Theoretical Computer Science 412 (2011) 3303–3322

Contents lists available at ScienceDirect

Theoretical Computer Science

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

Fixing Zeno gaps

Peter Höfner [a], Bernhard Möller [b,*]

[a] National ICT Australia (NICTA), Australia

[b] Institut für Informatik, Universität Augsburg, D-86159 Augsburg, Germany

A R T I C L E   I N F O

It is our pleasure to dedicate this paper to Jan Bergstra on the occasion of his 60th birthday. Jan's productivity and diligence have always been inspiring to us; moreover, we gratefully acknowledge his always speedy and effective help in many editorial issues. It is also admirable what wide range his scientific interests span, making him a Jan-of-all-trades in the very best sense.

Keywords:
Fixpoints
Iteration
Semiring
Kleene algebra
Omega algebra
Hybrid systems

A B S T R A C T

In computer science, fixpoints play a crucial role. Most often least and greatest fixpoints are sufficient. However, there are situations where other ones are needed. In this paper, we study, on an algebraic base, a special fixpoint of the function $f(x) = a \cdot x$ that describes infinite iteration of an element a. We show that the greatest fixpoint is too imprecise. Special problems arise if the iterated element contains the possibility of stepping on the spot (e.g. `skip` in a programming language) or if it allows Zeno behaviour. We present a construction for a fixpoint that captures these phenomena in a precise way. The theory is presented and motivated using an example from hybrid system analysis.

1. Introduction

Fixpoints occur nearly everywhere in computer science. They obviously play a crucial role in recursion and hence in algorithms. Other examples are in finding final states, for example, when determining a result set from a deductive database [33]. But, they also occur in compiler construction and data flow analysis (see [40] for an overview), in the lambda calculus (e.g., [17]), in concurrency [11] or in the verification of bytecode [46]. In denotational semantics, fixpoints are used to define the meaning of recursive definitions [21]. Fixpoints are also used in set theory, e.g., the fixed-point lemma for normal functions [36].

The existence of fixpoints can be established in various ways. A fundamental result is the well-known theorem of Knaster and Tarski [32,47], which guarantees for every isotone endofunction on a complete lattice, a complete sublattice of fixpoints. The assumption of a complete lattice can be weakened, e.g. to directed-complete partial orders [39].

If a class of functions happens to have unique fixpoints, things are easy (e.g. [13,48]). However, generally there are a variety of fixpoints of a given function. For a given task, one has has to choose a distinguished one of these.

Most often, the least fixpoint of a function is sufficient. In the case of a continuous function, it can be determined by countable iteration according the theorem of Kleene [31]. In the case of a non-continuous function, transfinite iteration may be necessary to reach the least fixpoint.

* Corresponding author. Tel.: +49 821 598 2164.
E-mail addresses: peter.hoefner@nicta.com.au (P. Höfner), moeller@informatik.uni-augsburg.de (B. Möller).

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.03.018

One particular concept that can be defined as a least fixpoint is finite iteration; it is useful for modelling programming constructs like the while-loop, but has also been introduced into process algebra [9]. The algebraic counterpart is Kleene's star operator [30] which has been thoroughly investigated in [20] and axiomatised in [34].

However, least fixpoints do not always suffice (even if they exist). Therefore other fixpoints were used. Park, for example, used a greatest fixpoint to model infinite iteration and fairness [44]. In such cases the least fixpoints are usually uninteresting or trivial. Greatest fixpoints are also used for parallel or nondeterministic programs [5,10,11] and for the algebraic description of a simple programming language [12]. The algebraic axiomatisation of a particular greatest fixpoint is provided by Cohen's omega algebra [19].

Sometimes, neither the greatest nor the least fixpoint is adequate. Manna and Shamir, for example, introduced the concept of the optimal fixpoint [37,38], the greatest lower bound of all maximal fixpoints. They used it to model the semantics of recursive programs. In particular circumstances, even combinations of various fixpoints have to be used (e.g. [18]).

In the present paper, we show another situation of this kind. More precisely, we study, on an algebraic base, a special fixpoint of the function $f(x) =_{df} a \cdot x$ that describes infinite iteration of an element a. Here, a abstractly stands for a set of transitions which may be discrete or continuous, and $\cdot$ denotes sequential composition. Frequently, the least fixpoint of f is the element 0 that represents the empty behaviour and hence is uninteresting. Therefore the greatest fixpoint a^ω of f was studied. However, in many cases a^ω is too large and imprecise, admitting behaviour that is not wanted. For instance, if the iterated element a contains the possibility of stepping on the spot (e.g. `skip` in a programming language), then a^ω coincides with the greatest element $\top$ of the underlying lattice, which represents the completely unrestricted behaviour that sometimes is called `chaos`.

Stepping on the spot is a special case of *Zeno* behaviour in which an infinite number of sufficiently short transitions add up to a finite overall duration. This, at least conceptually, may for instance occur in hybrid systems, that is, in heterogeneous systems characterised by the interaction of discrete and continuous dynamics. The investigations of the present paper originate in an algebraic setting for describing hybrid systems [26,25]. As special cases of that model, streams [16] and omega-regular expressions occur. It turns out that in the case of Zeno behaviour, the description using the greatest fixpoint a^ω for describing infinite iteration is again way too imprecise, since it allows arbitrary behaviour "after" a Zeno effect, whereas actually nothing should be observable any more after such an occurrence. Therefore a fixpoint between the two extremes – the least and the greatest fixpoint – is interesting and useful.

The present paper is about such a more precise fixpoint. To the best of our knowledge, such a fixpoint has not been studied by other authors, not even in the impressive and comprehensive book by Bloom and Ésik [15]. We have introduced this fixpoint already in [26], however, by a very concrete definition that does not lend itself easily to proofs of further properties of the operator and is not in a form that can be tackled in a general, more abstract algebraic setting. An improved presentation was given in the dissertation [25]. This is the basis for the treatment here, which eventually leads to an abstract algebraic axiomatisation.

The paper is organised as follows: In Section 2, we introduce an algebraic model that can be used to argue about hybrid systems. In particular, we introduce an example that is used throughout the paper. After that, we show in Section 3 that Zeno effects can occur in hybrid systems (at least theoretically) and therefore the model has to be able to reflect these phenomena. In Sections 4–6, we abstract from the concrete model and present the general algebraic background. In detail, we define algebraic structures for choice and concatenation in Section 4, describe how finite and infinite elements can be described in the algebra in Section 5 and discuss finite and infinite iteration in Section 6. After we have set up the problem and the foundations, we start in Section 7 to fix the Zeno gap and to give a construction for a new fixpoint. More precisely, we embed elements of the algebra of hybrid systems in an algebra of guarded strings. After we have analysed in detail why the greatest fixpoint is too imprecise for hybrid system analysis in Section 8, we define an adequate fixpoint in Section 9. This concrete definition is then lifted to a purely algebraic one (Section 10) and completed by a small calculus in Section 11. Before concluding the paper with a short outlook, we discuss Zeno behaviour in general algebraic terms in Section 12.

2. An algebraic model of hybrid systems

In this section we introduce the algebraic model we use to argue about hybrid systems. Another algebraic framework dealing with hybrid systems is the process algebra presented in [14]. It is obtained by extending a combination of two extensions of the algebra of communicating processes (ACP) [8], namely the process algebra with continuous relative timing from [7] and the process algebra with propositional signals from [6]. It has, in addition to equational axioms, some rules to derive further equations with the help of real analysis. However, it does not contain transformation rules for larger systems like the ones we have derived in earlier papers; moreover, it does not define operators for the analysis of the finite and infinite parts of behaviours nor does it use fixpoints.

Before we get into the details of our model, we illustrate semi-formally by an example from hybrid systems analysis that, as mentioned in the introduction, a greatest fixpoint definition of infinite iteration is too imprecise to describe Zeno effects properly and should be replaced by another one. Specifically, we look at a bouncing ball with energy loss in an ideal situation (no drag, etc.). It is one of the standard examples for hybrid system analysis in the literature.

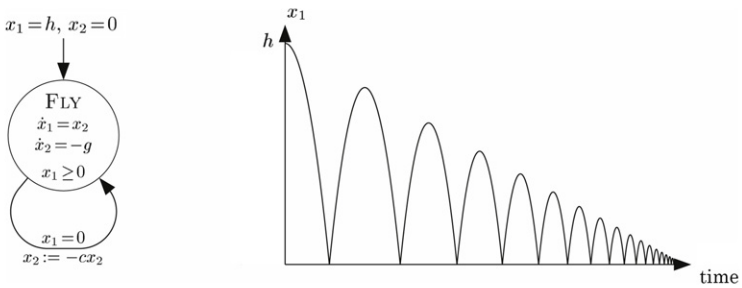


Fig. 1. Hybrid automaton of a bouncing ball and corresponding trajectory w.r.t. x_1.

Example 2.1. A ball, which is assumed to be a point-mass, falls from an initial altitude and bounces back from the ground, losing part of its energy when touching the ground. Between each bounce, the behaviour of the ball is described by a differential equation; hence this is the continuous part of the hybrid system. The discrete part occurs when the ball touches the ground and its velocity changes immediately (modelled by an inelastic collision). The corresponding hybrid automaton and one of its trajectories (restricted to x_1) are given in Fig. 1. We omit the formal definition of the involved hybrid automaton (and of hybrid automata in general); for the purpose of this paper the intuition of what a hybrid automaton is should suffice. For more details, see [2,24].

The variable x_1 represents the altitude of the ball, x_2 its velocity. The initial altitude is h and the initial velocity of the ball is 0. If the ball is above the ground ($x_1 > 0$), its movement is governed by $\dot{x}_1 = x_2, \dot{x}_2 = -g$, where g is an arbitrary positive constant gravity force and $\dot{x}_i$ is the timewise derivative of x_i. These equations state that when the ball is above the ground, it is being drawn to the ground by gravity. Moreover, we assume a damping factor $0 \leq c < 1$ which makes the ball lose energy with every bounce. Zeno behaviour has a strict mathematical definition, but can be described informally as the system making an *infinite number* of jumps in a finite amount of time. In this example, the loss of energy makes the subsequent jumps closer and closer together in time (see the right part of Fig. 1). □

As we will show, infinite iteration and Zeno effects can be characterised by fixpoints at an algebraic level. The algebraic description we are looking at is based on sets of trajectories that reflect the variation of the values of the variables in a system over time.

To make the paper self-contained we repeat all basic notions, like trajectories and runs, as well as the main concepts of our model. The original description can be found in [25,26].

A trajectory maps time points to values; it also has a duration, i.e., an upper bound for the admissible time points. Let therefore V be a set of *values* and D a set of *durations* (e.g. $\mathbb{N}$, $\mathbb{Q}_{\geq 0}$, $\mathbb{R}_{\geq 0}$, ...). We assume a cancellative addition $+$ on D and an element $0 \in D$ such that $(D, +, 0)$ is a commutative monoid. Cancellativity means that for arbitrary elements d_1, d_2 and d_3 the implication $d_1 + d_3 = d_2 + d_3 \Rightarrow d_1 = d_2$ holds. Furthermore, we assume that the relation $d_1 \leq d_2 \Leftrightarrow_{df} \exists d \,.\, d_1 + d = d_2$ is a linear order on D. Then 0 is the least element and $+$ is isotone w.r.t. $\leq$. Moreover, 0 is indivisible, i.e., $d_1 + d_2 = 0 \Leftrightarrow d_1 = d_2 = 0$.

Often we also want to talk about behaviour that goes on infinitely. In other words, we not only want trajectories with finite duration but also ones with infinite duration. Therefore, we include into D the special value ∞. In this case, ∞ is required to be an annihilator w.r.t. $+$ and hence is the greatest element of D (and cancellativity of $+$ is restricted to elements in $D - \{\infty\}$).

For $d \in D$ we define the interval $\mathsf{intv}\, d$ of admissible times as

$$\mathsf{intv}\, d =_{df} \begin{cases} [0, d] & \text{if } d \neq \infty \\ [0, d[& \text{otherwise.} \end{cases}$$

Mathematically, a *trajectory* τ is a pair (d, g), where $d \in D$ and $g : \mathsf{intv}\, d \to V$. Then d is the *duration* of the trajectory and the image of $\mathsf{intv}\, d$ under g is its *range* $\mathsf{ran}(d, g)$.

The set of all trajectories is denoted by TRA. For a discrete infinite set of durations D, e.g. $D = \mathbb{N}$, trajectories are isomorphic to non-empty finite or infinite words over the value set V, as known from (omega-)regular languages. Moreover, if V consists of values of computations, then the elements of TRA can be viewed as sets of computation streams (e.g. [16]).

A special role is played by *zero-length trajectories* of the form $\underline{v} =_{df} (0, g)$ with $v \in V$ and $g(0) =_{df} v$; they represent single values of the system.

