}1 & 0\\ Q_2 & R_2\end{pmatrix}\right) = n\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix} n\begin{pmatrix}1 & 0\\ Q_1 \cup Q_2 & R_1 \cup R_2\end{pmatrix}\\
&= \begin{pmatrix}1 & 0\\ 0 & \overline{\overline{Q_1}\top} \cap 1\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & \overline{\overline{Q_1 \cup Q_2}\top} \cap 1\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & (\overline{\overline{Q_1}\top} \cap 1)(\overline{\overline{Q_1 \cup Q_2}\top} \cap 1)\end{pmatrix}\\
&= \begin{pmatrix}1 & 0\\ 0 & \overline{\overline{Q_1}\top} \cap \overline{\overline{Q_1 \cup Q_2}\top} \cap 1\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & \overline{\overline{Q_1}\top} \cap 1\end{pmatrix} = n\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix}
\end{aligned}
$$

($n5$) The calculation uses that $\overline{\overline{Q}\top} \subseteq Q$:

$$
\begin{aligned}
&\begin{pmatrix}1 & 0\\ Q & R\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} + n\left(\begin{pmatrix}1 & 0\\ Q & R\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & 0\end{pmatrix}\right)\begin{pmatrix}1 & 0\\ \top & 0\end{pmatrix}\\
&= \begin{pmatrix}1 & 0\\ Q & 0\end{pmatrix} + n\begin{pmatrix}1 & 0\\ Q \cup R\top & 0\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & 0\end{pmatrix} = \begin{pmatrix}1 & 0\\ Q & 0\end{pmatrix} + \begin{pmatrix}1 & 0\\ 0 & \overline{\overline{Q \cup R\top}\top} \cap 1\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & 0\end{pmatrix}\\
&= \begin{pmatrix}1 & 0\\ Q & 0\end{pmatrix} + \begin{pmatrix}1 & 0\\ \overline{\overline{Q \cup R\top}\top} & 0\end{pmatrix} = \begin{pmatrix}1 & 0\\ Q \cup \overline{\overline{Q}\top \cap \overline{R\top}} & 0\end{pmatrix} = \begin{pmatrix}1 & 0\\ Q \cup \overline{\overline{Q}\top} \cup R\top & 0\end{pmatrix}\\
&= \begin{pmatrix}1 & 0\\ Q \cup R\top & 0\end{pmatrix} = \begin{pmatrix}1 & 0\\ Q & R\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & 0\end{pmatrix}
\end{aligned}
$$

($n7$)

$$
n\begin{pmatrix}1 & 0\\ Q & R\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & 0\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & \overline{\overline{Q}\top} \cap 1\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & 0\end{pmatrix} = \begin{pmatrix}1 & 0\\ \overline{\overline{Q}\top} & 0\end{pmatrix} \leq \begin{pmatrix}1 & 0\\ Q & R\end{pmatrix}
$$

($n9$) The calculation uses that $R_1\overline{\overline{Q_2}\top} \subseteq \overline{\overline{R_1 Q_2}\top} \subseteq \overline{\overline{Q_1 \cup R_1 Q_2}\top}$:

