Fixing Zeno gaps

Peter Höfner, Bernhard Möller

Angaben zur Veröffentlichung / Publication details:

Höfner, Peter, and Bernhard Möller. 2011. "Fixing Zeno gaps." *Theoretical Computer Science* 412 (28): 3303–22. https://doi.org/10.1016/j.tcs.2011.03.018.

Fixing Zeno Gaps

Peter Höfner[1], Bernhard Möller[1]

Institut für Informatik, Universität Augsburg, D-86159 Augsburg, Germany

It is our pleasure to dedicate this paper to Jan Bergstra at 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.

Abstract

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.

Keywords: fixpoints, iteration, semiring, Kleene algebra, omega algebra, hybrid systems

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 in finding the elements in a database. But they also occur in compiler construction and data flow analysis (see [41] for an overview), in the lambda calculus (e.g., [17]), in concurrency [11] or in the verification of bytecode [47]. In denotational semantics, fixpoints are used to define the meaning of recursive definitions [21]. Fixed points are also used in set theory.

The existence of fixpoints can be established in various ways. A fundamental result is the well-known theorem of Knaster and Tarski [33, 48] that guarantees for every isotone endofunction even a complete sublattice of fixpoints. The assumption of a complete lattice can be weakened, e.g. to directed-complete partial orders [40].

If a class of functions happens to have unique fixpoints, things are easy (e.g. [13, 49]). However, generally there is a variety of fixpoints of a given function and one 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 [32]. In case of a non-continuous function, transfinite iteration may be necessary to reach the least fixpoint.

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 [31] 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 [45]. In such cases the least fixpoints are

Email addresses: `hoefner@informatik.uni-augsburg.de` (Peter Höfner), `moeller@informatik.uni-augsburg.de` (Bernhard Möller)

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 optimal fixpoints [38, 39], the greatest lower bound of all maximal fixpoints. They used this special class of fixpoints 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, e.g., 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 small transitions occur within finite time. 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 [27, 26]. 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 a^ω 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 [27], 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 (as yet unpublished) dissertation [26]. 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 define a concrete algebra of hybrid systems, which is used throughout the paper to present examples. In Section 3, we abstract from the concrete model and present the general algebraic background. After that, we show in Section 4 that Zeno effects can occur in hybrid systems (at least theoretically) and therefore the algebra has to be able to model these phenomena. Sections 5 and 6 show that the greatest fixpoint is inadequate for arguing about infinite iteration. To fix this problem we introduce a new fixpoint for the algebra of hybrid systems in Section 7. Before concluding the paper we lift the construction to a purely algebraic level in Section 8, derive some of its essential properties and discuss Zeno behaviour in general algebraic terms in Section 9.

2. An Algebra of Hybrid Systems

To make the paper self-contained we repeat the basic definitions of our model in [25, 27]. It is based on trajectories that reflect the variation of the values of the variables in a system over time.

Let V be a set of *values* and D a set of *durations* (e.g. $\mathbf{N}$, $\mathbf{Q}_{\geq 0}$, $\mathbf{R}_{\geq 0}$, ...). We assume a cancellative addition $+$ on D and an element $0 \in D$ such that $(D, +, 0)$ is a commutative monoid. 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., $d_1 + d_2 = 0 \Leftrightarrow d_1 = d_2 = 0$.

When talking about infinite durations, 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

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

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* $\mathsf{ran}\ (d, g)$.

A special role is played by *zero-length trajectories* of the form $\underline{x} =_{df} (0, g)$ with $x \in V$ and $g(0) =_{df} x$; 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 $g(t + d_1) = g_2(x)$ for all $t \in \mathsf{intv}\, d_2$. This is well defined by cancellativity of $+$ on durations other than ∞. 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 $d = \infty$.

Figure 1 illustrates the main idea for composing trajectories. Sometimes the condition $g_1(d_1) = g_2(0)$ for composing trajectories is too restrictive. In [27], a possibility to relax the condition is given. It allow jumps at the composition point for the function describing the timewise behaviour. However, for the present paper the above definition of composition is sufficient.

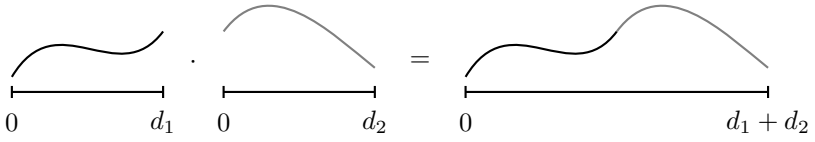

Figure 1: Composition of two finite trajectories

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

The *purely finite* and *purely infinite parts* of a process A are defined as

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

Composition is lifted to processes A, B as follows:

$$A \cdot B =_{df} \mathsf{inf}\, A \cup \{a \cdot b \,|\, a \in \mathsf{fin}\, A,\ b \in B,\ a \cdot b \text{ defined}\} \tag{1}$$

The set I 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. Moreover, $\cdot$ distributes through arbitrary unions in its left argument and through non-empty ones in its right argument. in particular, $\cdot$ is $\subseteq$-isotone in both arguments.

Sets of zero-length trajectories, i.e., subprocesses of I, correspond to sets of values and can be used to restrict processes. Let R be such a set and A be an arbitrary process. Then $R \cdot A$ consists of those trajectories of A whose initial value lies in R, while $A \cdot R$ is the set of trajectories of A whose final value, if any, is in R. Algebraically, such elements below the multiplicative identity are known as *tests* [37, 35] and therefore we set $\mathsf{test}(\mathsf{PRO}) =_{df} \mathcal{P}(I)$.

Since PRO is a power set lattice and $\cdot$ is isotone we can, by the Knaster-Tarski theorem, define the finite and infinite iterations A^* and A^ω of a process A as the fixpoints

$$A^* =_{df} \mu X\,.\, A \cdot X + I\,, \qquad A^\omega =_{df} \nu X\,.\, A \cdot X\,,$$

where for a function F, the expressions $\mu X\,.\, F(X)$ and $\nu X\,.\, F(X)$ denote the least and greatest fixpoint of F, resp.

We use this algebraic model 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.

3. Abstraction: Weak Kleene and Omega Algebras

Let us now have a closer look at the algebraic structure of the algebra of hybrid systems presented in the previous section.

Definition 3.1.

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 $\cdot$ distributives 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 implies immediately $\leq$-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 [20], one might also call a weak quantale a *weak standard Kleene algebra*.

Checking all the axioms for the case of processes, we get

Lemma 3.2.

