

Beyond Zeno-behaviour

Citation for published version (APA):

Cuijpers, P. J. L., Reniers, M. A., & Engels, A. G. (2001). *Beyond Zeno-behaviour*. (Computer science reports; Vol. 0104). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2001

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

Beyond Zeno-behaviour

by

P.J.L. Cuijpers, M.A. Reniers, A.G. Engels

01/04

ISSN 0926-4515

All rights reserved

editors: prof.dr. J.C.M. Baeten
prof.dr. P.A.J. Hilbers

Reports are available at:
<http://www.win.tue.nl/win/cs>

Beyond Zeno-behaviour

P.J.L. Cuijpers, M.A. Reniers, A.G. Engels

Department of Mathematics and Computer Science,

Division of Computer Science,

Eindhoven University of Technology,

P.O. Box 513, 5600 MB Eindhoven, The Netherlands,

{P.J.L.Cuijpers, M.A.Reniers, A.G.Engels}@tue.nl

Abstract

When modelling and analysing hybrid systems using techniques from computing science we may encounter problems with so-called Zeno-behaviour. This is the phenomenon that an infinite number of events accumulates before a finite time (Zeno-time). When this happens the standard techniques from computing science fail to distinguish between events that happen after that sequence of events. Many of those techniques have a semantics based on labelled transition systems.

In this article, we concentrate on those transition systems and try to find a solution for the Zeno-problem. We first introduce transitions over infinite sequences, since an infinite number of events needs to be described. Then we (re-)define a notion of convergence over sequences in a metric space. Considering a transition system with a metric state space and transitions labelled by sequences we can define a notion of prefix- and accumulation-closedness. Finally within prefix- and accumulation-closed transition systems, bisimilarity turns out to distinguish between various kinds of transfinite behaviour. The bouncing ball, an example from hybrid system theory, is used to illustrate the relevance of these new definitions.

1 Introduction

A system containing both discrete and continuous behaviour is called a hybrid system. The analysis of such systems relies on techniques from computing science as well as from system theory. The combination of those two fields leads to new, theoretic and practical, challenges, one of which is called Zeno-behaviour.

When modelling and analysing hybrid systems using techniques from computing science we may encounter the phenomenon that an infinite sequence of events happen before a certain time instance [9, 1]. This behaviour is called Zeno-behaviour, after the Eleatic philosopher (488 BC) who first described such a phenomena in the famous example of a race between Achilles and a turtle. The time instance at which the events accumulate is called Zeno-time. In such a case, the standard formalisms used in computing science fail to distinguish between events that happen after that interval. Johansson et al. [7] performed a regularization of Zeno Hybrid Automata in 1998. Even more recently Bérard and Picaconny [1] adressed the problem of Zeno-words in timed automata using the notion of acceptance of transfinite words that was introduced by Büchi [5] in 1973. Using a notion of limit of an infinite sequence they were able to solve the problem of Zeno-behaviour concerning trace-equivalence of automata. In this paper the context is broadened to transition systems, which form the general underlying model for many of the formalisms used in computing science (including automata), while we adress the problem of Zeno-behaviour regarding the notion of (strong-)bisimilarity, which is stronger than trace-equivalence.

Limits of sequences are strongly related to metric spaces. Combining this insight with the transition systems that we are studying, we arrive at a notion of transition systems that take a metric space as their state space. Previously, metric transition systems were studied by Kent [8] in the construction of a closure powerspace, and van Breugel [3] to give semantics to programming languages. Here we use them, extended with transitions over transfinite sequences to tackle the Zeno-problem.

We start with an introduction on Zeno-behaviour by studying a typical example of a dynamical system containing such behaviour: the bouncing ball. It is described using an ordinary transition system and we show that bisimilarity is not capable of distinguishing between differences in behaviour after Zeno-time. In the next section, transition systems are discussed in which the labels may consist of infinite sequences of labels. We define a notion of

prefix-closedness of a transition system and show why this notion is useful. In the fourth section we propose metric spaces to be used as the set of states of a transition system. This gives us the chance to formalize a notion of convergence. Finally we redo our bouncing ball example in the new context and show that bisimilarity now does distinguish between different transfinite behaviours.

2 Zeno-behaviour

Consider a ball that is bouncing on a surface like depicted in figure 1. With every bounce it loses a factor ξ^2 of its energy ($\xi \in (0, 1)$). For simplicity we assume that we start observing the system at a time t_0 when the ball leaves the surface with a velocity v_0 . Suppose that after the n^{th} bounce at time t_n this ball hits the surface with a velocity v_n .