We define composition of trajectories (d_1, g_1) and (d_2, g_2) as

$$(d_1, g_1) \cdot (d_2, g_2) =_{df} \begin{cases} (d_1 + d_2, g) & \text{if } d_1 \neq \infty \wedge g_1(d_1) = g_2(0) \\ (d_1, g_1) & \text{if } d_1 = \infty \\ \text{undefined} & \text{otherwise} \end{cases}$$

with $g(t) = g_1(t)$ for all $t \in [0, d_1]$ and $g(t + d_1) = g_2(t)$ for all $t \in \mathsf{intv}\, d_2$. This is well defined by cancellativity of $+$ on durations other than ∞. Moreover, composition is associative. For a zero-length trajectory $\underline{v}$ we have $\underline{v} \cdot (d, g) = (d, g)$ if $v = g(0)$; otherwise the composition is undefined. Likewise, $(d, g) \cdot \underline{v} = (d, g)$ if $v = g(d)$ or $d = \infty$.

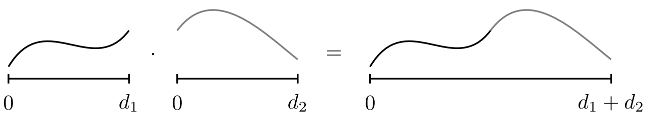


Fig. 2. Composition of two finite trajectories.

Fig. 2 illustrates the main idea for composing trajectories. Sometimes the condition $g_1(d_1) = g_2(0)$ for composing trajectories is too restrictive. In [26], a possibility to relax the condition is given. It allows jumps at the composition point for the function describing the timewise behaviour. However, for the present paper the above definition of composition, where trajectories must match at the gluing point, is sufficient.

A *process* is a set of trajectories, consisting of possible behaviours of a hybrid system. Note that we do not put any restrictions (such as prefix-closure) on a process. The set of all processes is denoted by PRO; the greatest process is TRA.

Single trajectories only describe one specific behaviour, whereas processes are able to describe all possible behaviours of a system. This is the reason why our algebra of hybrid systems is built in terms of processes rather than single trajectories.

Composition can be lifted from trajectories to processes. For that we define the sets $\mathsf{inf}\,A$ and $\mathsf{fin}\,A$ of *purely infinite* and *purely finite parts* of a process A, resp.

$$\mathsf{inf}\,A =_{df} \{(d, g) \mid (d, g) \in A,\ d = \infty\}, \qquad \mathsf{fin}\,A =_{df} \{(d, g) \mid (d, g) \in A,\ d \neq \infty\} = A - \mathsf{inf}\,A.$$

Using these sets, composition of processes A and B is now defined by

$$A \cdot B =_{df} \mathsf{inf}\,A \cup \{\tau_1 \cdot \tau_2 \mid \tau_1 \in \mathsf{fin}\,A,\ \tau_2 \in B,\ \tau_1 \cdot \tau_2 \text{ defined}\}. \tag{1}$$

The set $I =_{df} \{\underline{v} \mid v \in V\}$ of all zero-length trajectories is the neutral element for this operation. Since it does not change anything it is closely related to the command `skip` in programming languages. Sets of zero-length trajectories, i.e., subprocesses of I, correspond to sets of values and can be used to restrict processes. Let $R \subseteq I$ be such a set and A be an arbitrary process. Then $R \cdot A$ consists of those trajectories of A whose initial values lie in R, while $A \cdot R$ is the set of trajectories of A whose final values, if any, are in R. Moreover, $\cdot$ distributes through arbitrary unions in its left argument and through non-empty ones in its right argument. Formally, this means

$$\left(\bigcup_{i\in I} B_i\right) \cdot A = \bigcup_{i\in I}(B_i \cdot A), \qquad A \cdot \left(\bigcup_{j\in J} C_j\right) = \bigcup_{j\in J}(A \cdot C_j),$$

for arbitrary I and $J \neq \emptyset$. In particular, $\emptyset \cdot A = \emptyset$ and $\cdot$ is $\subseteq$-isotone in both arguments.

Example 2.2. To model the bouncing ball algebraically, we set $D = \mathbb{R}_{\geq 0}$ and $V = \mathbb{R}^2$, and define a process

$$Z =_{df} \{(d, (x_1, x_2)) \mid \dot{x}_1 = x_2,\ \dot{x}_2 = -g\}.$$

The pair (x_1, x_2) is short for a function that takes times to pairs each consisting of an altitude and a velocity. Let now $\circ$ stand for some iteration operator. Then the bouncing behaviour should be characterised by

$$Y \cdot (\hat{Z})^{\circ},$$

where $Y =_{df} \{(0, g) \mid g(0) = h\}$ models the initialisation and the process $\hat{Z}$ extends all trajectories in Z at both ends suitably to enforce the changes in direction and velocity when the ball touches the ground. For details concerning the extension, we refer to [25]. In this paper we will focus on the iteration operators and skip other details. □

Since PRO is a power set lattice and $\cdot$ is isotone we can use the Knaster–Tarski theorem to describe a particular form of infinite iteration as a greatest fixpoint. We define *omega iteration* A^{ω} of a process A by

$$A^{\omega} =_{df} \nu X \,.\, A \cdot X,$$

where, for a function F, the expression $\nu X \,.\, F(X)$ denotes greatest fixpoint of F.

One might expect that in the example of the bouncing ball, $Y \cdot (\hat{Z})^{\omega}$ exactly characterises the trajectory of Fig. 1. But this is not the case and, as we will show in the next section, this definition does not adequately cope with Zeno effects.

3. Zeno effects

Zeno of Elea's famous paradox about Achilles and the tortoise is well known. Obviously such effects may occur in PRO. This is also true for most of the other approaches to hybrid systems. However, only few authors explicitly deal with Zeno effects (e.g. [28,3]). To illustrate the differences in treatment, let us have a closer look at the bouncing ball example.

Example 3.1. In our example we assume an ideal setting. In particular, the ball only loses energy if and only if it touches the ground. In the model, the energy loss depends on the ball's velocity (energy) x_2. It is reduced by multiplying x_2 with the damping factor c. If $c \neq 0$ the velocity x_2 never vanishes completely, since the energy is never 0 when the ball touches the ground. Hence the ball indeed bounces infinitely often. □

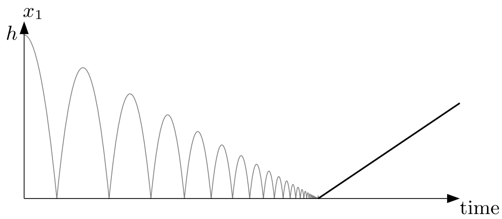


Fig. 3. Another trajectory of $Y \cdot (\hat{Z})^\omega$.

Contrarily, for example, in [45] the authors avoid Zeno effects for the bouncing ball by changing the setting and making the damping factor a variable which can change before each jump. If at the moment when the ball touches the ground the velocity goes below a limit the damping factor becomes 0 and the ball remains on the floor. Obviously, both approaches are purely theoretic, since both do not reflect reality (neither the damping change nor that the ball performs infinitely many jumps).

Now we show why omega iteration is too imprecise to cope with Zeno effects properly.

Lemma 3.2. *For arbitrary process A we have* $A^\omega = A^\omega \cdot \mathsf{TRA}$.

Proof. Let again be $F(X) =_{df} A \cdot X$. First, since A^ω is a fixpoint of F and by associativity of $\cdot$, we have

$$A^\omega \cdot \mathsf{TRA} = (A \cdot A^\omega) \cdot \mathsf{TRA} = A \cdot (A^\omega \cdot \mathsf{TRA}),$$

so that $A^\omega \cdot \mathsf{TRA}$ is also a fixpoint of F. Since A^ω is the greatest fixpoint of F, we obtain $A^\omega \cdot \mathsf{TRA} \subseteq A^\omega$. On the other hand, since $I \subseteq \mathsf{TRA}$ we have by neutrality of I and isotony of $\cdot$ that

$$A^\omega = A^\omega \cdot I \subseteq A^\omega \cdot \mathsf{TRA}. \quad \square$$

This property is well known from the algebraic theory of omega iteration (e.g. [19]). Informally it means that a process resulting from omega (infinite) iteration has to be closed under adding arbitrary suffixes to its trajectories.

Example 3.3. Applying this to our example of the bouncing ball (cf. Example 2.1), we see that $Y \cdot (\hat{Z})^\omega$ contains the trajectory of Fig. 1. In fact, the process contains an infinite number of trajectories. Their initial sections coincide with the trajectory of Fig. 1, but after reaching the Zeno point some miraculous behaviour may occur. This means that the ball might lie on the ground forever or somebody can lift the ball to a new initial altitude or something else may happen. In general, the process contains arbitrary extensions of the trajectory of Fig. 1. One is given in Fig. 3. $\square$

Still, the property $A^\omega = A^\omega \cdot \mathsf{TRA}$ is not completely unnatural: for arbitrary $B \in \mathsf{PRO}$ the process $B \cdot \mathsf{TRA}$ is the extension closure of B. Hence $A^\omega = A^\omega \cdot \mathsf{TRA}$ reflects the view that, operationally, after a Zeno gap the behaviour does not matter, since the gap cannot be "crossed" anyway. For a discussion of these phenomena in the context of hybrid automata see [43].

However, our goal in the rest of the paper is to define a revised operator for infinite iteration that discards trajectories that would contain behaviour after a Zeno gap. At the same time we want to pass to a more abstract algebraic treatment, since proofs at the concrete level of processes tend to be cumbersome and involved. Therefore, we now introduce the algebraic theory we are going to use.

4. Weak idempotent semirings and quantales

Let us have a closer look at the algebraic structure of the operations on hybrid systems presented in Section 2.

Definition 4.1. 1. A *weak semiring* is a quintuple $(S, +, 0, \cdot, 1)$ such that $(S, +, 0)$ is a commutative monoid and $(S, \cdot, 1)$ is a monoid such that $\cdot$ distributes over $+$ in both arguments and is *left-strict*, i.e., $0 \cdot a = 0$. The weak semiring is *idempotent* if $+$ is idempotent, i.e., $a + a = a$. In this case, the *natural order* $\leq$ on S is given by $a \leq b \Leftrightarrow_{df} a + b = b$.
   The natural order induces an upper semilattice in which $a + b$ is the supremum of a and b and 0 is the least element. Distributivity immediately implies $\leq$-isotony of $\cdot$ in both arguments.
2. A *semiring* is a weak semiring in which composition is also right-strict; when we want to emphasise this, we also speak of a *full* semiring.
3. A weak idempotent semiring is *Boolean* if its semilattice is even a Boolean algebra with complement $\overline{a}$ and infimum $a \sqcap b =_{df} \overline{\overline{a} + \overline{b}}$. In this case we have the *shunting rule*

$$a \sqcap b \leq c \Leftrightarrow a \leq \overline{b} + c. \tag{2}$$

4. An idempotent weak semiring S is called a *weak quantale* if S is a complete lattice under the natural order and $\cdot$ is universally disjunctive in its left argument. Following Conway [20], one might also call a weak quantale a *weak standard Kleene algebra*.

Checking all the axioms for the case of processes we get

Theorem 4.2. 1. *The processes under union as addition and composition as multiplication form a Boolean weak quantale* $\mathsf{PRO} =_{df} (\mathcal{P}(\mathsf{TRA}), \cup, \emptyset, \cdot, I)$. *Here,* $\emptyset$ *is the process without any trajectory; hence it can perform nothing. The other extreme* TRA *contains all possible trajectories, i.e., it can perform anything.*
2. *Additionally,* $\cdot$ *is positively disjunctive in its right argument.*

A further important Boolean semiring (that is even a full quantale) is REL, the algebra of binary relations over a set under union and relational composition.

Another example are *guarded strings* (e.g., [23,35]), which will be central in our main construction concerning the analysis of infinite iteration of processes.

Notation. As usual, a *finite word* over a set $\mathcal{A}$ is a finite sequence of zero or more elements from $\mathcal{A}$; an *infinite word* is an infinite sequence. The *concatenation* of words w_1 and w_2 is denoted by $w_1.w_2$ if w_1 is finite and w_1 otherwise. The set of all finite words over $\mathcal{A}$ is denoted by $\mathcal{A}^*$, the set of all infinite words by $\mathcal{A}^\omega$, the first element of a non-empty word w by $\mathsf{first}(w)$ and the last element of a non-empty finite word w by $\mathsf{last}(w)$.

Definition 4.3. A *guarded string* over arbitrary sets P of *states* and Σ of *transitions* is a non-empty word ρ over $P \cup \Sigma$ such that $\mathsf{first}(\rho) \in P$ and in which elements from P and Σ alternate. Moreover, if the word is finite, $\mathsf{last}(\rho) \in P$. The product of guarded strings ρ_0 and ρ_1 is the guarded string

$$\rho_0 \bowtie \rho_1 =_{df} \begin{cases} w_0.p.w_1 & \text{if } \rho_0 \text{ is finite, } \rho_0 = w_0.p \text{ and } \rho_1 = p.w_1 \\ \rho_0 & \text{if } \rho_0 \text{ is infinite} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The set $(P\,.\,\Sigma)^*\,.\,P \cup (P\,.\,\Sigma)^\omega$ of all guarded strings over P and Σ is denoted by $\mathsf{GS}(P, \Sigma)$; the set $\mathsf{fin}\,\mathsf{GS}(P, \Sigma) = (P\,.\,\Sigma)^*\,.\,P$ denotes the set of all finite guarded strings.