$$
\begin{aligned}
&\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix} n\begin{pmatrix}1 & 0\\ Q_2 & R_2\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & \top\end{pmatrix} = \begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & \overline{\overline{Q_2}\top} \cap 1\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & \top\end{pmatrix}\\
&= \begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix}\begin{pmatrix}\overline{\overline{Q_2}\top} & \overline{\overline{Q_2}\top}\end{pmatrix} = \begin{pmatrix}1 & 0\\ Q_1 \cup R_1\overline{\overline{Q_2}\top} & R_1\overline{\overline{Q_2}\top}\end{pmatrix}\\
&= \begin{pmatrix}1 & 0\\ Q_1 & 0\end{pmatrix} + \begin{pmatrix}\overline{\overline{Q_2}\top} & \overline{\overline{Q_2}\top}\\ R_1\overline{\overline{Q_2}\top} & R_1\overline{\overline{Q_2}\top}\end{pmatrix} \leq \begin{pmatrix}1 & 0\\ Q_1 & 0\end{pmatrix} + \begin{pmatrix}\overline{\overline{R_1 Q_2}\top} & \overline{\overline{R_1 Q_2}\top}\\ \overline{\overline{R_1 Q_2}\top} & \overline{\overline{R_1 Q_2}\top}\end{pmatrix}\\
&\leq \begin{pmatrix}1 & 0\\ Q_1 & 0\end{pmatrix} + \begin{pmatrix}\overline{\overline{Q_1 \cup R_1 Q_2}\top} & \overline{\overline{Q_1 \cup R_1 Q_2}\top}\\ \overline{\overline{Q_1 \cup R_1 Q_2}\top} & \overline{\overline{Q_1 \cup R_1 Q_2}\top}\end{pmatrix}\\
&= \begin{pmatrix}1 & 0\\ Q_1 & 0\end{pmatrix} + \begin{pmatrix}1 & 0\\ 0 & \overline{\overline{Q_1 \cup R_1 Q_2}\top} \cap 1\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & \top\end{pmatrix}\\
&= \begin{pmatrix}1 & 0\\ Q_1 & 0\end{pmatrix} + n\begin{pmatrix}1 & 0\\ Q_1 \cup R_1 Q_2 & R_1 R_2\end{pmatrix}\begin{pmatrix}1 & 0\\ \top & \top\end{pmatrix}\\
&= \begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} + n\left(\begin{pmatrix}1 & 0\\ Q_1 & R_1\end{pmatrix}\begin{pmatrix}1 & 0\\ Q_2 & R_2\end{pmatrix}\right)\begin{pmatrix}1 & 0\\ \top & \top\end{pmatrix}
\end{aligned}
$$

($n10$)

$$
\begin{aligned}
&\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix}\begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix}^{\curlywedge}\begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} \\
=\ &\begin{pmatrix} 1 & 0 \\ Q_1 \cup R_1\top & R_1\top \end{pmatrix}\begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix}^{\curlywedge}\begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} \\
=\ &\begin{pmatrix} 1 & 0 \\ Q_1 \cup R_1\top \cup R_1\top Q_2 & R_1\top R_2 \end{pmatrix}^{\curlywedge}\begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} \\
=\ &\begin{pmatrix} 1 & 0 \\ Q_1 \cup R_1\top & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix}
\end{aligned}
$$

□

As a consequence, we can use the approximation order given in [12]:

$$x \sqsubseteq y \ \Leftrightarrow\ x \le y + \mathsf{L} \wedge n(\mathsf{L})y \le x + n(x)\top$$

For conscriptions, $n(\mathsf{L}) = 1$ holds, whence the order elaborates to

$$
\begin{aligned}
&\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} \sqsubseteq \begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix} \\
\Leftrightarrow\ &\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} \le \begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ \top & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \top & R_2 \end{pmatrix} \wedge \\
&\begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix} \le \begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} + n\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix} \\
&\qquad = \begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{Q_1}\top} \cap 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix} \\
&\qquad = \begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ \overline{\overline{Q_1}\top} & \overline{\overline{Q_1}\top} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \cup \overline{\overline{Q_1}\top} \end{pmatrix} \\
\Leftrightarrow\ &Q_2 \subseteq Q_1 \wedge R_1 \subseteq R_2 \subseteq R_1 \cup \overline{\overline{Q_1}\top}
\end{aligned}
$$

The relation $\overline{\overline{Q_1}\top}$ is a vector that represents the states where all infinite executions are present. The intuition underlying the approximation order is that in states with all infinite executions, any finite execution can be added. In states where at least one infinite execution is missing, no finite execution can be added. Infinite executions can only be removed.

It follows that all properties shown in [12, Theorems 1–4] hold for conscriptions. This includes various properties of the operation n and of $\sqsubseteq$ and representations of $\sqsubseteq$-least fixpoints in terms of $\le$-least and $\le$-greatest fixpoints. For reference they are reproduced in the appendix of this paper.

3.3 Iteration

For instantiating further results concerning iteration, we show that conscriptions form a Kleene algebra and an omega algebra. The Kleene star is derived by the

standard automata-based matrix construction [3], according to which

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^* = \begin{pmatrix} e^* & a^*bf^* \\ d^*ce^* & f^* \end{pmatrix} \qquad \text{where} \qquad \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} a \cup bd^*c \\ d \cup ca^*b \end{pmatrix}$$

The Kleene star of a relation is its reflexive-transitive closure. For conscriptions, $e = 1 \cup 0R^*Q = 1$ and $f = R \cup Q1^*0 = R$, so the Kleene star elaborates to

$$\begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^* = \begin{pmatrix} 1^* & 1^*0R^* \\ R^*Q1^* & R^* \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ R^*Q & R^* \end{pmatrix}$$

The result is a conscription, whence the operation satisfies the Kleene algebra axioms.