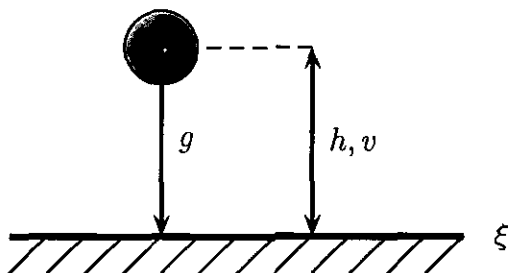


Figure 1: Bouncing Ball

Then it bounces off with a velocity $-\xi v_n$. Under the pull of a gravitational force g it touches the surface again after a time interval $t_{n+1} - t_n = \frac{2\xi}{g} v_n$ with the same speed as with which it took off (in opposite direction, $v_{n+1} = \xi v_n$). Summarizing we find that the system can be described by:

$$v_{n+1} = \xi v_n, \tag{1}$$

$$t_{n+1} = t_n + \frac{2\xi}{g} v_n. \tag{2}$$

Using development of series we find:

$$\begin{aligned}
 v_n &= \xi^n v_0, \\
 t_n &= t_0 + \frac{2v_0}{g} \sum_{i=1}^n \xi^i \\
 &= t_0 + \frac{2v_0}{g} \left(\frac{\xi - \xi^{n+1}}{1 - \xi} \right).
 \end{aligned}$$

This process contains Zeno-behaviour since it takes an infinite sequence of events (bounces) to put the ball to rest while all these events occur before a certain time (Zeno-time):

$$\begin{aligned}
 v_{Zeno} &= \lim_{n \rightarrow \infty} v_n = 0, \\
 t_{Zeno} &= \lim_{n \rightarrow \infty} t_n = t_0 + \frac{2v_0}{g} \left(\frac{\xi}{1 - \xi} \right).
 \end{aligned}$$

Before we go on, we need a proper definition of what is known in computing science as a *labelled transition system* or LTS. Many techniques from computing science have a semantics based on some kind of LTS. A common definition is given below.

Definition 2.1 (LTS) *A labelled transition system is a tuple $\langle X, \Sigma, E, I \rangle$ where X is the set of states, Σ is some alphabet (set of labels), $E \subseteq X \times \Sigma \times X$ is the set of transitions and $I \subseteq X$ is the set of initial states.*

When looking at dynamical systems, the states in X define the state of the system and the labels in Σ are interpreted as actions that can be taken. The triple $(x, \sigma, x') \in E$ then signifies that if the system is in a state x it can get into state x' by performing some action σ . This is also denoted as $x \xrightarrow{\sigma} x'$.

Now, let us reconsider the bouncing ball. The state of this system is determined by the space \mathbb{V} from which the velocity takes its value and the space \mathbb{T} from which time takes its value. We may describe the bouncing ball by an LTS $B_1 = \langle X_1, \Sigma_1, E_1, I_1 \rangle$ such that:

$$\begin{aligned}
X_1 &= \mathbb{V} \times \mathbb{T}, \\
\Sigma_1 &= \{bounce\}, \\
E_1 &= \left\{ ((v_1, t_1), bounce, (v_2, t_2)) \mid v_1 \neq 0, v_2 = \xi v_1, t_2 = t_1 + \frac{2\xi}{g} v_1 \right\}, \\
I_1 &= \{(i_v, i_t)\}.
\end{aligned}$$

Here, X_1 denotes the space spanned by all the velocity-time pairs (v, t) . The only label $bounce \in \Sigma_1$ of the system denotes a bounce of the ball. The set I_1 of initial states contains only the starting point (i_v, i_t) . In the definition of the transitions E_1 we recognize the equations (1) and (2) in the beginning of this section. The description may be interpreted as:

“Starting from the state (i_v, i_t) the system can get into a new state $(\xi i_v, i_t + \frac{2\xi}{g} i_v)$ by performing a bounce, being in this new state a bounce will bring it to the state $(\xi\xi i_v, i_t + \frac{2i_v}{g}(\xi + \xi\xi))$, etc.”

Figure 2 contains a visualization of this labelled transition system. States are indicated by circles, transitions by arrows between circles, and initial states by double circles.

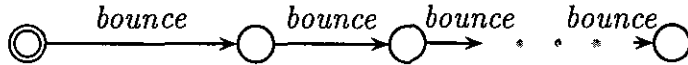


Figure 2: Labelled Transition System for B_1

Let us also consider a second system $B_2 = \langle X_2, \Sigma_2, E_2, I_2 \rangle$ which is the same as the first except for the additional edges $(0, t) \xrightarrow{kick} (v_b, t)$ for all t :

$$\begin{aligned}
X_2 &= \mathbb{V} \times \mathbb{T}, \\
\Sigma_2 &= \{bounce, kick\}, \\
E_2 &= \left\{ ((v_1, t_1), bounce, (v_2, t_2)) \mid v_1 \neq 0, v_2 = \xi v_1, t_2 = t_1 + \frac{2\xi}{g} v_1 \right\} \cup \\
&\quad \left\{ ((v_1, t_1), kick, (v_2, t_2)) \mid v_1 = 0, v_2 = v_b, t_1 = t_2 \right\}, \\
I_2 &= \{(i_v, i_t)\}.
\end{aligned}$$

The interpretation of this system might be that after the ball has come to rest, it is kicked to a speed v_b and starts bouncing again (see figure 3). It seems