Intuitively, $\rho_0 \bowtie \rho_1$ glues the guarded strings ρ_0 and $\rho_1$together if the last state of ρ_0 and the first state of ρ_1 are equal. Guarded strings are used in the context of labelled transition systems [4] and for the abstract interpretation of program schemes [29].

Lemma 4.4. *The powerset algebra* $\mathsf{GUA}(P, \Sigma) =_{df} (\mathcal{P}(\mathsf{GS}(P, \Sigma)), \cup, \emptyset, \bowtie, P)$ *over two alphabets P and* Σ*, with multiplication defined by*

$$L_0 \bowtie L_1 =_{df} \{\rho_0 \bowtie \rho_1 : \rho_0 \in L_0, \rho_1 \in L_1 \text{ and } \rho_0 \bowtie \rho_1 \text{ defined}\},$$

forms a weak Boolean quantale. The algebra $\mathsf{FGUA}(P, \Sigma) =_{df} (\mathcal{P}(\mathsf{fin}\,\mathsf{GS}(P, \Sigma)), \cup, \emptyset, \bowtie, P)$ *of sets of finite guarded strings forms a Boolean subquantale of it that is even full.*

5. Purely finite and purely infinite elements

It turns out that the algebraic view admits a simple but very useful abstract characterisation of sets of finite or infinite elements.

From the definition of $\mathsf{inf}\,A$, $\mathsf{inf}\,A$ and (1) of trajectory composition in Section 2 the following properties are immediate. First, for a process $A \in \mathsf{PRO}$, we have $A \cdot \emptyset = \mathsf{inf}\,A$. Hence a process is purely infinite, i.e., consists of infinite trajectories only, iff $A = \mathsf{inf}\,A = A \cdot \emptyset$. Dually, a process B is purely finite, i.e., consists of finite trajectories only, iff its purely infinite part is trivial, that is, iff $\mathsf{inf}\,B = \emptyset$.

Since these properties are given solely in terms of composition, they generalise in a straightforward way to weak semirings.

Definition 5.1. Assume an idempotent weak semiring S.

1. The purely infinite part of $a \in S$ is $\mathsf{inf}\,a =_{df} a \cdot 0$. We call *a purely infinite* if $a \cdot 0 = a$. This property is equivalent to a being a *left zero*, i.e., to $\forall\, b : a \cdot b = a$.
2. Often there exists a largest purely infinite element N, characterised by $a \le \mathsf{N} \Leftrightarrow a \le a \cdot 0$. The formula is equivalent to $a \le \mathsf{N} \Leftrightarrow a \cdot 0 = a$. In PRO, $\mathsf{N} = \{(d, g) : d = \infty\}$ is the set of all trajectories of infinite length.
3. Dually, we call an element *a purely finite* if $\mathsf{inf}\,a = a \cdot 0 = 0$, i.e., if its purely infinite part is trivial.
4. In many semirings there exists a largest purely finite element F characterised by $a \le \mathsf{F} \Leftrightarrow a \cdot 0 \le 0$. In PRO, $\mathsf{F} = \{(d, g) : d < \infty\}$ consists of all trajectories of finite length.

Looking at the example $\mathsf{GUA}(P, \Sigma)$ of guarded strings presented in the previous section, we see that these definitions perfectly reflect the intuition: For an element $L \in \mathsf{GUA}(P, \Sigma)$ the set $\mathsf{inf}\,L$ contains all guarded strings of L with infinite length; whereas $\mathsf{fin}\,L$ selects all finite words of L.

By neutrality 1 is purely finite. Moreover, by isotony, all elements ≤ 1 are also purely finite. Next, it is easy to check that the sets of purely finite and purely infinite elements each are closed under $+$ and $\cdot$.

The definition of N implies, for all a that $a \cdot \mathsf{N} \le \mathsf{N}$ and $\mathsf{N} \cdot a \le \mathsf{N}$. The definition of F implies $\mathsf{F} \cdot \mathsf{F} = \mathsf{F}$.

In a Boolean weak quantale N and F always exist and satisfy

$$\mathsf{N} = \top \cdot 0, \qquad \mathsf{F} = \overline{\mathsf{N}},$$

where $\top =_{df} \overline{0}$ denotes the greatest element.

The decomposability of an element into its purely finite and purely infinite parts is of central importance for our further discussion:

Definition 5.2. An idempotent weak semiring S is called *separated* if for all $a \in S$ we have $a = \mathsf{fin}\, a + \mathsf{inf}\, a$ and $\mathsf{fin}\, a$ and $\mathsf{inf}\, a$ are disjoint, i.e., $\forall b \in S : b \leq \mathsf{fin}\, a \wedge b \leq \mathsf{inf}\, a \Rightarrow b = 0$.

From this definition, we get immediately that, in a separated idempotent weak semiring, N and F exist iff there is a greatest element $\top$. Every Boolean weak semiring is separated, since there $\mathsf{fin}\, a = a \sqcap \mathsf{F}$ and $\mathsf{inf}\, a = a \sqcap \mathsf{N}$. In particular, PRO and GUA are separated. For further details on separation see [41].

The purely finite and purely infinite parts of a composition satisfy

Lemma 5.3. *Let S be a separated weak semiring and $a, b \in S$.*

1. $a \cdot b = \mathsf{inf}\, a + \mathsf{fin}\, a \cdot b$.
2. $\mathsf{inf}\,(a \cdot b) = \mathsf{inf}\, a + \mathsf{fin}\, a \cdot \mathsf{inf}\, b$.
3. $\mathsf{fin}\,(a \cdot b) = \mathsf{fin}\,(\mathsf{fin}\, a \cdot b) = \mathsf{fin}\, a \cdot \mathsf{fin}\, b$.

Part 1 states that the second partner of a composition may have an impact only when the first partner is not infinite. Part 2 is an intuitive decomposition property for the purely infinite part. It also implies that a purely infinite element can never result from composing two purely finite ones. The last statement is the converse: the composition of two elements is purely finite only if both partners are. These properties will be of particular interest in connection with iteration.

Further laws about purely finite and purely infinite parts can be found e.g in [41,25,26].

6. Iteration: weak Kleene and omega algebras

As we have shown in Section 4, finite and infinite iteration are important aspects for the analysis of hybrid systems. Moreover, we have illustrated that fixpoints of $x = a \cdot x$ should be used. Therefore, we now turn to an algebraic characterisation of iteration.

Definition 6.1. 1. A *weak Kleene algebra* [42] is a structure $(S, {}^*)$ consisting of a weak idempotent semiring S and an operation * for iterating an element an arbitrary but finite number of times. Such an operation has to satisfy the left *unfold* and *induction* axioms

$$1 + a \cdot a^* \leq a^*, \qquad b + a \cdot c \leq c \Rightarrow a^* \cdot b \leq c. \tag{3}$$

2. Given two weak Kleene algebras S, T, a mapping $\phi : S \to T$ is called a *weak Kleene algebra homomorphism* if it satisfies $\phi(0) = 0$, $\phi(1) = 1$, $\phi(a + b) = \phi(a) + \phi(b)$, $\phi(a \cdot b) = \phi(a) \cdot \phi(b)$ and $\phi(a^*) = \phi(a)^*$.
3. A *weak omega algebra* [19,41] is a pair $(S, {}^\omega)$ such that S is a weak Kleene algebra and $^\omega$ satisfies the *unfold* and *coinduction* axioms

$$a^\omega = a \cdot a^\omega, \qquad c \leq a \cdot c + b \Rightarrow c \leq a^\omega + a^* \cdot b. \tag{4}$$

These axioms imply that $a^* \cdot b$ is the least fixpoint of $b + a \cdot x = x$ and that a^ω is the greatest fixpoint of $a \cdot x = x$; the least fixpoint of $a \cdot x = x$ is 0 if a is right-strict. This entails that * and $^\omega$ are isotone w.r.t. the natural order $\leq$.

Two further consequences of these axioms are that each omega algebra has the greatest element $\top =_{df} 1^\omega$, more generally, $a \leq a \cdot a \Rightarrow a^\omega = a \cdot \top$, and that $a^\omega = a^\omega \cdot \top$ (see [41]). This is the formal reflection of the phenomena concerning infinite iteration of subidentities (elements less than or equal to 1, which model stepping on the spot or *idling*) or Zeno effects which were discussed in the introduction.

We can guarantee the existence of these operations in weak quantales, since every weak quantale is also a complete lattice and hence the Knaster–Tarski fixpoint theorem applies.

Lemma 6.2 (*[41]*)**.** *For a function f let $\mu x \,.\, f(X)$ and $\nu x \,.\, f(x)$ denote the least and greatest fixpoint of f, resp.*

1. *Every weak quantale can be extended to a weak Kleene algebra by defining $a^* =_{df} \mu x \,.\, a \cdot x + 1$.*
2. *If the weak quantale is a completely distributive lattice then it can be extended to a weak omega algebra by setting $a^\omega =_{df} \nu x \,.\, a \cdot x$. In this case, $\nu x \,.\, a \cdot x + b = a^\omega + a^* \cdot b$.*

This definition of omega has already been used in Section 2 for the weak quantale PRO and applies to GUA as well. Hence we can use all the general laws for finite iteration * and infinite iteration $^\omega$ for processes and guarded strings. Here is a collection of such laws, most of which are well known from the theory of omega-regular languages.

Lemma 6.3. *Assume a weak omega algebra. Then omega is isotone, i.e.,* $a \le b \Rightarrow a^\omega \le b^\omega$ *and the following omega-regular laws hold:*

$$\begin{array}{ll} a^\omega \cdot a^\omega \le a^\omega, & (a^\omega)^\omega \le a^\omega, \\ a^+ \cdot a^\omega = a^\omega, & a^* \cdot a^\omega = a^\omega, \\ a \cdot (b \cdot a)^\omega = (a \cdot b)^\omega, & a^\omega \cdot b \le a^\omega, \\ (a \cdot b)^\omega \le (a+b)^\omega, & (a+b)^\omega = (a^* \cdot b)^\omega + (a^* \cdot b)^* \cdot a^\omega, \\ (a+b)^\omega = a^\omega + (a^* \cdot b) \cdot (a+b)^\omega. & \end{array}$$

So far we have introduced algebraic characterisations of purely finite and purely infinite elements as well as of finite and infinite iteration. Let us now explore the interplay between these concepts. First, the set of purely finite elements is also closed under * in a weak Kleene algebra. Moreover we get the following laws.

Lemma 6.4. *Let S be a separated weak omega algebra and* $a \in S$.

1. $a^\omega = (\mathsf{fin}\, a)^* \cdot \mathsf{inf}\, a + (\mathsf{fin}\, a)^\omega$,
2. $\mathsf{inf}\, a^\omega = (\mathsf{fin}\, a)^* \cdot \mathsf{inf}\, a + \mathsf{inf}\,((\mathsf{fin}\, a)^\omega)$,
3. $\mathsf{fin}\, a^\omega = \mathsf{fin}\,((\mathsf{fin}\, a)^\omega) \le (\mathsf{fin}\, a)^\omega$.

The proofs are again straightforward or can be found in [41].

If a is a process, Part 1 says that infinite iteration of trajectories from a can take two forms: it may proceed a while with finite trajectories, but then adds an infinite trajectory which prohibits further iteration — or it keeps iterating finite trajectories forever. Part 2 says that infinite behaviour results from entering an infinite part after a finite iteration of finite parts of the iterated process or by iterating finite parts of that process that all have long enough durations that their infinite iteration takes infinite duration. Part 3 fits well with intuition, since in PRO it means that Zeno effects (infinite iterations that take finite duration) can only occur when some trajectories in a process a are finite.

Lemma 6.5. *In a weak omega algebra,* $a^\omega = a$ *if a is purely infinite.*

Let us point out further simple consequences of the definition of omega in the setting of weak omega algebra. In this, we generalise from PRO.

Definition 6.6. An element a of a weak omega algebra with N and F is called *divergent* or *Zeno-free*, if $a^\omega \le \mathsf{N}$. It is called *Zeno* if it is not Zeno-free and it is called *convergent* if $a^\omega \le \mathsf{F}$.

The least element 0 is the only element which is convergent, divergent and Zeno-free, since $0^\omega = 0$. Moreover, by transitivity of $\le$, if a is Zeno-free and $b \le a$ then b is Zeno-free, too.

Lemma 6.7. *In a full omega algebra (where* 0 *is also a right annihilator) every element is convergent.*

The following lemma provides an important necessary condition for Zeno-freeness.

Lemma 6.8. *In a Boolean weak omega algebra, if a is Zeno-free then* $a \sqcap 1 = 0$.