The standard automata-based construction does not work for the omega operation as the resulting matrix is not a conscription. This problem, which arises also for prescriptions and extended designs, is solved by typed omega algebras as detailed in [8]. The resulting operation is

$$\begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^\omega = \begin{pmatrix} 1 & 0 \\ R^\omega \cup R^*Q & R^\omega \end{pmatrix}$$

It satisfies the omega algebra axioms. The operation $^\omega$ on relations describes the states from which infinite transition paths exist. We obtain the following result.

Theorem 3. *Conscriptions form an n-omega algebra with the operations*

$$\begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^* = \begin{pmatrix} 1 & 0 \\ R^*Q & R^* \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^\omega = \begin{pmatrix} 1 & 0 \\ R^\omega \cup R^*Q & R^\omega \end{pmatrix}$$

Proof. Axiom $(n12)$ follows since $\mathsf{L}x = \mathsf{L}$. As $n(\mathsf{L}) = 1$, axiom $(n11)$ follows by

$$\begin{aligned}
&\begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^\omega = \begin{pmatrix} 1 & 0 \\ R^\omega \cup R^*Q & R^\omega \end{pmatrix} \leq \begin{pmatrix} 1 & 0 \\ R^\omega \cup R^*Q & R^\omega \cup \overline{\overline{R^*Q}\top} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ R^*R^\omega \cup R^*Q \cup R^*\overline{\overline{R^*Q}\top} & R^*R^\omega \cup R^*\overline{\overline{R^*Q}\top} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ R^*Q \cup R^*(R^\omega \cup \overline{\overline{R^*Q}\top}) & R^*(R^\omega \cup \overline{\overline{R^*Q}\top}) \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ R^*Q & R^* \end{pmatrix} \begin{pmatrix} 1 & 0 \\ R^\omega \cup \overline{\overline{R^*Q}\top} & R^\omega \cup \overline{\overline{R^*Q}\top} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^* \begin{pmatrix} 1 & 0 \\ \overline{\overline{R^\omega \cup R^*Q}\top} & \overline{\overline{R^\omega \cup R^*Q}\top} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^* \begin{pmatrix} 1 & 0 \\ 0 & \overline{\overline{R^\omega \cup R^*Q}\top} \cap 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^* n\begin{pmatrix} 1 & 0 \\ R^\omega \cup R^*Q & R^\omega \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^* n\left(\begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^\omega\right) \begin{pmatrix} 1 & 0 \\ \top & \top \end{pmatrix}
\end{aligned}$$

The calculation uses that R^ω is a vector and that $R^*\overline{\overline{R^*Q}\top} = \overline{\overline{R^*Q}\top} \subseteq R^*Q$. □

Therefore all properties shown in [12, Theorems 5–6] hold for conscriptions. This includes further properties of the operation n and representations of iteration in terms of the Kleene star and omega operations; see again the appendix. In particular, conscriptions form an extended binary itering and an itering as follows.

Corollary 1. *Conscriptions form an extended binary itering and an itering with*

$$\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} \star \begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ R_1^\omega \cup R_1^*(Q_1 \cup Q_2) & R_1^* R_2 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 \\ Q & R \end{pmatrix}^\circ = \begin{pmatrix} 1 & 0 \\ R^\omega \cup R^* Q & R^* \end{pmatrix}$$

Proof. The extended binary itering instance follows by [12, Theorem 6]. Moreover, conscriptions satisfy the property $(x \star y)z = x \star (yz)$:

$$\begin{aligned}
&\left(\begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} \star \begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix}\right) \begin{pmatrix} 1 & 0 \\ Q_3 & R_3 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ R_1^\omega \cup R_1^*(Q_1 \cup Q_2) & R_1^* R_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ Q_3 & R_3 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ R_1^\omega \cup R_1^*(Q_1 \cup Q_2) \cup R_1^* R_2 Q_3 & R_1^* R_2 R_3 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ R_1^\omega \cup R_1^*(Q_1 \cup Q_2 \cup R_2 Q_3) & R_1^* R_2 R_3 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} \star \begin{pmatrix} 1 & 0 \\ Q_2 \cup R_2 Q_3 & R_2 R_3 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ Q_1 & R_1 \end{pmatrix} \star \left(\begin{pmatrix} 1 & 0 \\ Q_2 & R_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ Q_3 & R_3 \end{pmatrix}\right)
\end{aligned}$$

Hence conscriptions form an itering with $x^\circ = x \star 1$. □

It follows that all consequences of iterings and binary iterings shown in [9, 10] hold for conscriptions. They include separation theorems generalised from omega algebras and Back's atomicity refinement theorem.