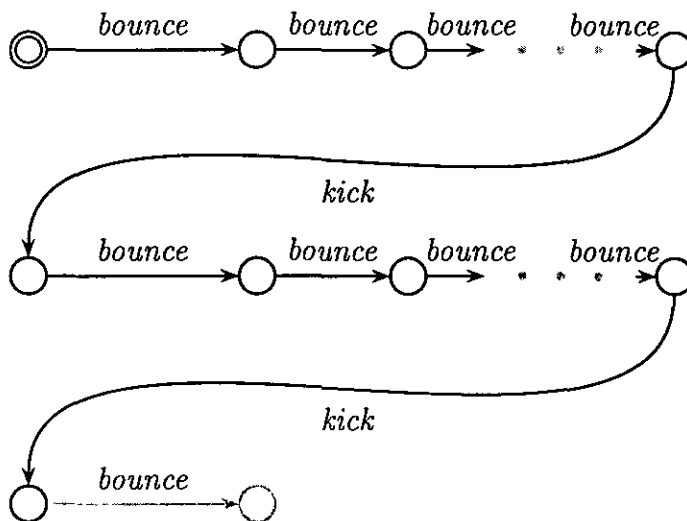


Figure 3: Labelled Transition System for B_2

clear that there is a difference between these two systems. However, even one of the strongest notions of equivalence that is often used in computing science (strong-bisimilarity), generally does not make a distinction between B_1 and B_2 . (Isomorphism does make the distinction but is considered impractical for many purposes.)

Definition 2.2 (strong-bisimilarity) *Two LTSs $S_1 = \langle X_1, \Sigma_1, E_1, I_1 \rangle$ and $S_2 = \langle X_2, \Sigma_2, E_2, I_2 \rangle$ are considered bisimilar (denoted $S_1 \approx S_2$) iff there exists a relation $\mathcal{R} \subseteq X_1 \times X_2$ such that for all initial states $i_1 \in I_1$ there exists a state $i_2 \in I_2$ that is related ($i_1 \mathcal{R} i_2$) and vice versa, and furthermore for all $x_1 \in X_1, x_2 \in X_2, \sigma \in \Sigma_1 \cup \Sigma_2$*

- $(x_1 \mathcal{R} x_2 \text{ and } x_1 \xrightarrow{\sigma} x'_1)$ implies there exists $x'_2 \in X_2$ such that $(x_2 \xrightarrow{\sigma} x'_2 \text{ and } x'_1 \mathcal{R} x'_2)$, and
- $(x_1 \mathcal{R} x_2 \text{ and } x_2 \xrightarrow{\sigma} x'_2)$ implies there exists $x'_1 \in X_1$ such that $(x_1 \xrightarrow{\sigma} x'_1 \text{ and } x'_1 \mathcal{R} x'_2)$.

Intuitively, bisimilarity considers two LTSs equivalent if both have a similar branching structure, starting from their initial states. In the case of the

bouncing ball there is actually no branching, but the path is infinitely long. Suppose that the initial velocity $i_v \neq 0$, then (in both B_1 and B_2) there is only a transition to a new state (v, t) in which still $v \neq 0$. Always, when starting from a non-zero velocity the systems will be in a state with a non-zero velocity after any arbitrary number of transitions. Therefore, the systems never end up in a state $(0, t)$ for any t . The relation $\mathcal{R} \subseteq X_1 \times X_2$ such that $(x_1, x_2) \in \mathcal{R}$ iff $x_1 = x_2$ and $x_1 \neq (0, t)$ will fulfill the requirements in the definition of bisimilarity. From this we conclude that B_1 and B_2 are bisimilar (if $i_v \neq 0$).

As we have seen, two systems that display Zeno-behaviour can be bisimilar while intuitively they are considered different. This lack in distinctive power is caused by the fact that bisimilarity cannot ‘see beyond’ the point at which a sequence of states accumulates (the Zeno-point). In the next section, we focus on how infinite sequences can be considered in general in labelled transition systems. After that, we formalize what we mean by accumulation and give a definition of the accumulation set of an infinite sequence of states in the context of a metric space. Finally, we revisit the bouncing ball example to see what has changed.

3 Sequence-labelled Transition Systems

In ordinary labelled transition systems there exists a transition $x \xrightarrow{\sigma} x'$ iff $(x, \sigma, x') \in E$. When thinking about the extension to sequences of labels it is quite intuitive to define a kind of transitive closure that says: if there is a transition $x \xrightarrow{\sigma_1} x'$ and a transition $x' \xrightarrow{\sigma_2} x''$ then there should also be a transition $x \xrightarrow{\sigma_1 \sigma_2} x''$. The first thing that we have to do if we want such a transition in the system is to extend the set of labels Σ with all sequences (finite and infinite) of labels. But before we do that we need some preliminary notions on ordinal numbers and sequences.

Like in the paper by Bérard and Picaronny [1], we use ordinals mainly to number elements of sequences. We would like to recall that the finite ordinals are the natural numbers and the first non-finite ordinal is denoted by ω . (Note that the sum operation on ordinals is not commutative. For example, $1+\omega = \omega$ while $\omega+1 = \text{successor of } \omega$.) In this paper we only consider ordinals smaller than ω^ω . They have a polynomial decomposition $\alpha = \sum_{k=p}^0 \omega^k \cdot n_k$, where $p, n_0, n_1 \dots$ are natural numbers. (As expected $\omega^0 = 1$). Furthermore a limit ordinal is an ordinal for which $n_0 = 0$ in the polynomial decomposition.