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 be perform nothing. The other extreme* $\top$ *contains all possible trajectories, i.e., it can perform anything.*

2. *Additionally,* $\cdot$ *is positively disjunctive in its right argument.*

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

Another example are *guarded strings* (e.g., [23, 36]), 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 Σ is a finite sequence of zero or more elements from Σ; an *infinite word* is an infinite sequence. The empty word — the unique sequence of length 0 — is denoted by ϵ and *concatenation* of words v and w by $v.w$ if v in finite and v otherwise. The set of all finite words over Σ is denoted by Σ^*, the set of all infinite words by Σ^ω, the first element of a non-empty word v by $\mathsf{first}(v)$ and the last element of a finite word w by $\mathsf{last}(w)$.

Definition 3.3. A *guarded string* over the sets P of *states* and Σ of *transitions* is a non-empty word v such that $\mathsf{first}(v) \in P$ and in which elements from P and Σ alternate. Moreover, if the word is finite, $\mathsf{last}(v) \in P$. The product of guarded strings ρ_0 and ρ_1 is the guarded string

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

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

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 [30].

Lemma 3.4. *The powerset algebra* $\mathsf{GUA}(P, A) =_{df} (\mathcal{P}(\mathsf{GS}(P, \Sigma)), \cup, \emptyset, \bowtie, P)$ *over two alphabets P and* Σ*, with multiplication defined by*

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

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

Now we turn to an algebraic characterisation of iteration.

Definition 3.5.

1. A *weak Kleene algebra* [43] 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. A *weak omega algebra* [19, 42] 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 [42]). This is the formal reflection of the phenomena concerning infinite iteration of subidentities (elements less or equal than 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 3.6 ([42]).

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 construction 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

Lemma 3.7. *Assume a weak omega algebra. Then omega is isotone, i.e., $a \leq b \;\Rightarrow\; a^\omega \leq b^\omega$ and the following omega-regular laws hold:*

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

In Section 2 we have already introduced the purely finite and purely infinite parts of a process. A general algebraic treatment of these notions can be given using their behaviour under composition. Def. (1) entails, for process $A \in \mathsf{PRO}$, that $A \cdot \emptyset = \mathsf{inf}\, A$. 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, if its purely infinite part is trivial, that is, iff $\mathsf{inf}\, B = \emptyset$.

Definition 3.8. 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 $a \leq \mathsf{N} \;\Leftrightarrow\; a \cdot 0 = a$. 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 $a \leq \mathsf{F} \;\Leftrightarrow\; a \cdot 0 = 0$. In PRO, $\mathsf{F} = \{(d, g) : d < \infty\}$ consists of all trajectories of finite length.

By neutrality of 1 and isotony, all elements ≤ 1 are purely finite. Moreover, it is easy to check that the sets of purely finite and purely infinite elements are each closed under $+$ and $\cdot$; in a weak Kleene algebra the set of purely finite elements is also closed under *.

The definition of N implies, for all a,

$$\mathsf{N} \cdot a \leq \mathsf{N} \qquad \text{and} \qquad a \cdot \mathsf{N} \leq \mathsf{N}\,. \tag{5}$$

The definition of F implies

$$\mathsf{F} \cdot \mathsf{F} = \mathsf{F}\,. \tag{6}$$

In Boolean weak quantales 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 3.9. 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 an 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 [42].

The above equations imply

$$(\mathsf{inf}\, a) \sqcap b = \mathsf{inf}\,(a \sqcap b)\,, \qquad (\mathsf{fin}\, a) \sqcap b = \mathsf{fin}\,(a \sqcap b)\,. \tag{7}$$

The purely finite and purely infinite parts of a composition satisfy

$$a \cdot b = \mathsf{inf}\, a + \mathsf{fin}\, a \cdot b\,, \tag{8}$$

$$\mathsf{inf}\,(a \cdot b) = \mathsf{inf}\, a + \mathsf{fin}\, a \cdot \mathsf{inf}\, b\,, \tag{9}$$

$$\mathsf{fin}\,(a \cdot b) = \mathsf{fin}\,(\mathsf{fin}\, a \cdot b) = \mathsf{fin}\, a \cdot \mathsf{fin}\, b\,. \tag{10}$$

We now state further laws about purely finite and purely infinite parts.

Lemma 3.10. *Let S be a separated weak semiring with greatest element $\top$ and $a, b, c, d \in S$.*

1. $a \leq \mathsf{F} \Leftrightarrow a = \mathsf{fin}\, a \Leftrightarrow \mathsf{inf}\, a = 0$ *and*
   $a \leq \mathsf{N} \Leftrightarrow a = \mathsf{inf}\, a \Leftrightarrow \mathsf{fin}\, a = 0$.
2. *If a is purely finite then $a \cdot b = b$ iff $a \cdot \mathsf{fin}\, b = \mathsf{fin}\, b$ and $a \cdot \mathsf{inf}\, b = \mathsf{inf}\, b$.*

Assume now that S is a separated weak omega algebra.

3. $a^\omega = (\mathsf{fin}\, a)^* \cdot \mathsf{inf}\, a + (\mathsf{fin}\, a)^\omega$,
4. $\mathsf{inf}\, a^\omega = (\mathsf{fin}\, a)^* \cdot \mathsf{inf}\, a + \mathsf{inf}\,((\mathsf{fin}\, a)^\omega)$,
5. $\mathsf{fin}\, a^\omega = \mathsf{fin}\,((\mathsf{fin}\, a)^\omega) \leq (\mathsf{fin}\, a)^\omega$.

The proofs are straightforward or can be found in [42].

Part (1) gives equivalent characterisations of purely finite and purely infinite elements which are calculationally useful in various circumstances. Part 2 say that, for a purely finite element a, another element b is a fixpoint iff both its purely finite and purely infinite parts are. If a is a process, Part (3) says that infinite iteration of trajectories from a can take two forms: it may proceed a while with finite trajectories, but then add an infinite trajectory which prohibits further iteration — or it keeps iterating finite trajectories forever.

Part (5) 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. Part (4) 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. In the next section we will look at Zeno effects in detail.

4. Zeno Effects

Zeno of Elea's famous paradox about Achilles and the tortoise is well known. Obviously those effects may occur in PRO. We illustrate Zeno effects by a bouncing ball.