Proof. Since the algebra is Boolean, $a \sqcap 1$ is idempotent w.r.t. composition and hence $(a \sqcap 1)^\omega = (a \sqcap 1) \cdot \top$. Now, by isotony and neutrality of 1,

$$a \sqcap 1 \le (a \sqcap 1) \cdot \top = (a \sqcap 1)^\omega \le a^\omega \le \mathsf{N},$$

i.e., $a \sqcap 1 \le \mathsf{N}$. Taking the meet with 1 on both sides gives $a \sqcap 1 \le \mathsf{N} \sqcap 1 = 0$. □

7. Embedding processes into guarded strings

To analyse Zeno phenomena, it is useful to get a more detailed view how the trajectories in A^ω for a process $A \in \mathsf{PRO}$ are constructed. Since the gluing of trajectories "forgets" the point of gluing, it cannot always be reconstructed from a trajectory $\tau \in A^\omega$ from which pieces precisely it was assembled. Therefore we want to record possible construction histories of τ in the form of infinite guarded strings which contain the pieces. To this end we first use an injection ι to represent the process A as a language $L \in \mathsf{GUA}(P, \Sigma)$ of guarded strings (cf. Definition 4.3) with suitable P and Σ. The infinite iteration L^ω then consists of all construction histories that use infinitely many pieces from A.

It is easy to see that all guarded strings ρ in L^ω have infinite length if their length is greater than 1. To glue all the pieces recorded in such a ρ together into one overall trajectory we apply a suitable homomorphism ϕ.

A diagrammatic description of this approach is given in Fig. 4.

We now specialise Σ to the set $\mathsf{fin}\,(\mathsf{TRA})$ of *finite* trajectories and P to I, where I is the set of all zero-length trajectories. Hence the elements of I correspond to sets of values, i.e., to subsets of V. For abbreviation we set $\mathsf{FGFT} =_{df} \mathsf{FGUA}(I, \mathsf{fin}\,(\mathsf{TRA}))$ (the mnemonic is "finite guarded strings of finite trajectories"). To define an embedding of purely finite processes into FGFT we define a function ι that maps a trajectory $\tau = (d, g)$ with finite duration d to a guarded string of length 3:

$$\iota(\tau) =_{df} \underline{g(0)} \,.\, \tau \,.\, \underline{g(d)}.$$

$$\begin{tikzcd} \mathsf{GUA}(P, \Sigma) \arrow[rr, "\omega"] & & \mathsf{GUA}(P, \Sigma) \arrow[d, "\phi"] \\ \mathsf{PRO} \arrow[u, "\iota"] & & \mathsf{PRO} \end{tikzcd}$$

Fig. 4. Embedding Trajectories into Guarded Strings.

Here $\rho_1 . \rho_2$ denotes concatenation of ρ and σ, as described before. This construction makes the initial and the final value of τ explicit. A zero-duration trajectory $\underline{v}$ is also mapped to a guarded string of length 3, namely $\underline{v}.\underline{v}.\underline{v}$. Again, we lift ι pointwise to a function $\iota : \mathsf{fin}\,(\mathsf{TRA}) \to \mathsf{FGFT}$. Note that ι is not a weak-semiring homomorphism, since, e.g., $\iota(I) = \{\underline{v}.\underline{v}.\underline{v} \mid v \in V\}$ which is different from the unit I of FGFT. This is intentional: whereas at the level of trajectories we have $\underline{v} \cdot \underline{v} = \underline{v}$, at the level of guarded strings we have $\iota(\underline{v}) \bowtie \iota(\underline{v}) = \underline{v}.\underline{v}.\underline{v}.\underline{v}.\underline{v} \neq \underline{v}.\underline{v}.\underline{v} = \iota(\underline{v})$. Thus, while repetition of a zero-length trajectory cannot be noticed, it can very well be noticed in the embedding. In particular, an infinite repetition of a zero-length trajectory, which amounts to zero-length infinite stepping on the spot, can be represented by an infinite guarded string.

The above definition preserves the condition for definedness of composition, i.e., $\tau_1 \cdot \tau_2$ is defined if and only if $\iota(\tau_1) \bowtie \iota(\tau_2)$ is defined. This is the reason why we use guarded strings with gluing rather than plain sequences of trajectories with concatenation as our representation of construction histories.

Furthermore, by general results about pointwise lifting, we get the following result.

Corollary 7.1. *The mapping ι is disjunctive. In particular, $\iota(A \cup B) = \iota(A) \cup \iota(B)$. Moreover, $\iota(\emptyset) = \emptyset$.*

The composition of images of purely finite processes under ι then yields alternating sequences of elements of I and $\mathsf{fin}\,(\mathsf{TRA})$ in $\mathsf{GFT} =_{df} \mathsf{GUA}(I, \mathsf{fin}\,(\mathsf{TRA}))$. The sequences might have infinite length.

Next we construct a homomorphism from finite guarded strings to processes. Later on we will extend this to infinite strings. For finite guarded strings a projection from FGFT to TRA is inductively defined by

$$\phi(\underline{v}) = \underline{v} \quad \text{and} \quad \phi(w.\tau) = \phi(w) \cdot \tau \ ,$$

where $v \in V$ and $\tau \in \mathsf{fin}\,\mathsf{TRA}$. By this definition we immediately get $\phi(\iota(\tau)) = \tau$ and $\phi(\rho_1 \bowtie \rho_2) = \phi(\rho_1) \cdot \phi(\rho_2)$. Lifting ϕ pointwise to sets of guarded strings yields the following result.

Lemma 7.2. *$\phi : FGFT \to \mathsf{PRO}$ is a weak-Kleene-algebra homomorphism. Moreover by pointwise lifting, ϕ is also disjunctive.*

Proof. Except for the equation for finite iteration all calculations are straightforward and follow either from the definitions or from pointwise lifting. The star equation is shown by fixpoint fusion (cf. [1]). The fusion law states that $g \circ h = f \circ g \Rightarrow g(\mu h) = \mu f$ if f, g and h are isotone functions and g is continuous and strict. For a set Γ of guarded strings we choose $f(x) = \phi(\Gamma) \cdot x + \phi(1)$, $g(x) = \phi(x)$ and $h(x) = \Gamma \cdot x + 1$. By definition all these functions are isotone and g is continuous and strict. Moreover we have

$$g(h(x)) = g(\Gamma \cdot x + 1) = \phi(\Gamma \cdot x + 1) = \phi(\Gamma) \cdot \phi(x) + \phi(1) = f(\phi(x)) = f(g(x)).$$

The third step follows by additivity and multiplicativity of ϕ. Hence by fixpoint fusion we have $g(\mu h) = \mu f$. In particular, for an element $\Gamma \in \mathsf{FGFT}$ we have $\phi(\Gamma^*) = g(\mu h) = \mu f = \phi(\Gamma)^*$. $\square$

Moreover, $\phi(\iota(A)) = A$ for all purely finite processes A. By universal algebra a Kleene algebra homomorphism preserves (in)equations.

8. Omega iteration for processes

Obviously the homomorphism ϕ cannot be extended directly to infinite guarded strings, since the inductive definition does not work. We define ϕ on an infinite guarded string ρ as the supremum of the ϕ-values of the finite prefixes of ρ, i.e., we calculate the "limit" of all prefix values. The prefix relation on guarded strings is defined as usual: $\rho_1 \in \mathsf{fin}\,\mathsf{GS}(P, \Sigma)$ is a finite prefix of ρ_2, written as $\rho_1 \sqsubseteq \rho_2$, iff there is a $\rho_3 \in \mathsf{GS}(P, \Sigma)$ such that $\rho_1 \bowtie \rho_3 = \rho_2$. Infinite guarded strings are maximal with respect to this order. Moreover, in $\mathsf{GUA}(P, \Sigma)$ each (infinite) guarded string ρ is the supremum of all its (finite) prefixes:

$$\rho = \sup\{\sigma \mid \sigma \sqsubseteq \rho\} = \sup\{\sigma \mid \sigma \sqsubseteq \rho,\ |\sigma| < \infty\}. \tag{5}$$

If ρ has finite length the set of its prefixes is finite, hence ρ is even the maximum of that set.

We now return to our homomorphism ϕ which we want to connect with the prefix relation. To this end we first define a prefix relation on trajectories.

Definition 8.1. The *prefix relation* $\sqsubseteq$ between trajectories $\tau_1 = (d_1, g_1)$ and $\tau_2 = (d_2, g_2)$ is defined as

$$\tau_1 \sqsubseteq \tau_2 \ \Leftrightarrow_{df}\ d_1 \leq d_2 \wedge g_2|_{\mathsf{intv}\, d_1} = g_1,$$

where the stroke $|_X$ means function restriction to set X. The first conjunct on the right-hand side is equivalent to $\mathsf{intv}\, d_1 \subseteq \mathsf{intv}\, d_2$.

Lemma 8.2. *The prefix relation $\sqsubseteq$ on trajectories is a partial order with $\tau_1 \sqsubseteq \tau_2$ if and only if $\exists\, \tau_3 : \tau_1 \cdot \tau_3 = \tau_2$. Infinite trajectories are maximal with respect to this order.*

Now, on finite guarded strings the homomorphism ϕ is $\sqsubseteq$-isotone, i.e.,

$$\rho \sqsubseteq \sigma \Rightarrow \phi(\rho) \sqsubseteq \phi(\sigma). \tag{6}$$

The proof is straightforward using the definition of the prefix relation.

Let us now return to the question how to determine A^ω for a purely finite process A. To describe infinite concatenations of trajectories taken from A, we use the homomorphism ϕ and the fact that each guarded string is the limit of its prefixes.

Definition 8.3. For a guarded string ρ we define the set $\mathsf{pre}(\rho)$ of trajectories that correspond to prefixes of ρ by

$$\mathsf{pre}(\rho) =_{df} \{\phi(\sigma) \mid \sigma \sqsubseteq \rho,\ |\sigma| < \infty\}.$$

Note that $\mathsf{pre}(\rho) \subseteq \mathsf{TRA}$.

Now we exploit the fact that in GUA there are no strings of length 0 and hence there is no possibility of Zeno effects there. In particular, each element ρ of $(\iota(A))^\omega$ has infinite length (cf. Definition 4.3). Moreover, there are an infinite number of prefixes, i.e., $|\mathsf{pre}(\rho)| = \infty$. Infinite iteration then results by passing to some sort of "limit" of $\mathsf{pre}(\rho)$. Unfortunately, in contrast to Eq. (5), the supremum of $\mathsf{pre}(\rho)$ need not exist in PRO. We illustrate this fact by the following example.

Example 8.4. Consider the process $A =_{df} \{(\frac{1}{n^2}, g) \mid g(x) = n^2 \cdot x + n,\ n \in \mathbb{N}\}$, where the time domain D and the value set V are equal to $\mathbb{R}_{\geq 0}$. Some elements of $(\iota(A))^\omega$ are $\underline{1} \,.\, (1, g) \,.\, \underline{2} \,.\, (\frac{1}{4}, g) \,.\, \underline{3} \,.\, (\frac{1}{9}, g) \ldots$ and $\underline{3} \,.\, (\frac{1}{9}, g) \,.\, \underline{4} \,.\, (\frac{1}{16}, g) \,.\, \underline{5} \,.\, (\frac{1}{9}, g) \ldots$. Generally, we get that

$$(\iota(A))^\omega = \left\{ \underline{n} \,.\, \left(\frac{1}{n^2}, g\right) .\, \underline{n+1} \,.\, \left(\frac{1}{(n+1)^2}, g\right) .\, \underline{n+2} \ldots \mid n \in \mathbb{N} \right\}.$$

Let us have a closer look at the element $\underline{1} \,.\, (1, g) \,.\, \underline{2} \,.\, (\frac{1}{4}, g) \ldots$. All finite prefixes of this infinite guarded string have the form $\rho = \underline{1} \,.\, (1, g) \ldots \underline{n}$ ($n \in \mathbb{N}$). By this, $\phi(\rho)$ has duration $\sum_{i=1}^{n} \frac{1}{i^2}$. The supremum of these prefixes is a trajectory over a right-open interval of duration $d_\infty =_{df} \sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$; hence the supremum does not exist in PRO where all trajectories of finite duration have to be defined on closed intervals. When one tries to define a limit trajectory by completing the open interval $[0, d_\infty[$ to a closed one, the problem how to define $g(d_\infty)$ arises. Shortly we will define an extended supremum operator that solves the problem by allowing all possible values $v \in V$ at time d_∞. □

The example of the bouncing ball is quite similar. However, it is much harder to write down all elements of $(\iota(\hat{Z}))^\omega$. This is due to the fact that each bounce can be split at infinitely many places.

Theorem 8.5. *Let A be a purely finite process and let $H : \mathsf{PRO} \to \mathsf{PRO}$ be the function defined by $H(X) =_{df} A \cdot X$.*

1. *If X is expanded by H, i.e., $X \subseteq H(X)$, then for every $\xi \in X$ there is a guarded string $\rho \in \mathsf{GFT}$ such that $\tau \sqsubseteq \xi$ for all $\tau \in \mathsf{pre}(\rho)$.*
2. *$A^\omega = \{\xi \in \mathsf{TRA} \mid \exists\, \rho \in \inf \mathsf{GFT} : \forall\, \tau \in \mathsf{pre}(\rho) : \tau \sqsubseteq \xi\}$.*

Proof. 1. Consider $\xi \in X$. We inductively construct a sequences of prefixes of ξ. Since $X \subseteq A \cdot X$, there are $\tau_0 \in A$ and $\xi_0 \in X$ with $\xi = \tau_0 \cdot \xi_0$. Since $\xi_0 \in X$, we can again do the same step and define trajectories $\tau_1 \in A$ and $\xi_1 \in X$ such that $\xi = \tau_0 \cdot \xi_0 = \tau_0 \cdot \tau_1 \cdot \xi_1$. In general for $\xi_i \in A \cdot H$ there are trajectories $\tau_{i+1} \in A$ and $\xi_{i+1} \in X$ with $\xi_i = \tau_{i+1} \cdot \xi_{i+1}$. By construction $\prod_{i=1}^{n} \tau_i \sqsubseteq \xi$. Now we choose ρ as the supremum of all these trajectories lifted to guarded strings, i.e, $\sup\{\rho \mid \rho \in \Join_{i=1}^{n} \iota(\tau_i),\ n \in \mathbb{N}\}$ and we are done.