4 Extended Conscriptions

Extended conscriptions combine aspects of three computation models:

- They represent aborting executions in addition to finite and infinite executions; so do extended designs [13].
- They represent aborting, infinite and finite executions independently; so does the model introduced in [9].
- Aborting executions can refer to final states; so do infinite executions in conscriptions.

Infinite executions of extended conscriptions are restricted to refer to initial states only.

An *extended conscription* is a 3×3 matrix whose entries are relations over the state space A. The matrix has the following form

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ P & Q & R \end{pmatrix}$$

where Q is a vector, that is, $Q\top = Q$. The relation P represents the aborting executions, Q represents the states from which infinite executions exist and R represents the finite executions. Hence the endless loop is represented by the extended conscription

$$\mathsf{L} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ 0 & \top & 0 \end{pmatrix}$$

Sequential composition and non-deterministic choice of extended conscriptions are given by matrix product and componentwise union, respectively. The operation n for extended conscriptions is derived by the method applied to conscriptions in Section 3.2. The result is

$$n\begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ P & Q & R \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ 0 & 0 & Q \cap 1 \end{pmatrix}$$

The simpler form $Q \cap 1$ is due to the fact that Q is a vector. This operation satisfies the axioms given in [9] for models of strict computations and the axioms given in [12]. The approximation order instantiates to

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ P_1 & Q_1 & R_1 \end{pmatrix} \sqsubseteq \begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ P_2 & Q_2 & R_2 \end{pmatrix} \Leftrightarrow \begin{array}{c} P_1 \subseteq P_2 \subseteq P_1 \cup Q_1 \wedge \\ Q_2 \subseteq Q_1 \wedge \\ R_1 \subseteq R_2 \subseteq R_1 \cup Q_1 \end{array}$$

The intuition is that in states with an infinite execution, any aborting and finite executions can be added. In states with no infinite execution, no executions can be added. Infinite executions can only be removed.

The standard matrix construction for the Kleene star and the typed matrix construction for the omega operation yield the following operations. Moreover extended conscriptions form an itering that does not satisfy $x^\circ = x^\omega 0 + x^*$ in general:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ P & Q & R \end{pmatrix}^* = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ R^*P & R^*Q & R^* \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ P & Q & R \end{pmatrix}^\omega = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ R^\omega \cup R^*P & R^\omega \cup R^*Q & R^\omega \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ P & Q & R \end{pmatrix}^\circ = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \top & 0 \\ R^*P & R^\omega \cup R^*Q & R^* \end{pmatrix}$$

The following result summarises the algebraic properties of extended conscriptions. Hence all properties shown in [9, 12] hold for extended conscriptions.

Theorem 4. *Extended conscriptions form a lattice-ordered semiring, an n-algebra, an n-omega algebra, an itering and an extended binary itering.*

Proof. The lattice-ordered semiring, n-algebra and n-omega algebra instances follow by calculations as in the proof of Theorems 1–3. The itering and extended binary itering instances follow as in the proof of Corollary 1. □

5 Further computation models

Comparing the various computation models – designs, prescriptions, extended designs, conscriptions, extended conscriptions – it is natural to further generalise extended conscriptions by eliminating the restriction placed on the infinite executions. This is done in a similar way as for conscriptions and for the aborting executions of extended conscriptions.

A computation in the resulting model is a 3×3 matrix of the following form:

$$(P|Q|R) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ P & Q & R \end{pmatrix}$$

There are no restrictions on P, Q or R. The calculations to obtain the operation n, the approximation order, the Kleene star, the omega operation, the itering operation and the binary itering operation follow the method of Section 3. The following result summarises the algebraic structure.