Example 4.1. The bouncing ball is one of the standard examples in the literature on hybrid systems. 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 an existing trajectory (restricted to x_1) is given in Figure 2. For simplicity we omit the formal definition of the involved hybrid automaton and hybrid automata in general. For the purpose of this paper the intuition of what a hybrid automaton is, should be sufficient. 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 is 0. If the ball is above the ground ($x_1 > 0$), its flow is governed by $\dot{x}_1 = x_2$, $\dot{x}_2 = -g$, where g is an arbitrary positive gravity force. 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

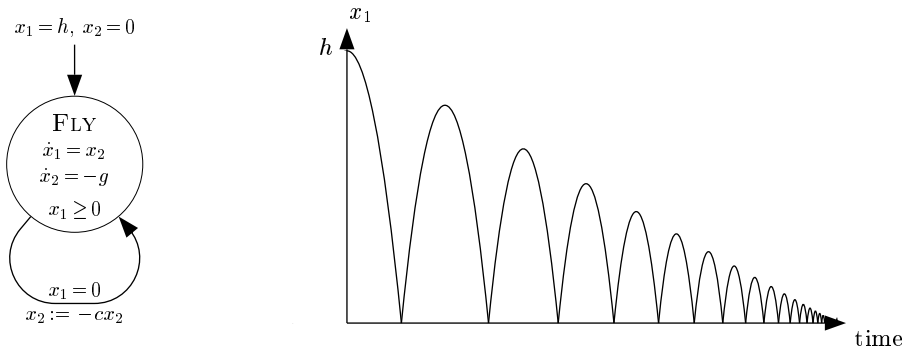

Figure 2: Hybrid automaton of a bouncing ball and corresponding trajectory w.r.t. x_1

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 (cf. right part of Figure 2). To model the bouncing ball algebraically in PRO, we set $D = \mathbf{R}_{\geq 0}$ and $V = \mathbf{R}^2$, and define a process

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

Let now $\circ$ stand for some iteration operator, e.g., ω or $\dagger$ as introduced in the following sections. Then the whole system should be characterised by

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

where $Y =_{df} \{(0, g) \,|\, g(0) = h\}$ models the initialisation and the process $\hat{Z}$ extends all trajectories in Z at both ends suitably to enforce the changes in direction when the ball touches the ground. For details concerning the extension, we refer to [26]. In this paper we will focus on these iteration operators and skip other details. □

In the area of hybrid systems, only a few authors treat Zeno effects (e.g. [29, 3]). Most of them do not treat Zeno effects within hybrid systems in detail, even if they appear in their theoretical models. For example, in [46] the authors avoid Zeno effects for the bouncing ball by changing the setting and making the damping factor a variable which can change between each jump. In this section we present a possibility of handling Zeno effects in PRO and characterise the Zeno and Zeno-free parts of hybrid systems.

To speak about Zeno effects we can use the purely finite and purely infinite parts of processes.

Lemma 4.2. *In a weak omega algebra, $a^\omega = a$ if a is purely infinite.*

For an arbitrary process infinite iteration can be determined by the general decomposition law $a^\omega = (\mathsf{fin}\, a)^* \cdot \mathsf{inf}\, a + (\mathsf{fin}\, a)^\omega$ (see Lemma 3.10(3)). Therefore it suffices to determine a^ω for purely finite elements a.

Example 4.3. One might expect that $Y \cdot (Z_{\asymp})^\omega$ describes the trajectories of Figure 2. As we will show in the next two sections, this is not the case, since, as mentioned in the introduction, Z^ω is too loose. □

5. Embedding Processes into Guarded Strings

To analyse Zeno phenomena, it is useful to get a more detailed view of A^ω for a process $A \in \mathsf{PRO}$. To do so, we need to speak in algebraic terms about prefixes of trajectories. For that we embed PRO homomorphically into the algebra GUA of guarded strings and reduce the behaviour of omega to the well known one in GUA. (It is easy to see that all guarded strings of an element T^ω have infinite length if $T \in \mathsf{GUA}(P, \Sigma)$.) After that we use a projection to go back to the algebra of hybrid systems.

In Section 3, we introduced the weak semiring $\mathsf{GUA}(P, \Sigma)$ of guarded strings over two alphabets P and Σ. Here we specialise Σ to the set $\mathsf{fin}(\mathsf{TRA})$ of *finite* trajectories and P to $\mathsf{test}(\mathsf{PRO})$, the elements of which are sets of zero-length trajectories, or isomorphically, sets of values. To define an embedding of purely finite processes into the weak

semiring $\mathsf{FGUA}(\mathsf{test}(\mathsf{PRO}), \mathsf{fin}(\mathsf{TRA}))$ we define a function ι that maps a trajectory $\tau = (d, g)$ with finite duration d to a guarded string of length 3:

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

Here $v \,.\, w$ denotes concatenation of v and w, as described before. This construction makes the initial and the final value of τ explicit. A zero-duration trajectory $\underline{x}$ is also mapped to a guarded string of length 3, namely $\underline{x} \,.\, \underline{x} \,.\, \underline{x}$. Again, we lift ι pointwise to a function $\iota : \mathsf{fin}(\mathsf{TRA}) \to \mathsf{GUA}(\mathsf{test}(\mathsf{PRO}), \mathsf{fin}(\mathsf{TRA}))$. In particular, $\iota(\mathsf{test}(\mathsf{PRO})) = \{\underline{x} \,.\, \underline{x} \,.\, \underline{x} \,|\, x \in V\} \neq \mathsf{test}(\mathsf{PRO})$ (considered as a set of guarded strings of length one). Due to this, ι is not a homomorphism. The above definition preserves the composition condition, i.e., $\tau_1 \cdot \tau_2$ is defined if and only if $\iota(\tau_1) \bowtie \iota(\tau_2)$ is defined.

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

Corollary 5.1. *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 $\mathsf{test}(\mathsf{PRO})$ and $\mathsf{fin}(\mathsf{TRA})$ in $\mathsf{GUA}(\mathsf{test}(\mathsf{PRO}), \mathsf{fin}(\mathsf{TRA}))$. 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 $(\mathsf{test}(\mathsf{PRO}) \,.\, \mathsf{fin}(\mathsf{TRA}))^* \,.\, \mathsf{test}(\mathsf{PRO})$ to TRA is inductively defined by

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

where $x \in V$ and $\tau \in \mathsf{TRA}$. Here τ may also be a test. By this definition we immediately get $\phi(u \bowtie w) = \phi(u) \cdot \phi(w)$. Lifting ϕ pointwise to sets of guarded strings yields the following result.