2. As a preparation we set $\mathsf{OM}(A) =_{df} \{\xi \in \mathsf{TRA} \mid \exists\, \rho \in \inf \mathsf{GFT} : \forall\, \tau \in \mathsf{pre}(\rho) : \tau \sqsubseteq \xi\}$ and observe that finite trajectories τ are left cancellative w.r.t. composition, i.e., satisfy $\tau \cdot \tau_1 = \tau \cdot \tau_2 \Rightarrow \tau_1 = \tau_2$, provided $\tau \cdot \tau_1$ and $\tau \cdot \tau_2$ are defined. Moreover, by omega unfold every guarded string $\rho \in (\iota(A))^\omega$ has a prefix $\sigma_0 \in \iota(A)$ with $\sigma_0 \sqsubseteq \rho$.

   Now we show that $\mathsf{OM}(A)$ is expanded by H. Consider an arbitrary $\xi \in \mathsf{OM}(A)$. By definition there is a $\rho \in \inf \mathsf{GFT}$ with $\tau \sqsubseteq \xi$ for all $\tau \in \mathsf{pre}(\rho)$. By this and the above remark we know that there is a $\sigma_0 \sqsubseteq \rho$ with $\phi(\sigma_0) \in \mathsf{pre}(\rho)$ and $\phi(\sigma_0) \in \phi(\iota(A)) = A$. Then by finiteness of $\phi(\sigma_0)$ and the above cancellation property, there is a unique τ_1 with $\xi = \phi(\sigma_0) \cdot \tau_1$. Hence $\mathsf{OM}(A) \subseteq A \cdot \mathsf{OM}(A)$.

   Together with Part 1 this means that $\mathsf{OM}(A)$ is the greatest expanded element of H and hence its greatest fixpoint. Now the claim follows by Lemma 6.2.2. □

Part 2 shows again that omega iteration is too loose. Every trajectory in A^ω initially agrees with all the prefixes of some infinite guarded string. But it is completely undetermined how the trajectory continues after all these prefixes. Therefore it seems much more natural to determine a supremum $\sup(\mathsf{pre}(\rho))$ of all prefixes rather than allowing all extensions. Defining an adequate supremum and setting up a new iteration operator will be the content of the following sections.

9. A more precise iteration operator

As we have seen in Sections 3 and 8, A^ω is not completely adequate for reasoning about and exclusion of Zeno effects. For many purposes its extension-closedness gets in the way, since it yields a too loose description of infinite iteration.

For that reason we introduce another iteration operator † (in words: dagger) which narrows down the set of possible behaviours. However, in contrast to omega, its definition works up to now only for a special class of time domains.

Definition 9.1. A time domain D is *complete* if it is a complete lattice, i.e., D contains suprema for all its subsets.

To describe †, we use a supremum operator for $\mathsf{pre}(\rho)$ which essentially yields the proper supremum, when possible, and otherwise performs a suitable completion.

First we show how to combine the functions in a compatible set of trajectories into a single one.

Definition 9.2. Assume a complete time domain and let C be a chain of trajectories under the prefix ordering.

1. We set $d_C =_{df} \sup(D_C)$ where $D_C =_{df} \{d \mid (d, g) \in C\}$.
2. Then the *combined function* $\mathsf{comb}_C : [0, d_C[\to V$ is given by

   $$\mathsf{comb}_C(t) =_{df} g(t) \quad \text{for any } (d, g) \in C \text{ with } t \le d.$$

   Note that such a trajectory (d, g) must exist. Otherwise, by linearity of $\le$, we would have $d < t$ for all $d \in D_C$, i.e., t would be an upper bound of D_C and hence $d_C \le t$ by definition of d_C. But this contradicts $t \in [0, d_C[$.

This is well defined, since by definition of the prefix order $\sqsubseteq$ all trajectories in C whose duration exceeds t agree at time point t.

Now we can define the extended supremum operator. Let ρ be a guarded string and $C = \{\phi(\sigma) \mid \sigma \sqsubseteq \rho, |\sigma| < \infty\} = \mathsf{pre}(\rho)$ the corresponding chain of finite prefixes. There are three cases.

1. The duration $d_C = \infty$. Then ρ does not reflect a Zeno effect, and hence the set of prefixes has, as its limit, an infinite trajectory that is the normal supremum of the chain. Since we look at a single guarded string ρ, the set $\{\mathsf{pre}(\rho)\}$ is indeed a chain of finite prefixes of ρ.
2. The duration $d_C \ne \infty$ and ρ becomes "stationary", i.e., ends in an infinite sequence with trajectories of duration zero and identical value v. This means the special kind of Zeno behaviour of idling forever. In this case the chain of prefixes has a largest element which again is its normal supremum.
3. For the duration $d_C \ne \infty$ we have $d_C \ne \{d \mid (d, g) \in C\}$, i.e., $d_C > d$ for all trajectories $(d, g) \in C$. Then ρ reflects proper Zeno behaviour where the trajectories become longer and longer without ever reaching the "limit time" d_C. In this case we can only form a "pseudo-supremum", since there is no unique value to be used for defining the behaviour at the limit time d_C. We define the behaviour of the "limit" on the interval $[0, d_C[$ using the above function comb_C and pad that in all possible ways, i.e., with arbitrary values from the value domain V, at time d_C.

These three cases are sketched in Fig. 5.

Definition 9.3. For a guarded string ρ we define the *extended supremum* $\widehat{\sup}(C) \in \mathsf{PRO}$ of its corresponding prefix set $C = \mathsf{pre}(\rho)$ by

$$\widehat{\sup}(C) =_{df} \begin{cases} \{\sup(C)\} & \text{if } d_C = \infty \\ \{(d_C, g)\} & \text{if } (d_C, g) \in C \\ \{(d_w, \hat{g}) \mid \hat{g}(d_w) = v,\ v \in V, \\ \qquad \hat{g}(t) = \mathsf{comb}_C(t) \text{ if } t < d_w\} & \text{otherwise.} \end{cases}$$

The function ϕ from Section 8 can be extended to infinite guarded strings by setting

$$\phi(\rho) =_{df} \widehat{\sup}(\mathsf{pre}(\rho)) \text{ if } |\rho| = \infty,$$

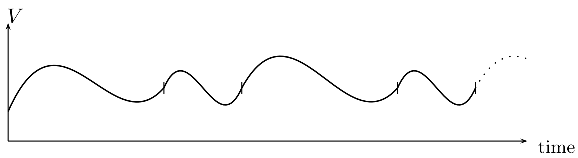
which, in turn, can be lifted pointwise to sets of guarded strings.

Unfortunately, this lifted ϕ is not a homomorphism any longer, since in general $\phi(\rho \bowtie \sigma) \ne \phi(\rho) \cdot \phi(\sigma)$ if ρ has infinite length. However, ϕ still commutes with multiplication and $\phi(\rho \bowtie \sigma) = \phi(\rho) \cdot \phi(\sigma)$ if ρ is finite (σ might be infinite).

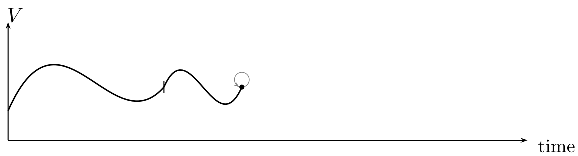
Corollary 9.4. *If A is a purely finite process then $A^\omega = \phi((\iota(A))^\omega) \cdot \top$.*

This corollary states that $\phi((\iota(A))^\omega)$ is equivalent to omega iteration provided arbitrary suffixes are added to its trajectories. In particular, every trajectory of $\phi((\iota(A))^\omega)$ coincides with one of A^ω on all time points till a Zeno point is reached. Looking at our example that means that $\phi((\iota(A))^\omega)$ contains at least the trajectory of Fig. 1. As we will show, $\phi((\iota(A))^\omega)$ consists of exactly those trajectories which "end" at a Zeno point (if there is one); it excludes any extensions after such a point has been reached.

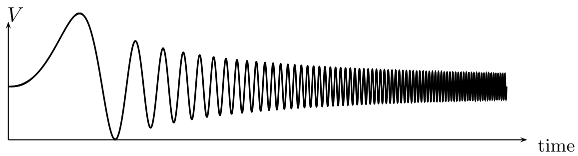
Now we are ready for the definition of our more precise iteration operator.



(a) No Zeno effect.



(b) "Stationary Zeno effect" – looping on the spot.



(c) "Real" Zeno effect.

Fig. 5. Different situations for infinite iteration.

Definition 9.5. For a purely finite process A, we define $A^\dagger =_{df} \phi((\iota(A))^\omega)$. For an arbitrary process A we set $A^\dagger = (\mathsf{fin}\,A)^* \cdot \mathsf{inf}\,A + (\mathsf{fin}\,A)^\dagger$ (cf. Lemma 6.4.1).[1]

The whole construction of † is summarised in the diagram of Fig. 6.

$$\begin{tikzcd}
\mathsf{FGUA}(I, \mathsf{TRA}) \arrow[r, "\omega"] & \mathsf{inf}\,\mathsf{FGUA}(I, \mathsf{TRA}) \arrow[d, "\phi"] \\
\mathsf{PRO} \arrow[u, "\iota"] \arrow[r, "\dagger"] & \mathsf{PRO}
\end{tikzcd}$$

Fig. 6. Construction of †.

This gives another characterisation for infinite iteration in PRO, which respects Zeno behaviour. With this construct, Zeno effects can be excluded by considering only the properly infinite trajectories in $\mathsf{inf}\,A^\dagger = A^\dagger \cap \mathsf{N}$. This could not be achieved reasonably with A^ω, since that includes trajectories which are infinite because they add an arbitrary infinite behaviour to a Zeno initial part. This is made precise by Part 1 of Theorem 8.5. Since the definition is based on omega iteration in GFT and projection ϕ we get for an arbitrary set of guarded strings $L \in \mathsf{GFT}$

$$\phi(L^\omega) = (\phi(L))^\dagger \tag{7}$$

if $L \cap I = \emptyset$. Moreover, from Definition 9.5 we get immediately

Corollary 9.6. *Infinite iteration of a zero-duration process is stationary, that is $P^\dagger = P$ when $P \subseteq I$. In particular we have $I^\dagger = I$ for the multiplicative identity I of* PRO, *whereas $I^\omega = \top = \mathsf{TRA}$.*

Theorem 9.7. *Let H be as in Theorem 8.5.*

1. *$A^\dagger$ is a fixpoint of H.*
2. *Let X be expanded by H, i.e., assume $X \subseteq H(X)$. Then every $\tau \in X$ has a prefix in $A^\dagger$.*

1 The notation † for an iteration operator seems to be due to Elgot (e.g. [22]). We feel that its use is justified, since it is similar in spirit to the one used in iterative algebraic theories (e.g. [15]).

3. $A^\omega = A^\dagger \cdot \top$.

Proof. 1. The proof is a straightforward calculation. To increase readability we write ιA instead of $\iota(A)$. We first remember that $\phi(\iota A) = A$. From this we get by definition of dagger, property of ϕ and omega unfold

$$A \cdot A^\dagger = A \cdot \phi((\iota A)^\omega) = \phi(\iota A) \cdot \phi((\iota A)^\omega) = \phi(\iota A \cdot (\iota A)^\omega) = \phi((\iota A)^\omega) = A^\dagger.$$

2. Consider an arbitrary $\xi \in X \subseteq A \cdot X$. By Theorem 8.5 there is a set $\mathsf{pre}(\rho)$ with $\tau \sqsubseteq \xi$ for all $\tau \in \mathsf{pre}(\rho)$. By definition of $\widehat{\sup}(\mathsf{pre}(\rho))$ we have for all $\sigma \in \widehat{\sup}(\mathsf{pre}(\rho))$ and all $\tau \in \mathsf{pre}(\rho)$ that $\tau \sqsubseteq \sigma$. If $\widehat{\sup}(\mathsf{pre}(\rho))$ contains only one single trajectory (no proper Zeno effect occurs) then $\sigma_0 =_{df} \widehat{\sup}(\mathsf{pre}(\rho)) \sqsubseteq \xi$. In the case of Zeno effects there is a trajectory $\sigma_0 \in \widehat{\sup}(\mathsf{pre}(\rho))$ with $\sigma_0 \sqsubseteq \xi$. σ is the "limit" of all $\tau \in \mathsf{pre}(\rho)$ that coincide with ξ for all time points $t \in \mathsf{intv}\, d_\tau$, where d_τ is the duration of τ; hence $\sigma \sqsubseteq \xi$. Since $\sigma_0 \in \widehat{\sup}(\mathsf{pre}(\rho) \subseteq A^\dagger$, we are done.
3. The claim directly follows from Corollary 9.4 and the definition of dagger. □

An immediate consequence of Part 3 is that $A^\dagger$ and A^ω coincide if A is Zeno-free.