Theorem 5. *Let A be a set and let $S = \{(P|Q|R) \mid P, Q, R \subseteq A \times A\}$. Then S is a lattice-ordered semiring, an n-algebra, an n-omega algebra, an itering and an extended binary itering using the following operations:*

$$\begin{aligned}
(P_1|Q_1|R_1) + (P_2|Q_2|R_2) &= (P_1 \cup P_2|Q_1 \cup Q_2|R_1 \cup R_2) \\
(P_1|Q_1|R_1) \curlywedge (P_2|Q_2|R_2) &= (P_1 \cap P_2|Q_1 \cap Q_2|R_1 \cap R_2) \\
(P_1|Q_1|R_1) \cdot (P_2|Q_2|R_2) &= (P_1 \cup R_1P_2|Q_1 \cup R_1Q_2|R_1R_2) \\
(P_1|Q_1|R_1) \star (P_2|Q_2|R_2) &= (R_1^*(P_1 \cup P_2)|R_1^\omega \cup R_1^*(Q_1 \cup Q_2)|R_1^*R_2) \\
n(P|Q|R) &= (0|0|\overline{\overline{Q}\top} \cap 1) \\
(P|Q|R)^* &= (R^*P|R^*Q|R^*) \\
(P|Q|R)^\omega &= (R^\omega \cup R^*P|R^\omega \cup R^*Q|R^\omega) \\
(P|Q|R)^\circ &= (R^*P|R^\omega \cup R^*Q|R^*) \\
0 &= (0|0|0) \\
\top &= (\top|\top|\top) \\
1 &= (0|0|1) \\
\mathsf{L} &= (0|\top|0)
\end{aligned}$$

The approximation order on S is

$$\begin{aligned}
(P_1|Q_1|R_1) \sqsubseteq (P_2|Q_2|R_2) \;\Leftrightarrow\; & P_1 \subseteq P_2 \subseteq P_1 \cup \overline{\overline{Q_1}\top} \wedge Q_2 \subseteq Q_1 \wedge \\
& R_1 \subseteq R_2 \subseteq R_1 \cup \overline{\overline{Q_1}\top}
\end{aligned}$$

Proof. The lattice-ordered semiring, n-algebra and n-omega algebra instances follow by calculations as in the proof of Theorems 1–3. The itering and extended binary itering instances follow as in the proof of Corollary 1. □

This computation model is the most precise among those considered in this paper: it can represent finite, infinite and aborting executions independently and without any restrictions. Previously investigated computation models are isomorphic to substructures of this model:

- extended conscriptions: $\{(P|Q|R) \mid Q = Q\top\}$,
- the model of [9]: $\{(P|Q|R) \mid P = P\top \wedge Q = Q\top\}$,
- extended designs: $\{(P|Q|R) \mid P = P\top \wedge Q = Q\top \wedge P \subseteq Q \wedge P \subseteq R\}$,
- conscriptions: $\{(P|Q|R) \mid P = 0\}$,
- prescriptions: $\{(P|Q|R) \mid P = 0 \wedge Q = Q\top\}$,
- designs: $\{(P|Q|R) \mid P = 0 \wedge Q = Q\top \wedge Q \subseteq R\}$.

Other restrictions lead to further computation models which can be represented by matrices, for example,

- $\{(P|Q|R) \mid P = P\top\}$ requires that aborting executions do not refer to final states;
- $\{(P|Q|R) \mid P = P\top \wedge Q = Q\top \wedge Q \subseteq P \wedge Q \subseteq R\}$ requires that aborting and infinite executions do not refer to final states and that in the presence of infinite executions, aborting or finite executions cannot be distinguished.

Further combinations are possible, but in each case it has to be verified that the subset is closed under operations such as sequential composition.

6 Conclusion

In this paper we have derived approximation orders for new computation models based on a Galois connection for infinite executions and on algebras previously introduced for other models. Once more this shows that the algebraic approach can essentially contribute to the development of computation models. Additionally we inherit a multitude of results that have been proved for the previously introduced algebras. Future work will be concerned with computation models involving time, such as those studied in [13, 14, 5].

Acknowledgement. I thank the anonymous referees for helpful comments.

References

1. Blanchette, J.C., Böhme, S., Paulson, L.C.: Extending Sledgehammer with SMT solvers. In: Bjørner, N., Sofronie-Stokkermans, V. (eds.) CADE 2011. LNCS, vol. 6803, pp. 116–130. Springer (2011)
2. Cohen, E.: Separation and reduction. In: Backhouse, R., Oliveira, J.N. (eds.) MPC 2000. LNCS, vol. 1837, pp. 45–59. Springer (2000)