Lemma 5.2. *$\phi : \mathsf{FGUA}(\mathsf{test}(\mathsf{PRO}), \mathsf{TRA}) \mapsto \mathsf{PRO}$ is a weak-Kleene-algebra homomorphism, i.e., $\phi(0) = 0$, $\phi(1) = 1$, $\phi(a \cup b) = \phi(a) \cup \phi(b)$, $\phi(a \bowtie b) = \phi(a) \cdot \phi(b)$ and $\phi(a^*) = \phi(a)^*$. Moreover by pointwise lifting, ϕ is also disjunctive.*

Proof. Except the equation for finite iteration all calculations are straightforward and follow either from the definitions or from pointwise lifting. The last equation is shown by fixpoint fusion (cf. [1]). We choose $f(x) = \phi(a) \cdot x + \phi(1)$, $g(x) = \phi(x)$ and $h(x) = a \cdot x + 1$. By definition all these functions are isotone and g is continuous. Moreover we have

$$g(h(x)) \;=\; g(a \cdot x + 1) \;=\; \phi(a \cdot x + 1) \;=\; \phi(a) \cdot \phi(x) + \phi(1) \;=\; f(\phi(x)) \;=\; f(g(x)) \,.$$

The third step follows by additivity and multiplicativity of ϕ. Hence by fixpoint fusion we have $g(\mu h) = \mu f$. In particular, we have for an element $a \in \mathsf{FGUA}(\mathsf{test}(\mathsf{PRO}), \mathsf{TRA})$

$$\phi(a^*) = g(\mu h) = \mu f = \phi(a)^* \,.$$

□

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

6. 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 of an infinite guarded string by the supremum of its finite prefixes, i.e., we calculate the "limit" of all prefixes. The prefix relation on guarded strings is defined as usual: $w_1 \in (P \,.\, \Sigma)^* \,.\, P$ is a finite prefix of w_2, written as $w_1 \sqsubseteq w_2$, iff there is a $u \in (P \,.\, \Sigma)^* \,.\, P \cup (P \,.\, \Sigma)^\omega$ such that $w_1 \bowtie u = w_2$. Infinite guarded strings are maximal with respect to this order. Moreover, in GUA each (infinite) guarded string w is the supremum of all its (finite) prefixes.

$$w \;=\; \sup\{u \,|\, u \sqsubseteq w\} \;=\; \sup\{u \,|\, u \sqsubseteq w, |u| < \infty\} \,. \tag{11}$$

If w has finite length the set of its prefixes is finite, hence w is even the maximum of that set. The homomorphism ϕ is $\sqsubseteq$-isotone, i.e.,

$$w_1 \sqsubseteq w_2 \;\Rightarrow\; \phi(w_1) \sqsubseteq \phi(w_2) \,. \tag{12}$$

More generally, we define

Definition 6.1. 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 6.2. *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.*

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

$$\mathsf{pre}(w) =_{df} \{\phi(u) \,|\, u \sqsubseteq w,\, |u| < \infty\} \,.$$

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 w of $(\iota(A))^\omega$ has infinite length (cf. Def. 3.3). Moreover, there are an infinite number of prefixes, i.e., $|\mathsf{pre}(w)| = \infty$. Infinite iteration then results by passing to some sort of "limit" of $\mathsf{pre}(w)$. Unfortunately, in contrast to Equation (11), the supremum of $\mathsf{pre}(w)$ need not exist in PRO. We illustrate this fact by the following example.

Example 6.4. Consider the process $A =_{df} \{(\frac{1}{n^2}, g) \,|\, g(x) = n^2 \cdot x + n,\, n \in \mathbf{N}\}$, where the time domain D and the value set V are equal to $\mathbf{R}_{\geq 0}$. By definition of the embedding $(\iota(A))^\omega$ only consists of one single element, namely

$$\underline{1} \,.\, (1, g) \,.\, \underline{2} \,.\, (\frac{1}{4}, g) \,.\, \underline{3} \,.\, (\frac{1}{9}, g) \ldots \,.$$

All finite prefixes of this infinite guarded string have the form $u = \underline{1} \,.\, (1, g) \ldots \underline{n}$ $(n \in \mathbf{N})$. By this $\phi(u)$ has duration $\sum_{i=1}^{n} \frac{1}{i^2}$. The supremum of these trajectories 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. When one tries to define a limit trajectory by completing the open interval $[0, d_\infty[$ to a closed one faces the problem how to define $g(d_\infty)$. Shortly we will define an extended supremum operator that solves the problem by allowing all possible values $v \in V$ at time d_∞. □

Theorem 6.5. *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. *Let X be expanded by H, i.e., assume $X \subseteq H(X)$. Then for every $\xi \in X$ there is a guarded string $w \in \mathsf{GUA}$ such that $\tau \sqsubseteq \xi$ for all $\tau \in \mathsf{pre}(w)$.*
2. *$A^\omega = \{\xi \in \mathsf{TRA} \,|\, \exists\, w \in \mathsf{inf\,GUA} : \forall\, \tau \in \mathsf{pre}(w) :\ \tau \sqsubseteq \xi\}$.*

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 w as the supremum of all these trajectories lifted to guarded strings, i.e, $\sup\{w \,|\, w \in \prod_{i=1}^{n} \iota(\tau_i),\, n \in \mathbf{N}\}$ and we are done.

2. As a preparation we set $\mathrm{OM}(A) =_{df} \{\xi \in \mathsf{TRA} \mid \exists\, w \in \inf \mathsf{GUA} : \forall\, \tau \in \mathsf{pre}(w) : \tau \sqsubseteq \xi\}$ and observe that finite trajectories τ are left cancellative w.r.t. composition, i.e., satisfy $\tau \cdot \rho = \tau \cdot \sigma \Rightarrow \rho = \sigma$, provided $\tau \cdot \rho$ and $\tau \cdot \sigma$ are defined. By omega unfold every guarded string $w \in (\iota(A))^\omega$ has a prefix $w_0 \in \iota(A)$ with $w_0 \sqsubseteq w$.

   Now we show that $\mathrm{OM}(A)$ is expanded by H. Consider an arbitrary $\xi \in \mathrm{OM}(A)$. By definition there is a $w \in \inf \mathsf{GUA}$ with $\tau \sqsubseteq \xi$ for all $\tau \in \mathsf{pre}(w)$. By this and the above remark we know that there is a $w_0 \sqsubseteq w$ with $\phi(w_0) \in \mathsf{pre}(w)$ and $\phi(w_0) \in \phi(\iota(A)) = A$. Then by finiteness of $\phi(w_0)$ and the above cancellation property, there is a unique τ_1 with $\xi = \phi(w_0) \cdot \tau_1$. Hence $\mathrm{OM}(A) \subseteq A \cdot \mathrm{OM}(A)$.

   Together with Part (1) this means that $\mathrm{OM}(A)$ is the greatest expanded element of H and hence its greatest fixpoint. Now the claim follows by Lemma 3.6(2) □

The fact that A^ω contains arbitrary extensions of infinite A-iterations also explains why the property $A^\omega = A^\omega \cdot \top$ is not completely unnatural: for arbitrary $B \in \mathsf{PRO}$ the process $B \cdot \top$ is the extension closure of B. Hence $A^\omega = A^\omega \cdot \top$ reflects the fact that, operationally, after a Zeno gap the behaviour doesn't matter, since the gap cannot be "crossed" anyway. For a discussion of these phenomena in the context of hybrid automata see [44].