Lemma 9.8. *For an arbitrary process A,*

$$A^\dagger \le \mathsf{N} \Leftrightarrow A^\omega \le \mathsf{N} \quad \textit{and} \quad A^\omega \le \mathsf{N} \Rightarrow A^\dagger = A^\omega.$$

Further properties of dagger follow from the general ones derived in the next section.

10. An axiomatisation

We have shown that the greatest fixpoint of $a \cdot x = x$ is too loose and given a definition for a more appropriate fixpoint in the concrete algebra PRO. In this section we abstract this construction into the setting of weak semirings and omega algebras.

As a preparation we observe that Theorem 9.7.3 can be restated.

Lemma 10.1. *Assume $A, B \in \mathsf{PRO}$ such that $A \cdot B = B$. Then the following properties are equivalent.*

1. $A^\omega = B \cdot \mathsf{TRA}$.
2. $\forall C \in PRO : C = A \cdot C \Rightarrow C \subseteq B \cdot \mathsf{TRA}$.

Proof. ($1 \Rightarrow 2$) Since A^ω is the greatest solution of $X = A \cdot X$ it follows that $C \subseteq A^\omega = B \cdot \mathsf{TRA}$.
($2 \Rightarrow 1$) Since $A^\omega = A \cdot A^\omega$, we infer $A^\omega \subseteq B \cdot \mathsf{TRA}$. Moreover, by associativity,

$$A \cdot (B \cdot \mathsf{TRA}) = (A \cdot B) \cdot \mathsf{TRA} = B \cdot \mathsf{TRA},$$

and hence $B \cdot \mathsf{TRA} \subseteq A^\omega$. □

This motivates the following definition.

Definition 10.2. Let a and x be elements of an arbitrary semiring. We call x a *fixpoint of a* if $x = a \cdot x$. An element c is *spanning for a* if $x \le c \cdot \top$ for all fixpoints of a. A *spanning fixpoint of a* is a fixpoint of a that is also spanning for a.

Next we observe

Lemma 10.3. *Let x be a fixpoint of a in a weak Kleene algebra. Then x is a fixpoint of a^* as well.*

Proof. $x \le a^* \cdot x$ follows by $1 \le a^*$ and isotony. For the reverse inequation we calculate, using star induction and $+$ decomposition,

$$a^* \cdot x \le x \Leftarrow x + a \cdot x \le x \Leftrightarrow a \cdot x \le x. \quad \square$$

Now we give our abstract definition of the † operator. In it we use a generalisation of the notion of being spanning which will enable us to set up a simple connection with omega algebras.

Definition 10.4. A *dagger construction* is a tuple (T, S, ι, ϕ) such that $T = (T, \oplus, 0, \odot, 1, ^*, ^\omega)$ is a weak omega algebra, $S = (S, +, 0, \cdot, 1^*)$[2] is a separated weak Kleene algebra with greatest element $\top$ and $\iota : \mathsf{fin}\,(S) \to \mathsf{fin}\,(T)$ and $\phi : T \to S$ are functions with $\phi(\mathsf{fin}\, T) \subseteq \mathsf{fin}\, S$ that satisfy the following conditions, where, for $a \in S$, we set

$$a^\dagger =_{df} \begin{cases} \phi((\iota(a))^\omega) & \text{if } a \in \mathsf{fin}\, S, \\ (\mathsf{fin}\, a)^* \cdot \mathsf{inf}\, a + (\mathsf{fin}\, a)^\dagger & \text{otherwise.} \end{cases}$$

(1) ι distributes through $+$, i.e., $\iota(a + b) = \iota(a) \oplus \iota(b)$ for all $a, b \in S$.

[2] We overload the symbols 0, 1, $\le$ and *.

(2) ϕ is nearly homomorphic w.r.t the regular operators, i.e., for all $x, y \in T$,

$$\phi(x \oplus y) = \phi(x) + \phi(y), \qquad \phi(x^*) = \phi(x)^*, \quad \text{if } x \in \mathsf{fin}\, T \text{ then } \phi(x \odot y) = \phi(x) \cdot \phi(y)\ .$$

(3) ϕ is inverse to ι, i.e., for all $a \in \mathsf{fin}\,(S)$, we have $\phi(\iota(a)) = a$.
(4) ϕ projects omega to dagger, i.e., if $x \in \mathsf{fin}\, T$ then $\phi(x^\omega) = (\phi(x))^\dagger$.
(5) For all $a, b \in S$ the element $a^\dagger$ is *spanning for* a, b, i.e., for all $c \in S$ with $c \le a \cdot c + b$ we have $c \le a^\dagger \cdot \top + a^* \cdot b$. Note that an element is spanning for a in the old sense iff it is spanning for a and $b = 0$ in this new sense.
(6) For all subidentities $p \le 1$ ($p \in S$), we have $p^\dagger = p$. In particular $1^\dagger = 1$.

From Parts (1) and (2) we get immediately that ι and ϕ are isotone. Moreover, by Part (3) ι is injective and ϕ is surjective. This implies that also $\phi(0) = 0$ and $\phi(1) = 1$, so that ϕ is a homomorphism between weak Kleene algebras. Moreover, the formula of Part (4) is the abstract counterpart of Eq. (7).

Definition 10.5. A *dagger algebra* is a tuple $S = (S, +, \cdot, 0, 1, {}^*, {}^\dagger)$ such that the reduct $(S, +, \cdot, 0, 1, {}^*)$ is a weak Kleene algebra and there is a weak omega algebra T such that (T, S, ι, ϕ) is a dagger construction defining † as given above.

Currently it is not clear whether a given weak Kleene algebra can be extended to a dagger algebra in different ways. A more direct axiomatisation would be preferable, notably one from which uniqueness and existence can be inferred. However, it is difficult to determine precisely where the element $a^\dagger$ is located within the lattice of fixpoints of a. It is quite obvious, that it is in general neither the least nor the greatest fixpoint. Moreover, it cannot be constructed similarly to the optimal fixpoint of Manna and Shamir [37,38], since there is only one maximal fixpoint, namely the greatest fixpoint a^ω.

Based on Theorem 9.7 one might conjecture that $a^\dagger$ is the least spanning fixpoint of a. But this is generally not the case as the following counterexample shows. To develop it, we need a new notion.

Definition 10.6. A Boolean weak semiring has the *progress property* if $\overline{1} \cdot \overline{1} \le \overline{1}$.

The progress property means that the composition of non-empty steps leads to a non-empty overall step, i.e., progress (in time) cannot be undone. For instance, PRO and GFT have the progress property. By Boolean algebra, the progress property is equivalent to $1 \le \overline{\overline{1} \cdot \overline{1}}$. The element $\overline{\overline{1} \cdot \overline{1}}$ has been called step by von Karger [49]; it represent elements that cannot be split into non-subidentities.

The progress property entails $\overline{1} \cdot a \le \overline{1}$ and $a \cdot \overline{1} \le \overline{1}$ for all a. In particular, since $a^\dagger = a \cdot a^\dagger$, we can infer from $a \le \overline{1}$ also $a^\dagger \le \overline{1}$.

Moreover, the progress property is equivalent to

$$p \sqcap a \cdot b = (p \sqcap a) \cdot (p \sqcap b)$$

for all $p \le 1$ and arbitrary a, b. For the proofs see [26]. From this we obtain the decomposition properties

$$a \cdot b \sqcap 1 = (a \sqcap 1) \cdot (b \sqcap 1), \qquad a \cdot b \sqcap \overline{1} = (a \sqcap \overline{1}) \cdot b + a \cdot (b \sqcap \overline{1}).$$

Now we can give our counterexample.

Example 10.7. Consider an arbitrary element a of a semiring with the progress property. We will show that the least spanning fixpoint of $x = (a + 1) \cdot x$ is a^*. First, $(a + 1) \cdot a^* = a \cdot a^* + a^* = a^+ + a^* = a^*$, i.e. a^* is a fixpoint of $a + 1$. Next, by Lemma 10.3, $(a + 1)^*$ is spanning for $a + 1$, and by regular algebra $(a + 1)^* = a^*$. Finally, let c be a spanning fixpoint of $a + 1$. Then $c = (a + 1) \cdot c = a \cdot c + c$, i.e., $a \cdot c \le c$. Since c is spanning for $a + 1$ we obtain $a^* \le c \cdot \top$. Hence $1 \le c \cdot \top$ and therefore, by the above decomposition property, $1 = 1 \sqcap 1 \le c \cdot \top \sqcap 1 = (c \sqcap 1) \cdot (\top \sqcap 1) = c \sqcap 1$, which shows $1 \le c$. Altogether we have $1 + a \cdot c \le c$ and star induction shows $a^* \le c$.

But consider now the concrete algebra PRO over the time domain $\mathbb{R}_{\ge 0} \cup \{\infty\}$ and let the process A consist of a single constant trajectory of non-zero length. Then $(A \cup I)^\dagger$ contains one infinite constant trajectory, which however is not contained in A^*. □

Our dagger operator for processes is embedded into the abstract setting as follows.

Theorem 10.8. PRO *enriched by the dagger operation of the previous section is a dagger algebra in the abstract sense.*

Proof. The role of the algebra T is played by GFT. It is straightforward to check that ϕ and ι satisfy the required properties. Most of them have already been shown in the previous sections. □

11. A small calculus for dagger algebra

After defining an algebraic version of †, we now draw some conclusions. In particular we show that many of the well-known properties of omega iteration are still valid in our setting.

First, we state that ι behaves homomorphically under application of ϕ.

Lemma 11.1. *For all $a, b \in S$ we have the following properties.*

1. $\phi(\iota(a \cdot b)) = \phi(\iota(a) \odot \iota(b))$.
2. $(a \cdot b)^\dagger = \phi(\iota(a \cdot b)^\omega) = \phi((\iota(a) \odot \iota(b))^\omega)$.
3. $\phi(\iota(a^*)) = \phi(\iota(a)^*)$.
4. $\phi(\iota(a^+)) = \phi(\iota(a)^+)$.

Another useful consequence of the definition is that ι can be decomposed:

Lemma 11.2. *Assume a dagger algebra S. Then, for $a, b \in \mathsf{fin}\,(S)$, $\iota(a \cdot b) = \iota(\phi(\iota(a) \odot \iota(b)))$.*

Theorem 11.3. *The following properties hold in a dagger algebra.*

1. *Dagger is $\le$-isotone, i.e., $a \le b \Rightarrow a^\dagger \le b^\dagger$.*
2. *$a^\dagger$ is a fixpoint of $a \cdot x = x$, i.e., $a^\dagger = a \cdot a^\dagger$.*
3. $a^\dagger = a^* \cdot a^\dagger$.
4. $(a^+)^\dagger = a^\dagger$.
5. $(a \cdot b)^\dagger \le (a + b)^\dagger$.
6. $(a \cdot b)^\dagger = a \cdot (b \cdot a)^\dagger$.
7. $(a + b)^\dagger = (a^* \cdot b)^\dagger + (a^* \cdot b)^* \cdot a^\dagger$.
8. *If $p \le 1$ then $(p + b)^\dagger = b^\dagger + b^* \cdot p$.*

All these properties bear strong similarities to omega iteration and omega-regular languages. This is not surprising since omega iteration of words (guarded strings) is at the core of the dagger construction. Moreover, † describes, like omega, infinite iteration. Part 6, for example, states that infinite iteration of an element is the same as infinite iteration of an element that has been iterated finitely many times. Note, that the equation $(a^\dagger)^\dagger = a^\dagger$, which holds for $^\omega$ in omega-regular languages, is not valid. This can easily be seen by the example of the bouncing ball. $Y \cdot \hat{Z}^\dagger$ describes the situation of releasing the ball once; whereas $Y \cdot (\hat{Z}^\dagger)^\dagger$ models that the ball is released an infinite number of times. Part 6 shows that the infinite iteration of a composition of a and b is the same as doing first a once and then do an infinite iteration of $b \cdot a$; this is of course again due to the fact that the end of the iteration is never reached. Part 7 states that an infinite iteration of two elements a and b must either take b from time to time (but infinitely many times) or from some time point it chooses to do a forever.

Next we show that every dagger algebra can be made into an omega algebra.

Corollary 11.4. *Assume a dagger algebra $(S, +, \cdot, 0, 1, ^*, ^\dagger)$ and set $a^\omega =_{df} a^\dagger \cdot \top$. Then $(S, +, \cdot, 0, 1, ^*, ^\omega)$ is a weak omega algebra.*

Proof. First, $a \cdot a^\omega = a \cdot a^\dagger \cdot \top = a^\dagger \cdot \top = a^\omega$. Second, assume $c \le a \cdot c + b$. Since $a^\dagger$ is spanning for a and b, we infer $c \le a^\dagger \cdot \top + a^* \cdot b = a^\omega + a^* \cdot b$. □

In the case of a Boolean dagger algebra we have additional interesting properties.