3. Conway, J.H.: Regular Algebra and Finite Machines. Chapman and Hall (1971)
4. Dunne, S.: Recasting Hoare and He's Unifying Theory of Programs in the context of general correctness. In: Butterfield, A., Strong, G., Pahl, C. (eds.) 5th Irish Workshop on Formal Methods. Electronic Workshops in Computing, The British Computer Society (2001)
5. Dunne, S.: Conscriptions: A new relational model for sequential computations. In: Wolff, B., Gaudel, M.C., Feliachi, A. (eds.) UTP 2012. LNCS, vol. 7681, pp. 144–163. Springer (2013)
6. Dunne, S.E., Hayes, I.J., Galloway, A.J.: Reasoning about loops in total and general correctness. In: Butterfield, A. (ed.) UTP 2008. LNCS, vol. 5713, pp. 62–81. Springer (2010)
7. Guttmann, W.: General correctness algebra. In: Berghammer, R., Jaoua, A.M., Möller, B. (eds.) RelMiCS/AKA 2009. LNCS, vol. 5827, pp. 150–165. Springer (2009)
8. Guttmann, W.: Towards a typed omega algebra. In: de Swart, H. (ed.) RAMiCS 2011. LNCS, vol. 6663, pp. 196–211. Springer (2011)
9. Guttmann, W.: Algebras for iteration and infinite computations. Acta Inf. 49(5), 343–359 (2012)
10. Guttmann, W.: Unifying lazy and strict computations. In: Kahl, W., Griffin, T.G. (eds.) RAMiCS 2012. LNCS, vol. 7560, pp. 17–32. Springer (2012)
11. Guttmann, W.: Extended designs algebraically. Sci. Comput. Program. 78(11), 2064–2085 (2013)
12. Guttmann, W.: Infinite executions of lazy and strict computations (2013), submitted
13. Hayes, I.J., Dunne, S.E., Meinicke, L.: Unifying theories of programming that distinguish nontermination and abort. In: Bolduc, C., Desharnais, J., Ktari, B. (eds.) MPC 2010. LNCS, vol. 6120, pp. 178–194. Springer (2010)
14. Hayes, I.J., Dunne, S.E., Meinicke, L.A.: Linking Unifying Theories of Program refinement. Sci. Comput. Program. 78(11), 2086–2107 (2013)
15. Hoare, C.A.R., He, J.: Unifying theories of programming. Prentice Hall Europe (1998)
16. Kozen, D.: A completeness theorem for Kleene algebras and the algebra of regular events. Information and Computation 110(2), 366–390 (1994)
17. Möller, B.: Kleene getting lazy. Sci. Comput. Program. 65(2), 195–214 (2007)
18. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic, LNCS, vol. 2283. Springer (2002)
19. Paulson, L.C., Blanchette, J.C.: Three years of experience with Sledgehammer, a practical link between automatic and interactive theorem provers. In: Sutcliffe, G., Ternovska, E., Schulz, S. (eds.) Proceedings of the 8th International Workshop on the Implementation of Logics. pp. 3–13 (2010)
20. Schmidt, G., Ströhlein, T.: Relationen und Graphen. Springer (1989)

Appendix: Consequences of n-algebras

Because the models discussed in this paper form n-omega algebras, they satisfy all of the following results, which appear as Theorems 1–6 in [12]. The results have been verified in Isabelle/HOL [18], making heavy use of its integrated automated theorem provers and SMT solvers [19, 1]. The proofs can be found in the theory files at http://www.csse.canterbury.ac.nz/walter.guttmann/algebra/.

Proposition 1. *Let S be an n-algebra. Then $(n(S), +, \cdot, n(0), n(\top))$ is a semiring with right annihilator $n(0)$ and a bounded distributive lattice with meet $\cdot$. Moreover, n is $\leq$-isotone and the following properties hold for $x, y \in S$:*