Sequences are constructed through the concatenation of elements in an alphabet. If we have the set A as an alphabet, then A^n denotes the set of sequences $a_1a_2\dots a_n$, of n elements of the set A . For infinite sequences, n is an ordinal number. $A^{<n}$ denotes the set of all sequences of length smaller than n .

Definition 3.1 (SLTS) A sequence-labelled transition system is an LTS $\langle X, \Sigma, E, I \rangle$ in which Σ is a set of labels consisting of all sequences over an alphabet A , i.e. $\Sigma = A^{<\omega}$.

The purpose of considering sequence-labelled transition systems is to allow for the kind of transitive closure mentioned before. However, this definition actually gives us the chance to build very awkward transition systems that are not transitively closed. See, for example, the one depicted in figure 4. To exclude such awkward SLTSs we need to formalize the notion of prefix-closedness.

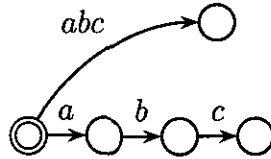


Figure 4: Not Prefix-closed SLTS

Definition 3.2 (prefix, postfix) Suppose we have a sequence $\underline{\sigma} = \sigma_1\sigma_2\dots\sigma_n$ then, for $i \leq n$, $\underline{\sigma}^i$ denotes the prefix $\sigma_1\sigma_2\dots\sigma_i$ and $\underline{\sigma}^i$ denotes the postfix $\sigma_i\sigma_{i+1}\dots\sigma_n$.

Definition 3.3 (prefix-closedness) An SLTS $S = \langle X, \Sigma, E, I \rangle$ is said to be prefix-closed iff for all $x, y \in X$ and $\underline{\sigma} \in \Sigma$

$$x \xrightarrow{\underline{\sigma}} y \Rightarrow \forall_{1 \leq i < n} \exists_{x'} (x \xrightarrow{\underline{\sigma}^i} x' \wedge x' \xrightarrow{\underline{\sigma}^{i+1}} y),$$

with n the length of $\underline{\sigma}$.

Trivially, an SLTS that only contains transitions with atomic labels (sequences of length 1) is prefix-closed by definition.

By introducing the notion of Prefix-closedness we have excluded labelled transition systems as in figure 4. Still, there is another intuition that is not satisfied yet. In the case of Zeno-behaviour we recognize that transitions over sequences with a limit ordinal length, should end in a certain limit-state, that is in some sense related to the transitions over all prefixes of the sequence.

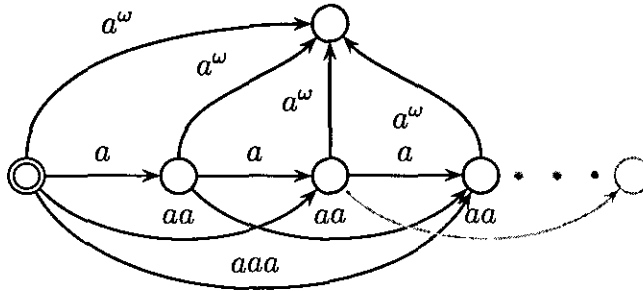


Figure 5: Prefix-closed Infinite SLTS

The example in figure 5 shows a prefix-closed SLTS in which the transitions a^ω ($\underline{\sigma}^\omega$ denotes the infinite repetition of the sequence $\underline{\sigma}$) point to a state that is not clearly related to the finite transitions. To solve this problem, we introduce a metric as a means to relate states and a notion of convergence to a limit-state based on that metric. Systems in which all infinite transitions end in their limit-states are said to be limit-closed.

4 Metric Spaces

In the introduction we noted that in case of Zeno-behaviour, a sequence of states accumulates at a certain accumulation state (or set of states as we will see further on). However, to be able to formally define such an accumulation we first need to have a notion of distance between states, i.e. they must be in a metric space [4]. (Strictly speaking we only need a topology on the states, [6, 2], but the state-spaces we are interested in all have a topology defined by a metric.) We base the definitions and the results that we give in this section on the usual definitions of topology [6, 2].

Definition 4.1 (metric space) *A metric space $\langle X, |\cdot, \cdot| \rangle$ is a set X together with a binary distance function (or metric) $|\cdot, \cdot| : X \times X \rightarrow \mathbb{R}^+$ such that for*

all $x, x', x'' \in X$

$$\begin{aligned} |x, x'| &= 0, \text{ iff } x = x', \\ |x, x'| &= |x', x|, \\ |x, x''| &\leq |x, x'| + |x', x''|. \end{aligned}$$

In this metric space we define a notion of accumulation.