Example 6.6. Applying this to our example of the bouncing ball (cf. Example 4.1), we see that $Y \cdot (Z_{\asymp})^\omega$ contain2 the trajectory of Figure 2. The process has an infinite number of trajectories. Their initial sections coincide with the trajectory of Figure 2, but after reaching the Zeno point some miraculous behaviour occurs. 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 Figure 2. One is given in Figure 3.

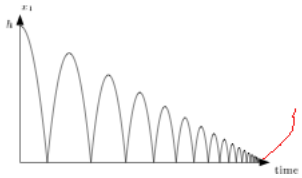

Figure 3: Another trajectory of Z^ω

□

Now, we generalise from PRO to a weak omega algebra S.

Definition 6.7. An element a of a weak omega algebra is called *divergent* or *Zeno-free*, if $a^\omega \leq \mathsf{N}$. It is called *Zeno* if it is not Zeno-free and it is called *convergent* if $a^\omega \leq \mathsf{F}$.

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

Lemma 6.8. *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 6.9. *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 a test and hence $(a \sqcap 1)^\omega = (a \sqcap 1) \cdot \top$. Now, by isotony and neutrality of 1,

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

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

7. A More Precise Iteration Operator

As we have seen, 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 special time domains.

To describe it, we define a supremum-operator for $\mathsf{pre}(w)$ which equals the proper supremum, if possible, and otherwise completes the open interval of durations involved to a closed one. This is done passing from a single trajectory to a whole set, namely all the ones who agree on the open interval and add an arbitrary value at the limit time. However the definition works only for special time domains. Let again A be purely finite and assume that the time domain D is *complete*, i.e., contains suprema for all its subsets. We set $d_w =_{df} \sup\{d \,|\, (d,g) \in \mathsf{pre}(w)\}$.

Definition 7.1. For a set of trajectory-prefixes $\mathsf{pre}(w) = \{\phi(u) \,|\, u \sqsubseteq w, |u| < \infty\}$, we define the *extended supremum* $\widehat{\sup} : \mathsf{PRO} \to \mathsf{PRO}$ by

$$\widehat{\sup}(\mathsf{pre}(w)) =_{df} \begin{cases} \{\sup(\mathsf{pre}(w))\} & \text{if } d_w = \infty \\ \{(d_w, g)\} & \text{if } (d_w, g) \in \mathsf{pre}(w) \\ \{(d_w, \hat{g}) \,|\, \hat{g}(d_w) = v,\ v \in V, & \\ \qquad \exists\, (d,g) \in \mathsf{pre}(w) : & \\ \qquad \hat{g}(t) = g(t) \text{ if } t \le d_w\} & \text{otherwise .} \end{cases}$$

If $d_w = \infty$, the limit of the set of prefixes do not show a Zeno effect and the result is a singleton process consisting just of one infinite trajectory. For $d_w \neq \infty$, two cases arise. The first case can only happen when the sequence of prefixes becomes stationary with infinitely many trajectories of duration zero and identical value v at the end. This means the special kind of Zeno behaviour of idling forever. The second case, where $d_w \neq \{d \,|\, (d,g) \in \mathsf{pre}(w)\}$, i.e., $d_w > d$ for all trajectories $(d,g) \in \mathsf{pre}(w)$, means proper Zeno behaviour where the trajectories become longer and longer without ever reaching the "limit time" d_w.

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

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

that can again be lifted pointwise to sets of guarded strings. Unfortunately, ϕ is not a homomorphism any longer, since in general $\phi(u \bowtie w) \neq \phi(u) \cdot \phi(w)$ if u has infinite length. However, ϕ still commutes with multiplication and $\phi(u \bowtie w) = \phi(u) \cdot \phi(w)$ if u is finite (w might be infinite).

Corollary 7.2. *If A is a purely finite process then $A^\omega = \phi((\iota(A))^\omega) \cdot \top$.*

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

Definition 7.3. 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 3.10(3)).[1]

The whole construction of † is summarised in the diagram of Figure 4.

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 6.5. Since the definition is based on omega iteration on GUA and projection ϕ we get for an arbitrary set of guarded strings $L \in \mathsf{GUA}(\mathsf{test}(\mathsf{PRO}), \mathsf{fin}\,(\mathsf{TRA}))$

$$\phi(L^\omega) = (\phi(L))^\dagger \tag{13}$$

if $L \cap \mathsf{test}(\mathsf{PRO}) = \emptyset$. Moreover, from Def. 7.3 we get immediately

[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]).

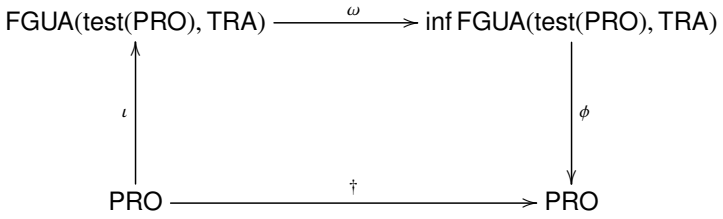

Figure 4: Construction of †

Corollary 7.4. *Infinite iteration of a zero-duration process is stationary, that is $P^\dagger = P$. In particular we have $I^\dagger = I$ for the multiplicative identity I of* PRO*, whereas $I^\omega = \top =$ TRA.*

Theorem 7.5. *Let H be as in Theorem 6.5.*

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 $A^\dagger$.*
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 observe again 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 6.5 there is a set $\mathsf{pre}(w)$ with $\tau \sqsubseteq \xi$ for all $\tau \in \mathsf{pre}(w)$. By definition of $\widehat{\sup}(\mathsf{pre}(w))$ we have for all $\sigma \in \widehat{\sup}(\mathsf{pre}(w))$ and all $\tau \in \mathsf{pre}(w)$ $\tau \sqsubseteq \sigma$ If $\widehat{\sup}(\mathsf{pre}(w))$ contains only one single trajectory (no proper Zeno effect occurs) $\sigma_0 =_{df} \widehat{\sup}(\mathsf{pre}(w)) \sqsubseteq \xi$. In the case of Zeno effects there is a trajectory $\sigma_0 \in \widehat{\sup}(\mathsf{pre}(w))$ with $\sigma_0 \sqsubseteq \xi$. σ is the "limit" of all $\tau \in \mathsf{pre}(w)$ that coincides with ξ at time d; hence $\sigma \sqsubseteq \xi$. Since $\sigma_0 \in \widehat{\sup}(\mathsf{pre}(w)) \subseteq A^\dagger$, we are done.
3. The claim directly follows from Corollary 7.2 and definition of dagger. □

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

Lemma 7.6. *For an arbitrary process A*

$$A^\dagger \leq \mathsf{N} \;\Leftrightarrow\; A^\omega \leq \mathsf{N} \;\Rightarrow\; A^\dagger = A^\omega \,.$$

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

8. 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 need the following definition and lemma, motivated by Theorem 7.5.