Lemma 11.5. *Assume a Boolean dagger algebra.*

1. $\mathsf{N}^\dagger = \mathsf{N}$ *and* $\top^\dagger = \top$.
2. *Part (6) of Definition 10.4 follows from the other parts if the algebra satisfies $1^\dagger = 1$ and has the progress property.*
3. $a^\dagger = (a \sqcap \overline{1})^\dagger + (a \sqcap \overline{1})^* \cdot (a \sqcap 1)$. *In particular $(a \sqcap \overline{1})^* \cdot (a \sqcap 1) \le a^\dagger$.*

12. Zeno phenomena algebraically

We now continue the algebraic discussion of Zeno phenomena we have started at the end of Section 8 using our dagger operator. Throughout this section we assume a Boolean dagger algebra S which has been enriched to an omega algebra according to Corollary 11.4.

We first study the interplay between purely infinite spanning fixpoints.

Lemma 12.1.

1. *Let $c \le \mathsf{N}$ be spanning for a and d be a fixpoint of a. Then $d \le c$.*
2. *If $c, d \le \mathsf{N}$ are spanning fixpoints of a then $c = d$. In other words, there is at most one purely infinite spanning fixpoint of a.*
3. *If $a^\omega \le \mathsf{N}$ then $a^\omega = a^\dagger$.*

In Definition 6.6 we have called an element a Zeno-free if $a^\omega \leq \mathsf{N}$. For the further analysis we take a more refined view and analyse the iteration of the non-idling part $a \sqcap \overline{1}$ of a.

Definition 12.2. We call *a Zeno-free up to idling* if $a \sqcap \overline{1}$ is Zeno-free, i.e., if $(a \sqcap \overline{1})^\omega \leq \mathsf{N}$.

Intuitively, the definition states that if there is an infinite iteration where each step means real progress, then the resulting element is purely infinite. That means that there cannot be Zeno phenomena.

Now we can prove the following decomposition property for $a^\dagger$.

Lemma 12.3. *If a is Zeno-free up to idling then* $a^\dagger = a^* \cdot (a \sqcap 1) + a^* \cdot (a \sqcap \overline{1})^\omega$.

Proof. From the assumption, Lemma 12.1.3 yields $(a \sqcap \overline{1})^\dagger = (a \sqcap \overline{1})^\omega$. Now the claim is immediate from Theorem 11.3.8. $\square$

In Definition 9.3, we have seen that the extended supremum, and hence the result of the dagger operation, divides into three parts: the infinite trajectories, the eventually idling trajectories and the trajectories with proper Zeno behaviour. We can now recreate this trichotomy algebraically.

Corollary 12.4. *Let a be purely finite.*

1. $a^\dagger = ((a \sqcap \overline{1})^\dagger \sqcap \mathsf{N}) + (a \sqcap \overline{1})^* \cdot (a \sqcap 1) + ((a \sqcap \overline{1})^\dagger \sqcap \mathsf{F})$.
2. *If a is purely finite then each of the three summands in the right-hand side of Part 1 is a fixpoint of a.*

We can process one of the three summands a bit further, since by regular algebra $(a \sqcap \overline{1})^* = a^*$.

Corollary 12.5. *For purely finite a we have* $a^\dagger = ((a \sqcap \overline{1})^\omega \sqcap \mathsf{N}) + a^* \cdot (a \sqcap 1) + ((a \sqcap \overline{1})^\omega \sqcap \mathsf{F})$.

Finally, we look at the purely infinite part.

Corollary 12.6. *For purely finite a,* $(a \sqcap \overline{1})^\omega \sqcap \mathsf{N} = ((a \sqcap \overline{1})^\dagger \sqcap \mathsf{N}) + ((a \sqcap \overline{1})^\dagger \sqcap \mathsf{F}) \cdot \mathsf{N}$.

Proof. Since $(a \sqcap \overline{1})^\dagger$ is spanning, we know $(a \sqcap \overline{1})^\omega = (a \sqcap \overline{1})^\dagger \cdot \top$. Now $\mathsf{fin}\,/\mathsf{inf}$ calculus shows the claim. $\square$

This exhibits again clearly that omega iteration ruthlessly crosses Zeno gaps and adds arbitrary behaviour afterwards.

13. Conclusion and future work

We have presented a construction for a fixpoint that captures phenomena of Zeno effects and idling in a precise way. In some sense it fixes Zeno gaps.

The construction was motivated by an example from hybrid system analysis. There, infinite iteration of the function $f(x) = a \cdot x$ plays a crucial role. So far, mostly the greatest fixpoint has been used to model this kind of iteration. However, as we have also shown in this paper, that is too imprecise. An example is that the greatest fixpoint "guesses" the behaviour after a Zeno point or after idling "forever". Based on the motivating example, we have defined a fixpoint that allows describing Zeno effects in the concrete algebra of hybrid systems. Then the concrete construction was lifted to a purely algebraic setting. In particular our characterisation is first-order with types, more precisely, Horn equational. Hence properties can be proved fully automatically using off-the-shelf theorem provers (e.g., [27]). Moreover we have derived a number of useful properties. Most of them were used in [25], where larger case studies are discussed.

Although the presented axiomatisation is first-order, a more direct axiomatisation would be preferable. Finding such a characterisation is part of our future work. It hopefully will help to analyse when a weak Kleene algebra can be extended to a dagger algebra and whether this extension is unique. At the moment we assume that the introduced fixpoint can be characterised as the sum of three different fixpoints of f. The first one should describe the idling part, the second one the proper Zeno effects and the third one should characterise properly infinite iteration resulting in purely infinite elements. This conjecture is based on the discussion of the previous section.

Another direction for future work is to apply our dagger operator in further case studies. These will include the analysis of hybrid systems in an algebraic setting, but also omega-regular languages. For the latter the connection between the dagger operator and Büchi automata has to be investigated.

Acknowledgements

We are grateful to Han-Hing Dang, Jules Desharnais, Roland Glück and an anonymous referee for valuable comments.

Appendix. Deferred proofs

Proof of 11.1. 1. By Definition 10.4(3) twice and Definition 10.4(2),

$$\phi(\iota(a \cdot b)) = a \cdot b = \phi(\iota(a)) \cdot \phi(\iota(b)) = \phi(\iota(a) \odot \iota(b)).$$

2. The first equation is immediate from the definition. By Definition 10.4(4), Part 1 and Definition 10.4(4) again,

$$\phi(\iota(a \cdot b)^{\omega}) = (\phi(\iota(a \cdot b)))^{\dagger} = (\phi(\iota(a) \odot \iota(b)))^{\dagger} = \phi((\iota(a) \odot \iota(b))^{\omega}).$$

3. By (3) twice and Definition 10.4(2),

$$\phi(\iota(a^*)) = a^* = \phi(\iota(a))^* = \phi(\iota(a)^*).$$

4. By the definition of $^+$, Part 1, Definition 10.4(2), Part 3, Definition 10.4(2) and the definition of $^+$ again,

$$\begin{aligned}\phi(\iota(a^+)) &= \phi(\iota(a \cdot a^*)) = \phi(\iota(a) \odot \iota(a^*)) = \phi(\iota(a)) \cdot \phi(\iota(a^*)) \\ &= \phi(\iota(a)) \cdot \phi(\iota(a)^*) = \phi(\iota(a) \odot \iota(a)^*) = \phi((\iota(a)^+). \quad \square\end{aligned}$$

Proof of 11.2. By Definition 10.4(2) we get $\iota(\phi(\iota(a) \odot \iota(b))) = \iota(\phi(\iota(a)) \cdot \phi(\iota(b))) = \iota(a \cdot b)$. $\square$

Proof of 11.3. We restrict ourselves to the case where the argument of dagger is finite. All other cases can be reduced to that case using the definition.

1. Dagger is defined as the composition of $\leq$-isotone functions.
2. By definition of †, unfold, homomorphism-like behaviour and definition again, we get

$$a^{\dagger} = \phi((\iota(a))^{\omega}) = \phi(\iota(a) \odot (\iota(a))^{\omega}) = \phi(\iota(a)) \cdot \phi((\iota(a))^{\omega}) = a \cdot a^{\dagger}.$$

3. First, $a^{\dagger} = 1 \cdot a^{\dagger} \leq a^* \cdot a^{\dagger}$. For the reverse inequation we use star induction and Part (2):

$$a^* \cdot a^{\dagger} \leq a^{\dagger} \;\Leftarrow\; a^{\dagger} + a \cdot a^{\dagger} \leq a^{\dagger} \;\Leftrightarrow\; \mathsf{TRUE}.$$

4. We calculate

$$\begin{aligned}
& (a^+)^{\dagger} \\
= \quad & \{\!\!\{ \text{ definition of dagger } \}\!\!\} \\
& \phi(\iota(a^+)^{\omega}) \\
= \quad & \{\!\!\{ \text{ by Definition 10.4(4) } \}\!\!\} \\
& (\phi(\iota(a^+)))^{\dagger} \\
= \quad & \{\!\!\{ \text{ by Lemma 11.1.4 } \}\!\!\} \\
& (\phi(\iota(a)^+))^{\dagger} \\
= \quad & \{\!\!\{ \text{ by Definition 10.4(4) } \}\!\!\} \\
& \phi((\iota(a)^+)^{\omega}) \\
= \quad & \{\!\!\{ \text{ weak omega algebra (Lemma 6.3) } \}\!\!\} \\
& \phi(\iota(a)^{\omega}) \\
= \quad & \{\!\!\{ \text{ definition of dagger } \}\!\!\} \\
& a^{\dagger}.
\end{aligned}$$

5. This follows from $a \cdot b \leq (a + b) \cdot (a + b) \leq (a + b)^+$ and Part (4).
6. We calculate

$$\begin{aligned}
& (a \cdot b)^{\dagger} \\
= \quad & \{\!\!\{ \text{ by Lemma 11.1.2 } \}\!\!\} \\
& \phi(\iota(a) \odot \iota(b))^{\omega}) \\
= \quad & \{\!\!\{ \text{ weak omega algebra (Lemma 6.3) } \}\!\!\} \\
& \phi(\iota(a) \odot (\iota(b) \odot \iota(a))^{\omega}) \\
= \quad & \{\!\!\{ \text{ by Definition 10.4(2) } \}\!\!\}
\end{aligned}$$

$$\begin{array}{ll} & \phi(\iota(a)) \cdot \phi((\iota(b) \odot \iota(a))^\omega) \\ = & \quad \{\!\![\text{ by Definition 10.4(2) and Lemma 11.1.1 }]\!\!\} \\ & a \cdot \phi((\iota(b \cdot a))^\omega) \\ = & \quad \{\!\![\text{ definition of dagger }]\!\!\} \\ & a \cdot (b \cdot a)^\dagger . \end{array}$$

7. We calculate

$$\begin{array}{ll} & (a+b)^\dagger \\ = & \quad \{\!\![\text{ definition of dagger }]\!\!\} \\ & \phi((\iota(a+b))^\omega) \\ = & \quad \{\!\![\text{ by Definition 10.4(2) }]\!\!\} \\ & \phi((\iota(a) \oplus \iota(b))^\omega) \\ = & \quad \{\!\![\text{ weak omega algebra (Lemma 6.3) and setting } z =_{df} \iota(a)^* \odot \iota(b)]\!\!\} \\ & \phi(z^\omega \oplus z^* \odot \iota(a)^\omega) \\ = & \quad \{\!\![\text{ by Definition 10.4(2) }]\!\!\} \\ & \phi(z^\omega) + \phi(z^* \odot \iota(a)^\omega) \\ = & \quad \{\!\![\text{ by Definitions 10.4(4) and 10.4(2) }]\!\!\} \\ & \phi(z)^\dagger + \phi(z^*) \cdot \phi(\iota(a)^\omega) \\ = & \quad \{\!\![\text{ by Lemma 11.1.3, Definitions 10.4(4) and 10.4(3) }]\!\!\} \\ & \phi(z)^\dagger + \phi(z)^* \cdot a^\dagger . \end{array}$$

It remains to determine $\phi(z)$:

$$\begin{array}{ll} & \phi(\iota(a)^* \odot \iota(b)) \\ = & \quad \{\!\![\text{ by (2) }]\!\!\} \\ & \phi(\iota(a)^*) \cdot \phi(\iota(b)) \\ = & \quad \{\!\![\text{ by Lemma 11.1.3 and Definition 10.4(3) }]\!\!\} \\ & \phi(\iota(a))^* \cdot b \\ = & \quad \{\!\![\text{ by (3) }]\!\!\} \\ & a^* \cdot b . \end{array}$$

This concludes the calculation.

8. Immediate from the previous part using that $p^* = 1$ and $p^\dagger = p$ when $p \le 1$. □

Proof of 11.5. As a preliminary, we note that in a Boolean weak semiring for elements $p, q \le 1$ we have $p \cdot q = p \sqcap q$.