Definition 4.2 (accumulation) *A sequence $\underline{x} = x_1x_2\dots x_n$ over a metric space X is said to accumulate (or cluster) at $y \in X$ iff either n is not a limit ordinal and $y = x_n$, or $\forall \epsilon > 0 \forall k < n \exists k \leq l < n |x_l, y| < \epsilon$.*

Note that a sequence may accumulate at multiple points. For example the behaviour of a process may be such that a sequence of events accumulates at, for example, a circle rather than at a single state. When we study automata on infinite words, rather than the more general transition systems, the set of accumulation points becomes the set of states that are visited infinitely often. This is directly related to the definition of limits by Bérard and Picaronny [1]. Therefore, we also denote the set of accumulation points using \lim .

Definition 4.3 (accumulation set) *The accumulation set of a sequence $\underline{x} = x_1x_2\dots x_n$, notation $\lim \underline{x}$, is the set $\lim \underline{x} = \{y \mid \underline{x} \text{ accumulates at } y\}$.*

It should be noted that this accumulation set is sometimes empty. Take for example the sequence 1,2,3,..., of course with distance defined as the absolute difference in value. There is no point to which the sequence accumulates because none of the points is visited twice (within arbitrary small distance). If the sequence lies in a compact metric space, it can be shown that the accumulation set contains at least one point [2].

Another useful property, which is not hard to prove, is that the accumulation set of a sequence is prefix-insensitive, i.e. we can remove the first part of the sequence and still obtain the same accumulation set.

Lemma 4.4 (prefix-insensitivity) *For any sequence \underline{x} of length n , with n a limit ordinal, and any $1 \leq i < n$ we find $\lim \underline{x} = \lim \underline{x}^i$.*

Proof The proof is almost trivial. The only difficulty might be in recognizing that, in the definition of accumulation, prefixes are not considered.

The value of k in the definition of accumulation, may be taken higher than the end of the prefix i . So as long as the postfixes are equal, both result in the same accumulation set. \boxtimes

5 Sequence-labelled Metric Transition Systems

Now we have defined the notion of metric space and know what convergence means in this light, we are ready to use a metric space as the set of states in a transition system. Completely analogous to the definitions given before we define the following.

Definition 5.1 (SLMTS) *A sequence-labelled metric transition system is an SLTS $\langle X, \Sigma, E, I \rangle$ in which the state space X is a metric space.*

In case of an SLMTS we would not only like it to be prefix-closed but we would also like to find that transitions over an infinite sequence end in the accumulation set of the corresponding sequence of states. If an SLMTS behaves in that way we call it accumulation-closed.

Definition 5.2 (accumulation-closed) *An SLMTS $S = \langle X, \Sigma, E, I \rangle$ is accumulation-closed iff for all $x, y \in X$ and $\underline{\sigma} \in \Sigma$:*

$$x \xrightarrow{\underline{\sigma}} y \Rightarrow \exists \underline{x} \forall_{1 \leq i < n} (x \xrightarrow{i \underline{\sigma}} x_i \wedge y \in \lim \underline{x}),$$

with n the length of $\underline{\sigma}$.

Often, when we are specifying a system, we only want to specify the transitions that contain atomic labels and assume that all possible transitions over sequences of labels that do not violate prefix- and accumulation-closedness are automatically included also.

Definition 5.3 (transitive closure) *Given an SLMTS $M = \langle X, \Sigma, E, I \rangle$ we inductively define the transitive closure of M (denoted M^T) as $M^T = \langle X, \Sigma, E_T, I \rangle$ with E_T the smallest set such that*

- $E \subseteq E_T$,
- $(x, \underline{\sigma}, x') \in E_T \wedge (x', \underline{\sigma}^{i+1}, y) \in E_T$ implies $(x, \underline{\sigma}, y) \in E_T$, and
- for all limit ordinals n and sequences \underline{x} of length n for which the accumulation set exists: $\forall_{1 \leq i < n} ((x_i, \sigma_i, x_{i+1}) \in E_T \wedge y \in \lim \underline{x})$ implies $(x_1, \underline{\sigma}, y) \in E_T$.

By induction on the definition, the proof of $M^T = M^{TT}$ is trivial, therefore this operation can indeed be considered a closure. We would, however, also like to see that prefix- and accumulation-closedness are invariant under transitive closure.

Lemma 5.4 (prefix-closedness invariance) *Let M be a sequence-labelled metric transition system. If M is prefix-closed, then M^T is also prefix-closed.*

Proof Using induction over the definition of the closure we only need to prove that any transition that is added by the closure operator does not violate the prefix-closedness of the system. We separate three cases, according to the three rules of the definition.

- The trivial case is when a transition that is added to E_T is also in E (first rule.)
- If a transition is added according to the second rule we know there is a decomposition into a prefix and postfix of the sequence over which the transitions are already in E_T . Since E_T is prefix-closed, all possible decompositions of the prefix and postfix are also in E_T . Therefore all possible decompositions of the new transition are added by the closure operator. Thus the new system is still prefix-closed.
- If a transition is added according to the third rule a similar reasoning holds. Since a complete decomposition into transitions over atomic sequences, that are all in E_T , leads to the new transition, all possible decompositions of the new transition are added by the closure operator. Therefore the new system is still prefix-closed.