Definition 8.1. 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 $x \leq c\top$ for all fixpoints of a. A *spanning fixpoint of a* is a fixpoint of a that is also spanning for a.

Lemma 8.2. *Let x be a fixpoint of a in a weak Kleene algebra. Then x is a fixpoint of a^* as well.*

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

$$a^* \cdot x \leq x \Leftarrow x + a \cdot x \leq x \Leftrightarrow a \cdot x \leq x\,.$$

□

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 8.3. 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

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

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

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

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

(c) ϕ is inverse to ι, i.e., for all $a \in \mathsf{fin}(S)$, we have $\phi(\iota(a)) = a$.

(d) ϕ projects omega to dagger, i.e., if $x \in \mathsf{fin}\, T$ then $\phi(x^\omega) = (\phi(x))^\dagger$.

(e) For all $a, b \in S$ the element $a^\dagger$ is *spanning for* a, b, i.e., for all $c \in S$ with $c \leq a \cdot c + b$ we have $c \leq a^\dagger \cdot \top + a^* \cdot b$. 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.

(f) For all subidentities $p \leq 1$ ($p \in S$), we have $p^\dagger = p$. In particular $1^\dagger = 1$.

From Parts (a) and (b) we get immediately that ι and ϕ are isotone. Moreover, by Part (c) 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 (d) is the abstract counterpart of Equation (13).

Definition 8.4. 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 that defines † 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 [38, 39], since there is only one maximal fixpoint, namely the greatest fixpoint a^ω.

Based on Theorem 7.5 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 8.5. A Boolean weak semiring has the *progress property* if $\overline{1} \cdot \overline{1} \leq \overline{1}$.

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 cannot be undone. For instance, PRO and GUA have the progress property. By Boolean algebra, the progress property is equivalent to $1 \leq \overline{\overline{1} \cdot \overline{1}}$. The element $\overline{\overline{1} \cdot \overline{1}}$ has been called step in [50]; it represent elements that cannot be split into non-subidentities.

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

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

Moreover, the progress property is equivalent to

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

for all $p \leq 1$ and arbitrary a, b. For the proofs see [27]. 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 8.6. 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 8.2, $(a + 1)^*$ 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., $a \cdot c \leq c$. Since c is spanning for $a + 1$ we obtain $a^* \leq c \cdot \top$. Hence $1 \leq c \cdot \top$ and therefore, by the above decomposition property, $1 = 1 \sqcap 1 \leq c \cdot \top \sqcap 1 = (c \sqcap 1) \cdot (\top \sqcap 1) = c \sqcap 1$, which shows $1 \leq c$. Altogether we have $1 + a \cdot c \leq c$ and star induction shows $a^* \leq c$.

But consider now the concrete algebra PRO over the time domain $\mathbf{R}_{\geq 0} \cup \{\infty\}$ 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.

Lemma 8.7. 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 GUA(test(PRO), fin(TRA)). It is straightforward to check that ϕ and ι satisfy the required properties. Most of them have already been shown in the previous sections. □

Let us now draw some conclusions from the abstract definition. First, we state that ι behaves homomorphically under application of ϕ.

Lemma 8.8. *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. $\phi(\iota(a^+)) = \phi(\iota(a)^+)$.

Proof.

1. By Def. 8.3(c) twice and Def. 8.3(b),
$$\phi(\iota(a \cdot b)) = a \cdot b = \phi(\iota(a)) \cdot \phi(\iota(b)) = \phi(\iota(a) \odot \iota(b)).$$
2. The first equation is immediate from the definition. By Def. 8.3(d), Part 1 and Def. 8.3(d) 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 (c) twice and Def. 8.3(b),
$$\phi(\iota(a^*)) = a^* = \phi(\iota(a))^* = \phi((\iota(a)^*).$$

4. By the definition of $^+$, Part 1, Def. 8.3(b), Part 3, Def. 8.3(b) and the definition of $^+$ again,

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

□

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

Lemma 8.9. *Assume a dagger algebra S. Then, for $a, b \in \mathsf{fin}(S)$, $\iota(a \cdot b) = \iota(\phi(\iota(a) \odot \iota(b)))$.*

Proof. By Def. 8.3(b) we get $\iota(\phi(\iota(a) \odot \iota(b))) = \iota(\phi(\iota(a)) \cdot \phi(\iota(b))) = \iota(a \cdot b)$. □

Theorem 8.10. *The following properties hold in a dagger algebra.*

1. *Dagger is $\leq$-isotone, i.e., $a \leq b \Rightarrow a^\dagger \leq b^\dagger$.*
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. *$(a \cdot b)^\dagger \leq (a + b)^\dagger$.*
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 $p \leq 1$ then $(p + b)^\dagger = b^\dagger + b^* \cdot p$.*

Proof. 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

$$\begin{array}{ll} & (a^+)^\dagger \\ = & \quad \{\!\!\{ \text{ definition of dagger } \}\!\!\} \\ & \phi(\iota(a^+)^\omega) \\ = & \quad \{\!\!\{ \text{ by Def. 8.3(d) } \}\!\!\} \\ & (\phi(\iota(a^+)))^\dagger \\ = & \quad \{\!\!\{ \text{ by Lemma 8.8.4 } \}\!\!\} \\ & (\phi(\iota(a)^+))^\dagger \\ = & \quad \{\!\!\{ \text{ by Def. 8.3(d) } \}\!\!\} \\ & \phi((\iota(a)^+)^\omega) \\ = & \quad \{\!\!\{ \text{ weak omega algebra (Lemma 3.7) } \}\!\!\} \\ & \phi(\iota(a)^\omega) \\ = & \quad \{\!\!\{ \text{ definition of dagger } \}\!\!\} \\ & a^\dagger . \end{array}$$

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

6. We calculate

$$
\begin{array}{ll}
 & (a \cdot b)^\dagger \\
= & \quad \{\!\!\{ \text{ by Lemma 8.8.2 } \}\!\!\} \\
 & \phi(\iota(a) \odot \iota(b))^\omega) \\
= & \quad \{\!\!\{ \text{ weak omega algebra (Lemma 3.7) } \}\!\!\} \\
 & \phi(\iota(a) \odot (\iota(b) \odot \iota(a))^\omega) \\
= & \quad \{\!\!\{ \text{ by Def. 8.3(b) } \}\!\!\} \\
 & \phi(\iota(a)) \cdot \phi((\iota(b) \odot \iota(a))^\omega) \\
= & \quad \{\!\!\{ \text{ by Def. 8.3(b) and Lemma 8.8.1 } \}\!\!\} \\
 & a \cdot \phi((\iota(b \cdot a))^\omega) \\
= & \quad \{\!\!\{ \text{ definition of dagger } \}\!\!\} \\
 & a \cdot (b \cdot a)^\dagger .
\end{array}
$$