1. First, $\mathsf{N}^\dagger = \mathsf{N} \cdot \mathsf{N}^\dagger = \mathsf{N}$. Second, by the general definition of dagger, $\top^\dagger = \mathsf{F}^* \cdot \mathsf{N} + \mathsf{F}^\dagger = \mathsf{F}^* \cdot \mathsf{N} + \mathsf{F} \cdot \mathsf{F}^\dagger \ge \mathsf{N} + \mathsf{F} = \top$, since $\mathsf{F}^* \ge 1$ and $F^\dagger \ge 1^\dagger = 1$ by $\mathsf{F} \ge 1$ and isotony of †.
2. First, by isotony and the definition of dagger, $p \le 1$ implies $p^\dagger \le 1^\dagger = 1$. Next, by Theorem 11.3(2) and the introductory remark we have $p^\dagger = p \cdot p^\dagger = p \sqcap p^\dagger$, which means $p^\dagger \le p$.

   By assumption and again the introductory remark we have $p \cdot p = p \sqcap p = p$. Since $p^\dagger$ is spanning for p we obtain, by Boolean algebra, distributivity and neutrality of 1,

   $$p \le p^\dagger \cdot \top = p^\dagger \cdot (1 + \overline{1}) = p^\dagger \cdot 1 + p^\dagger \cdot \overline{1} = p^\dagger + p^\dagger \cdot \overline{1}.$$

   Now we observe that the progress property entails $a \cdot \overline{1} \le \overline{1}$ for arbitrary a, as shown by the calculation

   $$a \cdot \overline{1} = (a \sqcap 1) \cdot \overline{1} + (a \sqcap \overline{1}) \cdot \overline{1} \le 1 \cdot \overline{1} + \overline{1} \cdot \overline{1} = \overline{1}.$$

   Using that we can take the meet with 1 on both sides of the above inequation and obtain

   $$p = p \sqcap 1 \le p^\dagger \sqcap 1 + p^\dagger \cdot \overline{1} \sqcap 1 = p^\dagger \sqcap 1 = p^\dagger.$$

3. This follows from Theorem 11.3.7 by splitting $a = (a \sqcap 1) + (a \sqcap \overline{1})$ and the fact that $x^* = 1$ for $x \le 1$. □

Proof of 12.1. 1. We have $d \le c \cdot \top = c$.

2. Immediate from Part 1 and antisymmetry.
3. Since a^ω is the greatest fixpoint of a and $a^\dagger$ is a fixpoint of a we have $a^\dagger \le a^\omega$. Hence $a^\omega \le \mathsf{N}$ implies $a^\dagger \le \mathsf{N}$. Moreover, both a^ω and $a^\dagger$ are spanning fixpoints of a, so that we can apply Part 2. □

Proof of 12.4. 1. By Theorem 11.3(8) we have $a^\dagger = (a \sqcap \overline{1})^\dagger + (a \sqcap \overline{1})^* \cdot (a \sqcap 1)$. Now the claim follows by splitting the first summand into its purely infinite and purely finite parts.

2. We first note that if x is a fixpoint of $a \sqcap \overline{1}$ then it is also a fixpoint of a, as is shown by the calculation

$$a \cdot x = (a \sqcap 1) \cdot x + (a \sqcap \overline{1}) \cdot x = (a \sqcap 1) \cdot x + x = x,$$

since $a \sqcap 1 \leq 1$.

Now we observe that for $\mathsf{X} \in \{\mathsf{F}, \mathsf{N}\}$, purely finite b and arbitrary c we have $(b \cdot c) \sqcap \mathsf{X} = b \cdot (c \sqcap \mathsf{X})$. Hence $(a \sqcap \overline{1})^\dagger \sqcap \mathsf{X} = ((a \sqcap \overline{1}) \cdot (a \sqcap \overline{1})^\dagger) \sqcap \mathsf{X} = (a \sqcap \overline{1}) \cdot ((a \sqcap \overline{1})^\dagger \sqcap \mathsf{X})$, so that by the above observation $(a \sqcap \overline{1})^\dagger \sqcap \mathsf{X}$ also is a fixpoint of a.

For the remaining summand we set $b =_{df} a \sqcap \overline{1}$ and $c =_{df} a \sqcap 1$ and calculate

$$a \cdot b^* \cdot c = b \cdot b^* \cdot c + c \cdot b^* \cdot c \leq b^+ \cdot c + b^* \cdot c = b^* \cdot c.$$

The reverse inequation reduces by star induction to

$$b \cdot a \cdot b^* \cdot c + c \leq a \cdot b^* \cdot c \;\Leftrightarrow\; b \cdot b \cdot b^* \cdot c + c \cdot c \cdot b^* \cdot c + c \leq b \cdot b^* \cdot c + b \cdot b^* \cdot c$$
$$\Leftrightarrow\; c \leq b \cdot b^* \cdot c + c \cdot b^* \cdot c \;\Leftarrow\; c \leq c \cdot c \;\Leftrightarrow\; \mathsf{TRUE}. \quad \square$$

References

[1] C. Aarts, R. Backhouse, E. Boiten, H. Doornbos, N. van Gasteren, R. van Geldrop, P. Hoogendijk, E. Voermans, J. van der Woude, Fixed-point calculus, Information Processing Letters 53 (3) (1995) 131–136.
[2] R. Alur, C. Courcoubetis, T.A. Henzinger, P.-H. Ho, Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems, in: R.L. Grossman, A. Nerode, A.P. Ravn, H. Rischel (Eds.), Hybrid Systems, in: Lecture Notes in Computer Science, vol. 736, Springer, 1993, pp. 209–229.
[3] A.D. Ames, A. Abate, S. Sastry, Sufficient conditions for the existence of Zeno behavior, in: IEEE Conference on Decision and Control, IEEE Press, 2005.
[4] A. Arnold, Finite Transition Systems, Prentice Hall, 1994.
[5] A. Arnold, M. Nivat, The metric space of infinite trees. Algebraic and topological properties, Fundamenta Informaticae 3 (4) (1980) 445–476.
[6] J.C.M. Baeten, J.A. Bergstra, Process algebra with propositional signals, in: ACP'95: Algebra of Communicating Processes, Elsevier, 1997, pp. 381–405.
[7] J.C.M. Baeten, C.A. Middelburg, Process Algebra with Timing, in: Monographs in Theoretical Computer Science, Springer, 2002.
[8] J.C.M. Baeten, W.P. Weijland, Process Algebra, Cambridge University Press, 1990.
[9] J.A. Bergstra, I. Bethke, A. Ponse, Process algebra with iteration and nesting, The Computer Journal 37 (4) (1994) 243–258.
[10] J.A. Bergstra, J. Klop, Fixed point semantics in process algebra, Technical Report IW 206/82, Centre for Mathematics and Computer Science, 1982.
[11] J.A. Bergstra, J. Klop, Algebra of communicating processes with abstraction, Theoretical Computer Science 37 (1985) 77–121.
[12] J.A. Bergstra, A. Ponse, An instruction sequence semigroup with involutive anti-automorphisms, Scientific Annals of Computer Science 19 (2009).
[13] J.A. Bergstra, J. Tiuryn, Regular extensions of iterative algebras and metric interpretations, Fundamenta Informaticae 4 (4) (1981) 997–1014.
[14] J.A. Bergstra, C.A. Middelburg, Process algebra for hybrid systems, Theoretical Computer Science 335 (2–3) (2005) 215–280.
[15] S. L. Bloom, Z. Ésik, Equational axioms for regular sets, Mathematical Structures in Computer Science 3 (1) (1993) 1–24.
[16] M. Broy, K. Stølen, Specification and Development of Interactive Systems: Focus on Streams, Interfaces, and Refinement, Springer, 2001.
[17] F. Cardone, J.R. Hindley, Lambda-calculus and Combinators in the 20th Century, in: Handbook of the History of Logic, vol. 5, Elsevier, 2009 (Chapter 14).
[18] Y. Chen, A fixpoint theory for non-monotonic parallelism, Theoretical Computer Science 308 (2003) 367–392.
[19] E. Cohen, Separation and reduction, in: R. Backhouse, J.N. Oliveira (Eds.), Mathematics of Program Construction, MPC 2000, in: Lecture Notes in Computer Science, vol. 1837, Springer, 2000, pp. 45–59.
[20] J.H. Conway, Regular Algebra and Finite Machines, Chapman & Hall, 1971.
[21] W. Cook, J. Palsberg, A denotational semantics of inheritance and its correctness, ACM SIGPLAN Notices 24 (10) (1989) 433–443.
[22] C. Elgot, The common algebraic structure of exit-automata and machines, Computing 6 (1970) 349–370.
[23] D. Harel, D. Kozen, J. Tiuryn, Dynamic Logic, MIT Press, 2000.
[24] T.A. Henzinger, The theory of hybrid automata, in: M.K. Inan, M.K. Kurshan (Eds.), Verification of Digital and Hybrid Systems, in: NATO ASI Series F: Computer and Systems Sciences, vol. 170, Springer, 2000, pp. 265–292.
[25] P. Höfner, Algebraic Calculi for Hybrid Systems, Books on Demand GmbH, 2009.
[26] P. Höfner, B. Möller, An algebra of hybrid systems, Journal of Logic and Algebraic Programming 78 (2009) 74–97.
[27] P. Höfner, G. Struth, Automated reasoning in Kleene algebra, in: F. Pfennig (Ed.), Automated Deduction, in: Lecture Notes in Artificial Intelligence, vol. 4603, Springer, 2007, pp. 279–294.
[28] K.H. Johansson, M. Egerstedt, J. Lygeros, S. Sastry, On the regularization of Zeno hybrid automata, Systems & Control Letters 38 (1999) 141–150.
[29] D.M. Kaplan, Regular expressions and the equivalence of programs, NATO ASI Series F: Computer and Systems Sciences 3 (4) (1969) 361–386.
[30] S.C. Kleene, Representation of events in nerve nets and finite automata, Technical Report RM-704, RAND Corporation, 1951, RAND Research Memorandum.
[31] S.C. Kleene, Introduction to Metamathematics, Van Nostrand, 1952.
[32] B. Knaster, Un thèoréme sur les fonctions d'ensembles, Annales de la Societe Polonaise de Mathematique 6 (1928) 133–134.
[33] G. Köstler, W. Kießling, H. Thöne, U. Güntzer, Fixpoint iteration with subsumption in deductive databases, Journal of Intelligent Information Systems 4 (1995) 123–148.
[34] D. Kozen, A completeness theorem for Kleene algebras and the algebra of regular events, Information and Computation 110 (2) (1994) 366–390.
[35] D. Kozen, Automata on guarded strings and applications, Matématica Contemporânea 24 (2003) 117–139.
[36] A. Levy, Basic Set Theory, Springer, 1979, Republished, 2002.
[37] Z. Manna, A. Shamir, The optimal fixedpoint of recursive programs, in: STOC'75: Proceedings of seventh annual ACM symposium on Theory of computing, ACM Press, 1975, pp. 194–206.
[38] Z. Manna, A. Shamir, The theoretical aspects of the optimal fixed point, SIAM Journal on Computing 5 (3) (1976) 414–426.
[39] G. Markowsky, B. Rosen, Bases for chain-complete posets, in: 16th Annual Symposium on Foundations of Computer Science, 13–15 October, 1975, The University of California, Berkeley, CA, USA, IEEE, 1975, pp. 34–47.
[40] T.J. Marlowe, B.G. Ryder, Properties of data flow frameworks: a unified model, Acta Informatica 28 (2) (1990) 121–163.
[41] B. Möller, Kleene getting lazy, Science of Computer Programming 65 (2007) 195–214.
[42] B. Möller, G. Struth, WP is WLP, in: W. MacCaull, M. Winter, I. Düntsch (Eds.), Relational Methods in Computer Science, in: Lecture Notes in Computer Science, vol. 3929, Springer, 2006, pp. 200–211.

[43] K. Nakamura, A. Fusaoka, On transfinite hybrid automata, in: M. Thiele, L. Thiele (Eds.), Hybrid Systems: Computation and Control, in: Lecture Notes in Computer Science, vol. 3414, Springer, 2005, pp. 495–510.
[44] D. Park, On the semantics of fair parallelism, in: D. Bjørner (Ed.), Proceedings of the Abstract Software Specifications, 1979 Copenhagen Winter School, in: Lecture Notes in Computer Science, Springer, 1980, pp. 504–526.
[45] A. Platzer, J.-D. Quesel, KeYmaera: a hybrid theorem prover for hybrid systems, in: A. Armando, P. Baumgartner, G. Dowek (Eds.), Automated Reasoning, in: Lecture Notes in Artificial Intelligence, vol. 5195, Springer, 2008, pp. 171–178.
[46] Z. Qian, Standard fixpoint iteration for Java bytecode verification, ACM Transactions on Programming Languages and Systems 22 (4) (2000) 638–672.
[47] A. Tarski, A lattice-theoretical fixpoint theorem and its applications, Pacific Journal of Mathematics 2 (5) (1955) 285–309.
[48] J. Tiuryn, Unique fixed points vs. least fixed points, Theoretical Computer Science 12 (1980) 229–254.
[49] B. von Karger, Temporal algebra, Mathematical Structures in Computer Science 8 (3) (1998) 277–320.