Thus, in any case, the new transition system is still prefix-closed, which concludes the proof. \boxtimes

Lemma 5.5 (accumulation-closedness invariance) *Given a sequence-labelled metric transition system M , if M is accumulation-closed, then M^T is also accumulation-closed.*

Proof As before we use induction over the definition of closure, separating three cases.

- Adding a transition by the first rule is again the trivial case.
- Adding a transition according to the second rule, by prefix-insensitivity of the accumulation set, comes down to showing that the transition over the postfix $\underline{\sigma}^i$ is accumulation-closed. Which is true since the postfix was already in E_T .
- The proof for adding a transition according to the third rule is also easy. The transition consists of a decomposition in atomic transitions, that are in E_T and accumulation-closed by definition. Adding a transition to the elements of the accumulation set of the sequence of these atomic transitions does therefore not change the accumulation-closedness.

This concludes the proof. □

Suppose that we have SLMTSs which only contain atomic transitions, and their transitive closures are bisimilar, then the original SLMTSs are also bisimilar. In the next section, where we revisit the bouncing ball, this shows not to be true the other way around. Bisimilarity after transitive closure is therefore a stronger kind of equivalence.

Lemma 5.6 (bisimilarity after closure) *For any two SLMTSs X and Y containing only atomic transitions we find $X^T \Leftrightarrow Y^T \Rightarrow X \Leftrightarrow Y$.*

Proof Let \mathcal{R} be a bisimulation relation such that $X^T \Leftrightarrow Y^T$. In particular this relation relates the points that are connected through atomic transitions. Since the transitive closure does not add atomic transitions, \mathcal{R} can also be used to relate X and Y . □

6 Concluding Remarks

Now, let us return to the bouncing ball. We already defined B_1 and B_2 as two different realisations of the bouncing ball in which the state of the system was characterized by $(v, t) \in \mathbb{R}^2$. This happens to be a metric space already, so, if we replace the set of labels Σ by all sequences of the event *bounce* then we converted B_1 and B_2 into sequence-labelled metric transition systems. Now, for the transitive closure of these systems (B_1^T and B_2^T) it is easy to show that $[v_0, 0] \xrightarrow{\text{bounce}^\omega} [\lim_{n \rightarrow \infty} v_n, \lim_{n \rightarrow \infty} t_n]$ because of accumulation-closedness. We already showed this to be equal to: $[v_0, 0] \xrightarrow{\text{bounce}^\omega} [0, t_{zeno}]$. The bisimulation relation that we gave does not hold anymore since originally states with zero-velocity were not related. Actually, there cannot exist a bisimulation relation that matches the two bouncing ball specifications because from every initial point we are able to get into a point $[0, t]$ using an infinite number of *bounces*. The 0-velocity points cannot be related since per definition there are no transitions from it in B_1^T , while there are in B_2^T .

As we have seen, using the transitive closure on sequence-labelled metric transition systems, the problem of Zeno-behaviour can be overcome. In practice however, it is impossible to calculate the infinite number of transitions that have to be added to construct the transitive closure. Our hopes are on finding an (algebraic) axiomatization that allow us to do calculations with and reason about transitively closed SLMTS without actually constructing them. In fact, the sequence-labelled metric transition systems that are described in this paper, might be useful as general semantics for methods in the field of hybrid systems. We suspect that accumulation-closure will prove to be a powerful tool in general, when coping with transfinite behaviour. In this perspective it might also provide a link between the computing science fields of process algebra and temporal logic, provided that we can find an appropriate metric or general topology on processes.

Acknowledgements: Finally, we would like to thank Aleksandar Juloski for making a large contribution, finding information on previous research in the field and furthermore (in random order) Paul van den Bosch, Jan Friso Groote, Maurice Heemels, Ingrid Flinsenberg, Victor Bos, Jeroen Kleijn and Tim Willemse for greater and smaller contributions due to some of the discussions we had on this topic.

References

- [1] B. Bérard and C. Picaronny. Accepting zeno words: a way toward timed refinements. *Acta Informatica*, 37(1):45–81, 2000.
- [2] C.R. Borges. *Elementary Topology and Applications*. World Scientific Publishing Co. Pte. Ltd., 2000.
- [3] F. van Breugel. A theory of metric labelled transition systems. In S. Andima, B. Flagg, J. Itzkowitz, Y. Kong, R. Kopperman, and P. Misra, editors, *Papers on General Topology and Applications: 11th Summer Conference at the University of Southern Maine*, volume 806 of *Annals of the New York Academy of Sciences*, pages 67–87, Gorham, August 1995. The New York Academy of Sciences.
- [4] V. Bryant. *Metric Spaces: Iteration and Application*. Cambridge University Press, 1985.
- [5] R. Büchi. The monadic second order theory of ω_1 . In Siefkes and R. Büchi, editors, *The Monadic Second Order Theory of All Countable Ordinals*, volume 328 of *Lecture Notes in Mathematics*, pages 1–126, Berlin, 1973. Springer.
- [6] J. Dugundji. *Topology*. Allyn and Bacon, Inc., Boston, 1966.
- [7] K.H. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry. On the regularization of zeno hybrid automata. *System and Control Letters*, 38:141–150, 1999. Special issue on Hybrid Systems.
- [8] R.E. Kent. The metric closure powerspace construction. In M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Proceedings of the 3rd Workshop on Mathematical Foundations of Programming Language Semantics*, volume 298 of *Lecture Notes in Computer Science*, pages 173–199, New Orleans, April 1987. Springer-Verlag.
- [9] A.J. van der Schaft and J.M. Schumacher. An introduction to hybrid dynamical systems. volume 251 of *Lecture Notes in Control and Information Sciences*, London, 2000. Springer-Verlag.