7. We calculate

$$
\begin{array}{ll}
 & (a+b)^\dagger \\
= & \quad \{\!\!\{ \text{ definition of dagger } \}\!\!\} \\
 & \phi((\iota(a+b))^\omega) \\
= & \quad \{\!\!\{ \text{ by Def. 8.3(b) } \}\!\!\} \\
 & \phi((\iota(a) \oplus \iota(b))^\omega) \\
= & \quad \{\!\!\{ \text{ weak omega algebra (Lemma 3.7) and setting } z =_{df} \iota(a)^* \odot \iota(b) \ \}\!\!\} \\
 & \phi(z^\omega \oplus z^* \odot \iota(a)^\omega) \\
= & \quad \{\!\!\{ \text{ by Def. 8.3(b) } \}\!\!\} \\
 & \phi(z^\omega) + \phi(z^* \odot \iota(a)^\omega) \\
= & \quad \{\!\!\{ \text{ by Def. 8.3(d) and Def. 8.3(b) } \}\!\!\} \\
 & \phi(z)^\dagger + \phi(z^*) \cdot \phi(\iota(a)^\omega) \\
= & \quad \{\!\!\{ \text{ by Lemma 8.8.3, Def. 8.3(d) and Def. 8.3(c) } \}\!\!\} \\
 & \phi(z)^\dagger + \phi(z)^* \cdot a^\dagger .
\end{array}
$$

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

$$
\begin{array}{ll}
 & \phi(\iota(a)^* \odot \iota(b)) \\
= & \quad \{\!\!\{ \text{ by (b) } \}\!\!\} \\
 & \phi(\iota(a)^*) \cdot \phi(\iota(b)) \\
= & \quad \{\!\!\{ \text{ by Lemma 8.8.3 and Def. 8.3(c) } \}\!\!\} \\
 & \phi(\iota(a))^* \cdot b \\
= & \quad \{\!\!\{ \text{ by (c) } \}\!\!\} \\
 & a^* \cdot b.
\end{array}
$$

   This concludes the calculation.

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

□

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

Corollary 8.11. *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 $c \leq a \cdot c + b$. Since $a^\dagger$ is spanning for a and b, we infer $c \leq a^\dagger \cdot \top + a^* \cdot b = a^\omega + a^* \cdot b$. □

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

Lemma 8.12. *Assume a Boolean dagger algebra.*

1. $\mathsf{N}^\dagger = \mathsf{N}$ *and* $\top^\dagger = \top$.
2. *Part (f) of Def. 8.3 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* $(a \sqcap \overline{1})^* \cdot (a \sqcap 1) \leq a^\dagger$.

Proof.

1. First, $\mathsf{N}^\dagger = \mathsf{N} \cdot \mathsf{N}^\dagger = \mathsf{N}$. Second, by the general definition of dagger, $\top^\dagger = \mathsf{F}^* \cdot \mathsf{N} + \mathsf{F}^\dagger = \mathsf{F}^* \cdot \mathsf{N} + \mathsf{F} \cdot \mathsf{F}^\dagger \geq \mathsf{N} + \mathsf{F} = \top$, since $\mathsf{F}^* \geq 1$ and $F^\dagger \geq 1^\dagger = 1$ by $\mathsf{F} \geq 1$ and isotony of †.
2. We recall that in a Boolean semiring all elements ≤ 1 are tests [37], for which, in particular, $\cdot$ and $\sqcap$ coincide.

   Now first, by isotony and the definition of dagger, $p \leq 1$ implies $p^\dagger \leq 1^\dagger = 1$. Since the algebra is Boolean this means that $p^\dagger$ is a test. Next, by Theorem 8.10(2) we have $p^\dagger = p \cdot p^\dagger$, which means $p^\dagger \leq p$.

   By assumption and the above, also p is a test. Therefore we have 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,

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

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

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

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

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

3. This follows from Theorem 8.10.7 by splitting $a = (a \sqcap 1) + (a \sqcap \overline{1})$ and the fact that $x^* = 1$ for $x \leq 1$.

□

9. Zeno Phenomena Algebraically

We now continue the algebraic discussion of Zeno phenomena we have started at the end of Section 6 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 8.11.

We first study the interplay between purely infinite spanning fixpoints.

Lemma 9.1.

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

Proof.

1. We have $d \leq c \cdot \top = c$.
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 $a^\dagger \leq a^\omega$. Hence $a^\omega \leq \mathsf{N}$ implies $a^\dagger \leq \mathsf{N}$. Moreover, both a^ω and $a^\dagger$ are spanning fixpoints of a, so that we can apply Part 2.

□

In Def. 6.7 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 9.2. 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 9.3. *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 9.1.3 yields $(a \sqcap \overline{1})^\dagger = (a \sqcap \overline{1})^\omega$. Now the claim is immediate from Theorem 8.10.8. □

In Def. 7.1, we have seen that the extended supremum, and hence 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 9.4. *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.*

Proof.

1. By Theorem 8.10(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

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

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

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

□

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

Corollary 9.5. *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 9.6. *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 fin /inf calculus shows the claim. □

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

10. 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., [28]). Moreover we have derived a number of useful properties. Most of them were used in [26], 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 by composing three different fixpoint of f. The first one should describe the idling part, the second one the real 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.

Acknowledgement. We are grateful to Han-Hing Dang for valuable comments.

References

[1] C. Aarts, R. Backhouse, E. Boiten, H. Doornbos, N. van Gasteren, R. van Geldrop, P. Hoogendijk, E. Voermans, and J. van der Woude. Fixed-point calculus. *Information Processing Letters*, 53(3):131–136, 1995.

[2] R. Alur, C. Courcoubetis, T. A. Henzinger, and 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, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *LNCS*, pages 209–229. Springer, 1993.

[3] A. D. Ames, A. Abate, and 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 and M. Nivat. The metric space of infinite trees. algebraic and topological properties. *Fundam. Inform.*, 3(4):445–476, 1980.

[6] J. C. M. Baeten and J. A. Bergstra. Process algebra with propositional signals. In *ACP '95: Algebra of Communicating Processes*, pages 381–405. Elsevier, 1997.