1. $n(x)n(y) = n(y)n(x)$
2. $n(x)n(x) = n(x)$
3. $n(x)n(y) \leq n(x)$
4. $n(x)n(y) \leq n(y)$
5. $n(x) \leq n(x + y)$
6. $n(x) \leq 1$
7. $n(x)0 = 0$
8. $n(x)n(0) = n(0)$
9. $n(x) \leq x + n(x0)$
10. $n(x + n(x)\top) = n(x)$
11. $n(n(x)\mathsf{L}) = n(x)$
12. $n(x)n(\mathsf{L}) = n(x)$
13. $n(x) \leq n(\mathsf{L})$
14. $n(x) \leq n(x\mathsf{L})$
15. $n(x)\mathsf{L} \leq x\mathsf{L}$
16. $n(0)\mathsf{L} = 0$
17. $n(\mathsf{L}) = n(\top)$
18. $n(x\top) = n(x\mathsf{L})$
19. $n(x)\top = n(x)\mathsf{L} + n(x0)\top$
20. $n(xn(y)\mathsf{L}) \leq n(xy)$
21. $xn(y)\top \leq xy + n(xy)\top$
22. $n(x)\top y \leq xy + n(xy)\top$
23. $xn(y)\mathsf{L} = x0 + n(xn(y)\mathsf{L})\mathsf{L}$
24. $xn(y)\mathsf{L} \leq x0 + n(xy)\mathsf{L}$
25. $n(\mathsf{L})x \leq x0 + n(x\mathsf{L})\top$
26. $n(\mathsf{L})\mathsf{L} = \mathsf{L}n(\mathsf{L}) = \mathsf{L}$
27. $\mathsf{L}\mathsf{L} = \mathsf{L}\top = \mathsf{L}\top\mathsf{L} = \mathsf{L}$
28. $\mathsf{L}x \leq \mathsf{L}$
29. $x\mathsf{L} \leq x0 + \mathsf{L}$
30. $x\top \curlywedge \mathsf{L} \leq x\mathsf{L}$
31. $x\top y \curlywedge \mathsf{L} = x\mathsf{L}y \curlywedge \mathsf{L}$
32. $x\top y \curlywedge \mathsf{L} \leq x0 + \mathsf{L}y$
33. $(x \curlywedge \mathsf{L})0 \leq x0 \curlywedge \mathsf{L}$
34. $n(x) = n(x \curlywedge \mathsf{L}) = n(x) \curlywedge \mathsf{L} + n(x0)$
35. $n(x)\mathsf{L} \leq x \curlywedge \mathsf{L} \leq n(\mathsf{L})x$
36. $n(x) \curlywedge \mathsf{L} \leq (n(x) \curlywedge \mathsf{L})\top \leq n(x)\mathsf{L} \leq x$
37. $x \leq y \Leftrightarrow x \leq y + \mathsf{L} \wedge n(\mathsf{L})x \leq y + n(y)\top$
38. $x \leq y \Leftrightarrow x \leq y + \mathsf{L} \wedge x \leq y + n(y)\top$
39. $n(y)x \leq xn(y) \Leftrightarrow n(y)x = n(y)xn(y)$
40. $n(x) \leq n(y) \Leftrightarrow n(x)\mathsf{L} \leq y$

In an n-algebra S the approximation relation $\sqsubseteq$ is defined by

$$x \sqsubseteq y \;\Leftrightarrow\; x \leq y + \mathsf{L} \wedge n(\mathsf{L})y \leq x + n(x)\top$$

where $x, y \in S$. The following result gives properties of $\sqsubseteq$.

Proposition 2. *Let S be an n-algebra.*

1. *The relation $\sqsubseteq$ is a partial order with least element L.*
2. *The operations $+$ and $\cdot$ and $\lambda x.x \curlywedge \mathsf{L}$ and $\lambda x.n(x)\mathsf{L}$ are $\sqsubseteq$-isotone.*
3. *If S is an itering, the operation $^\circ$ is $\sqsubseteq$-isotone.*
4. *If S is a Kleene algebra, the operation * is $\sqsubseteq$-isotone.*

Further results concern fixpoints of a function $f : S \to S$. Provided they exist, the $\leq$-least, $\leq$-greatest and $\sqsubseteq$-least fixpoints of f are denoted by μf, νf and κf, respectively:

$$\begin{array}{ll} f(\mu f) = \mu f & f(x) = x \Rightarrow \mu f \leq x \\ f(\nu f) = \nu f & f(x) = x \Rightarrow \nu f \geq x \\ f(\kappa f) = \kappa f & f(x) = x \Rightarrow \kappa f \sqsubseteq x \end{array}$$

We abbreviate $\kappa(\lambda x.f(x))$ by $\kappa x.f(x)$. Provided it exists, the $\sqsubseteq$-greatest lower bound of $x, y \in S$ is denoted by $x \sqcap y$:

$$x \sqcap y \sqsubseteq x \qquad x \sqcap y \sqsubseteq y \qquad z \sqsubseteq x \wedge z \sqsubseteq y \Rightarrow z \sqsubseteq x \sqcap y$$

Proposition 3. *Let S be an n-algebra, let $f : S \to S$ be $\leq$- and $\sqsubseteq$-isotone, and assume that μf and νf exist. Then the following are equivalent:*