Computer Science Reports

Department of Mathematics and Computer Science Technische Universiteit Eindhoven

If you want to receive reports, send an email to: m.m.j.l.philips@tue.nl (we cannot guarantee the availability of the requested reports)

In this series appeared:

97/02	J. Hooman and O. v. Roosmalen	A Programming-Language Extension for Distributed Real-Time Systems, p. 50.
97/03	J. Blanco and A. v. Deursen	Basic Conditional Process Algebra, p. 20.
97/04	J.C.M. Baeten and J.A. Bergstra	Discrete Time Process Algebra: Absolute Time, Relative Time and Parametric Time, p. 26.
97/05	J.C.M. Baeten and J.J. Vereijken	Discrete-Time Process Algebra with Empty Process, p. 51.
97/06	M. Franssen	Tools for the Construction of Correct Programs: an Overview, p. 33.
97/07	J.C.M. Baeten and J.A. Bergstra	Bounded Stacks, Bags and Queues, p. 15.
97/08	P. Hoogendijk and R.C. Backhouse	When do datatypes commute? p. 35.
97/09	Proceedings of the Second International Workshop on Communication Modeling, Veldhoven, The Netherlands, 9-10 June, 1997.	Communication Modeling- The Language/Action Perspective, p. 147.
97/10	P.C.N. v. Gorp, E.J. Luit, D.K. Hammer E.H.L. Aarts	Distributed real-time systems: a survey of applications and a general design model, p. 31.
97/11	A. Engels, S. Mauw and M.A. Reniers	A Hierarchy of Communication Models for Message Sequence Charts, p. 30.
97/12	D. Hauschildt, E. Verbeek and W. van der Aalst	WOFLAN: A Petri-net-based Workflow Analyzer, p. 30.
97/13	W.M.P. van der Aalst	Exploring the Process Dimension of Workflow Management, p. 56.
97/14	J.F. Groote, F. Monin and J. Springintveld	A computer checked algebraic verification of a distributed summation algorithm, p. 28
97/15	M. Franssen	λP^- : A Pure Type System for First Order Loginc with Automated Theorem Proving, p.35.
97/16	W.M.P. van der Aalst	On the verification of Inter-organizational workflows, p. 23
97/17	M. Vaccari and R.C. Backhouse	Calculating a Round-Robin Scheduler, p. 23.
97/18	Werkgemeenschap Informatiewetenschap redactie: P.M.E. De Bra	Informatiewetenschap 1997 Wetenschappelijke bijdragen aan de Vijfde Interdisciplinaire Conferentie Informatiewetenschap, p. 60.
98/01	W. Van der Aalst	Formalization and Verification of Event-driven Process Chains, p. 26.
98/02	M. Voorhoeve	State / Event Net Equivalence, p. 25
98/03	J.C.M. Baeten and J.A. Bergstra	Deadlock Behaviour in Split and ST Bisimulation Semantics, p. 15.
98/04	R.C. Backhouse	Pair Algebras and Galois Connections, p. 14
98/05	D. Dams	Flat Fragments of CTL and CTL*: Separating the Expressive and Distinguishing Powers. P. 22.
98/06	G. v.d. Bergen, A. Kaldewaij V.J. Dielissen	Maintenance of the Union of Intervals on a Line Revisited, p. 10.
98/07	Proceedings of the workshop on Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98) June 22, 1998 Lisbon, Portugal	edited by W. v.d. Aalst, p. 209
98/08	Informal proceedings of the Workshop on User Interfaces for Theorem Provers. Eindhoven University of Technology, 13-15 July 1998	