[7] J. C. M. Baeten and C. A. Middelburg. *Process Algebra with Timing*. Monographs in Theoretical Computer Science. Springer, 2002.

[8] J. C. M. Baeten and W. P. Weijland. *Process Algebra*. Cambridge University Press, 1990.

[9] J. Bergstra, I. Bethke, and A. Ponse. Process algebra with iteration and nesting. *Comput. J.*, 37(4):243–258, 1994.

[10] J. Bergstra and J. Klop. Fixed point semantics in process algebra. Technical Report IW 206/82, Centre for Mathematics and Computer Science, 1982.

[11] J. Bergstra and J. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37:77–121, 1985.

[12] J. Bergstra and A. Ponse. An instruction sequence semigroup with involutive anti-automorphisms. *CoRR*, abs/0903.1352, 2009.

[13] J. Bergstra and J. Tiuryn. Regular extensions of iterative algebras and metric interpretations. *Fundam. Inform.*, 4(4):997–1014, 1981.

[14] J. A. Bergstra and C. A. Middleburg. Process algebra for hybrid systems. *Theoretical Computer Science*, 335(2-3):215–280, 2005.

[15] S. L. Bloom and Z. Ésik. Equational axioms for regular sets. *Mathematical Structures in Computer Science*, 3(1):1–24, 1993.

[16] M. Broy and K. Stølen. *Specification and Development of Interactive Systems: Focus on streams, interfaces, and refinement*. Springer, 2001.

[17] F. Cardone and J. R. Hindley. *Lambda-calculus and Combinators in the 20th Century*, volume 5 of *Handbook of the History of Logic*, chapter 14. Elsevier, 2009.

[18] Y. Chen. A fixpoint theory for non-monotonic parallelism. *Theor. Comput. Sci.*, 308:367–392, 2003.

[19] E. Cohen. Separation and reduction. In R. Backhouse and J. N. Oliveira, editors, *Mathematics of Program Construction (MPC 2000)*, volume 1837 of *LNCS*, pages 45–59. Springer, 2000.

[20] J. H. Conway. *Regular Algebra and Finite Machines*. Chapman & Hall, 1971.

[21] W. Cook and J. Palsberg. A denotational semantics of inheritance and its correctness. *ACM SIGPLAN Notices*, 24(10):433–443, 1989.

[22] C. Elgot. The common algebraic structure of exit-automata and machines. *Computing*, 6:349–370.

[23] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.

[24] T. A. Henzinger. The theory of hybrid automata. In M. K. Inan and M. K. Kurshan, editors, *Verification of Digital and Hybrid Systems*, volume 170 of *NATO ASI Series F: Computer and Systems Sciences*, pages 265–292. Springer, 2000.

[25] P. Höfner. Automated reasoning for hybrid systems — Two case studies. In R. Berghammer, B. Möller, and G. Struth, editors, *Relations and Kleene Algebra in Computer Science*, volume 4988 of *LNCS*, pages 191–205. Springer, 2008.

[26] P. Höfner. *Algebraic Calculi for Hybrid Systems*. Books on Demand GmbH, 2009.

[27] P. Höfner and B. Möller. An algebra of hybrid systems. *Journal of Logic and Algebraic Programming*, 78:74–97, 2009.

[28] P. Höfner and G. Struth. Automated reasoning in Kleene algebra. In F. Pfennig, editor, *Automated Deduction*, volume 4603 of *LNAI*, pages 279–294. Springer, 2007.

[29] K. H. Johansson, M. Egerstedt, J. Lygeros, and S. S. On the regularization of Zeno hybrid automata. *Systems & Control Letters*, 38:141–150, 1999.

[30] D. M. Kaplan. Regular expressions and the equivalence of programs. *NATO ASI Series F: Computer and Systems Sciences*, 3(4):361–386, 1969.

[31] S. C. Kleene. Representation of events in nerve nets and finite automata. Technical Report RM-704, RAND Corporation, 1951. RAND Research Memorandum.

[32] S. C. Kleene. *Introduction to metamathematics*. Van Nostrand, 1952.

[33] B. Knaster. Un thèorëme sur les fonctions d'ensembles. *Ann. Soc. Polon. Math.*, (6):133–134, 1928.

[34] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994.

[35] D. Kozen. Kleene algebra with tests. *ACM Trans. Prog. Languages and Systems*, 19(3):427–443, 1997.

[36] D. Kozen. Automata on guarded strings and applications. *Matématica Contemporânea*, 24:117–139, 2003.

[37] E. Manes and D. Benson. The inverse semigroup of a sum-ordered semiring. *Semigroup Forum*, 31:129–152, 1985.

[38] Z. Manna and A. Shamir. The optimal fixedpoint of recursive programs. In *STOC '75: Proceedings of seventh annual ACM symposium on Theory of computing*, pages 194–206. ACM Press, 1975.

[39] Z. Manna and A. Shamir. The theoretical aspects of the optimal fixed point. *SIAM Journal on Computing*, 5(3):414–426, 1976.

[40] G. Markowsky and 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*, pages 34–47. IEEE, 1975.

[41] T. J. Marlowe and B. G. Ryder. Properties of data flow frameworks: A unified model. *Acta Informatica*, 28(2):121–163, 1990.

[42] B. Möller. Kleene getting lazy. *Science of Computer Programming*, 65:195–214, 2007.

[43] B. Möller and G. Struth. WP is WLP. In W. MacCaull, M. Winter, and I. Düntsch, editors, *Relational Methods in Computer Science*, volume 3929 of *Lecture Notes in Computer Science*, pages 200–211. Springer, 2006.

[44] K. Nakamura and A. Fusaoka. On transfinite hybrid automata. In M. Thiele and L. Thiele, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *LNCS*, pages 495–510. Springer, 2005.

[45] D. Park. On the semantics of fair parallelism. In D. Bjørner, editor, *Proceedings of the Abstract Software Specifications, 1979 Copenhagen Winter School*, LNCS, pages 504–526. Springer, 1980.

[46] A. Platzer and J.-D. Quesel. KeYmaera: A hybrid theorem prover for hybrid systems. In A. Armando, P. Baumgartner, and G. Dowek, editors, *Automated Reasoning*, volume 5195 of *LNAI*, pages 171–178. Springer, 2008.

[47] Z. Qian. Standard fixpoint iteration for java bytecode verification. *ACM Trans. Prog. Languages and Systems*, 22(4):638–672, 2000.

[48] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 2(5):285–309, 1955.

[49] J. Tiuryn. Unique fixed points vs. least fixed points. *Theoretical Computer Science*, 12:229–254, 1980.

[50] B. von Karger. Temporal algebra. *Mathematical Structures in Computer Science*, 8(3):277–320, 1998.