1. *κf exists.*
2. *κf and $\mu f \sqcap \nu f$ exist and $\kappa f = \mu f \sqcap \nu f$.*
3. *κf exists and $\kappa f = (\nu f \curlywedge \mathsf{L}) + \mu f$.*
4. *$n(\mathsf{L})\nu f \leq (\nu f \curlywedge \mathsf{L}) + \mu f + n(\nu f)\top$.*
5. *$n(\mathsf{L})\nu f \leq (\nu f \curlywedge \mathsf{L}) + \mu f + n((\nu f \curlywedge \mathsf{L}) + \mu f)\top$.*
6. *$(\nu f \curlywedge \mathsf{L}) + \mu f \sqsubseteq \nu f$.*
7. *$\mu f \sqcap \nu f$ exists and $\mu f \sqcap \nu f = (\nu f \curlywedge \mathsf{L}) + \mu f$.*
8. *$\mu f \sqcap \nu f$ exists and $\mu f \sqcap \nu f \leq \nu f$.*

Condition 4 of this proposition characterises the existence of κf in terms of μf and νf. Condition 3 shows how to obtain κf from μf and νf. This simplifies calculations as $\leq$ is less complex than $\sqsubseteq$. Further characterisations generalise to n-algebras as shown in the following result.

Proposition 4. *Let S be an n-algebra, let $f : S \to S$ be $\leq$- and $\sqsubseteq$-isotone, and assume that μf and νf exist. Then the following are equivalent and imply the statements of Proposition 3:*

1. *κf exists and $\kappa f = n(\nu f)\mathsf{L} + \mu f$.*
2. *$n(\mathsf{L})\nu f \leq \mu f + n(\nu f)\top$.*
3. *$n(\nu f)\mathsf{L} + \mu f \sqsubseteq \nu f$.*
4. *$\mu f \sqcap \nu f$ exists and $\mu f \sqcap \nu f = n(\nu f)\mathsf{L} + \mu f$.*

Proposition 5. *Let S be an n-omega algebra and $x, y, z \in S$. Then the following properties hold:*

1. $\mathsf{L}x^* = \mathsf{L}$
2. $(x\mathsf{L})^* = 1 + x\mathsf{L}$
3. $(x\mathsf{L})^\omega = x\mathsf{L} = x\mathsf{L}x\mathsf{L}$
4. $(x\mathsf{L})^*y \leq y + x\mathsf{L}$
5. $(x\mathsf{L} + y)^* = y^* + y^*x\mathsf{L}$
6. $(x\mathsf{L} + y)^\omega = y^\omega + y^*x\mathsf{L}$
7. $n(x) \leq n(x^\omega)$
8. $n(y^\omega + y^*z) = n(y^\omega) + n(y^*z)$
9. $x^* + n(x^\omega)\mathsf{L} = x^* + x^*n(x^\omega)\mathsf{L}$
10. $x^* + n(x^\omega)\mathsf{L} = x^* + xn(x^\omega)\mathsf{L}$
11. $yx^* + n(yx^\omega)\mathsf{L} = yx^* + yn(x^\omega)\mathsf{L}$
12. $x^*0 + n(x^\omega)\mathsf{L} = x^*0 + x^*n(x^\omega)\mathsf{L}$
13. $xx^*0 + n(x^\omega)\mathsf{L} = xx^*0 + xn(x^\omega)\mathsf{L}$
14. $yx^*0 + n(yx^\omega)\mathsf{L} = yx^*0 + yn(x^\omega)\mathsf{L}$
15. $n(\mathsf{L})x^\omega \leq x^*0 + n(x^\omega)\top$
16. $n(\mathsf{L})(y^\omega + y^*z) \leq y^*z + n(y^\omega + y^*z)\top$

Proposition 6. *Let S be an n-omega algebra, let $x, y, z \in S$, and let $f : S \to S$ be given by $f(x) = yx + z$.*

1. *The $\sqsubseteq$-least fixpoint of f is $\kappa f = (y^\omega \curlywedge \mathsf{L}) + y^*z = n(y^\omega)\mathsf{L} + y^*z$.*
2. *The operations $^\omega$ and $\lambda y.(\kappa x.yx + z)$ and $\lambda z.(\kappa x.yx + z)$ are $\sqsubseteq$-isotone.*
3. *S is an extended binary itering using $x \star y = n(x^\omega)\mathsf{L} + x^*y$.*