		edited by R.C. Backhouse, p. 180
98/09	K.M. van Hee and H.A. Reijers	An analytical method for assessing business processes, p. 29.
98/10	T. Basten and J. Hooman	Process Algebra in PVS
98/11	J. Zwanenburg	The Proof-assistent Yarrow, p. 15
98/12	Ninth ACM Conference on Hypertext and Hypermedia Hypertext '98 Pittsburgh, USA, June 20-24, 1998 Proceedings of the second workshop on Adaptive Hypertext and Hypermedia. Edited by P. Brusilovsky and P. De Bra, p. 95.	
98/13	J.F. Groote, F. Monin and J. v.d. Pol	Checking verifications of protocols and distributed systems by computer. Extended version of a tutorial at CONCUR'98, p. 27.
99/01	V. Bos and J.J.T. Kleijn	Structured Operational Semantics of χ , p. 27
99/02	H.M.W. Verbeek, T. Basten and W.M.P. van der Aalst	Diagnosing Workflow Processes using Woflan, p. 44
99/03	R.C. Backhouse and P. Hoogendijk	Final Dialgebras: From Categories to Allegories, p. 26
99/04	S. Andova	Process Algebra with Interleaving Probabilistic Parallel Composition, p. 81
99/05	M. Franssen, R.C. Veltkamp and W. Wesselink	Efficient Evaluation of Triangular B-splines, p. 13
99/06	T. Basten and W. v.d. Aalst	Inheritance of Workflows: An Approach to tackling problems related to change, p. 66
99/07	P. Brusilovsky and P. De Bra	Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, p. 119.
99/08	D. Bosnacki, S. Mauw, and T. Willemse	Proceedings of the first international syposium on Visual Formal Methods - VFM'99
99/09	J. v.d. Pol, J. Hooman and E. de Jong	Requirements Specification and Analysis of Command and Control Systems
99/10	T.A.C. Willemse	The Analysis of a Conveyor Belt System, a case study in Hybrid Systems and timed μ CRL, p. 44.
99/11	J.C.M. Baeten and C.A. Middelburg	Process Algebra with Timing: Real Time and Discrete Time, p. 50.
99/12	S. Andova	Process Algebra with Probabilistic Choice, p. 38.
99/13	K.M. van Hee, R.A. van der Toorn, J. van der Woude and P.A.C. Verkoulen	A Framework for Component Based Software Architectures, p. 19
99/14	A. Engels and S. Mauw	Why men (and octopuses) cannot juggle a four ball cascade, p. 10
99/15	J.F. Groote, W.H. Hesselink, S. Mauw, R. Vermeulen	An algorithm for the asynchronous <i>Write-All</i> problem based on process collision*, p. 11.
99/16	G.J. Houben, P. Lemmens	A Software Architecture for Generating Hypermedia Applications for Ad-Hoc Database Output, p. 13.
99/17	T. Basten, W.M.P. v.d. Aalst	Inheritance of Behavior, p.83
99/18	J.C.M. Baeten and T. Basten	Partial-Order Process Algebra (and its Relation to Petri Nets), p. 79
99/19	J.C.M. Baeten and C.A. Middelburg	Real Time Process Algebra with Time-dependent Conditions, p.33.
99/20	Proceedings Conferentie Informatiewetenschap 1999 Centrum voor Wiskunde en Informatica 12 november 1999, p.98 edited by P. de Bra and L. Hardman	
00/01	J.C.M. Baeten and J.A. Bergstra	Mode Transfer in process Algebra, p. 14
00/02	J.C.M. Baeten	Process Algebra with Explicit Termination, p. 17.
00/03	S. Mauw and M.A. Reniers	A process algebra for interworkings, p. 63.
00/04	R. Bloo, J. Hooman and E. de Jong	Semantical Aspects of an Architecture for Distributed Embedded Systems*, p. 47.

00/05	J.F. Groote and M.A. Reniers	Algebraic Process Verification, p. 65.
00/06	J.F. Groote and J. v. Wamel	The Parallel Composition of Uniform Processes wit Data, p. 19
00/07	C.A. Middelburg	Variable Binding Operators in Transition System Specifications, p. 27.
00/08	I.D. van den Ende	Grammars Compared: A study on determining a suitable grammar for parsing and generating natural language sentences in order to facilitate the translation of natural language and MSC use cases, p. 33.
00/09	R.R. Hoogerwoord	A Formal Development of Distributed Summation, p. 35
00/10	T. Willemse, J. Tretmans and A. Klomp	A Case Study in Formal Methods: Specification and Validation on the OM/RR Protocol, p. 14.
00/11	T. Basten and D. Bošnački	Enhancing Partial-Order Reduction via Process Clustering, p. 14
00/12	S. Mauw, M.A. Reniers and T.A.C. Willemse	Message Sequence Charts in the Software Engineering Process, p. 26
00/13	J.C.M. Baeten, M.A. Reniers	Termination in Timed Process Algebra, p. 36
00/14	M. Voorhoeve, S. Mauw	Impossible Futures and Determinism, p. 14
00/15	M. Oostdijk	An Interactive Viewer for Mathematical Content based on Type Theory, p. 24.
00/16	F. Kamareddine, R. Bloo, R. Nederpelt	Characterizing λ -terms with equal reduction behavior, p. 12
00/17	T. Borghuis, R. Nederpelt	Belief Revision with Explicit Justifications: an Exploration in Type Theory, p. 30.
00/18	T. Laan, R. Bloo, F. Kamareddine, R. Nederpelt	Parameters in Pure Type Systems, p. 41.
00/19	J. Baeten, H. van Beek, S. Mauw	Specifying Internet applications with <i>DiCons</i> , p. 9
00/20	Editors: P. v.d. Vet and P. de Bra	Proceedings: Conferentie Informatiewetenschap 2000, De Doelen, Utrecht, 5 april 2000, p. 98
01/01	H. Zantema and J. v.d. Pol	A Rewriting Approach to Binary Decision Diagrams, p. 27
01/02	T.A.C. Willemse	Interpretations of Automata, p. 41
01/03	G.I. Joigov	Systems for Open Terms: An Overview, p. 39