Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Theoretical Computer Science 300 (2003) 331–363

Theoretical
Computer Science

www.elsevier.com/locate/tcs

Automata over continuous time

Alexander Rabinovich

Department of Computer Science, Tel-Aviv University, Ramat Aviv 69978, Israel

Received 15 February 2000; received in revised form 18 December 2001; accepted 1 February 2002
Communicated by W. Thomas

Abstract

The principal objective of this paper is to lift basic concepts of the classical automata theory from discrete to continuous (real) time. It is argued that the set of *finite memory retrospective functions* is the set of functions realized by finite state devices. We show that the finite memory retrospective functions are speed-independent, i.e., they are invariant under ‘stretchings’ of the time axis. Therefore, such functions cannot deal with metrical aspects of the reals.

We classify and analyze phenomena which appear at continuous time and are invisible at discrete time.

1. Introduction

The principal objective of this paper is to lift basic concepts of the classical automata theory from discrete to continuous (real) time. The shift to continuous time brings to surface phenomena that are invisible at discrete time. A second major task of the paper is to provide a careful analysis of continuous time phenomena that are interesting for their own. The results of this paper were obtained in the framework of a general program worked out by Trakhtenbrot [19,15,13,20] for lifting classical automata theory from discrete to continuous time.

It is common to introduce automata theory as a study of sets of strings (or of ω-strings) accepted by finite machines (devices). However, the functions realized by various machines are more basic than the sets accepted by these devices. This is in accordance with the belief that in automata theory as well as in computability theory functions are more fundamental than sets. This point of view is implicit already in the

E-mail addresses: rabino@math.tau.ac.il, alexr@dcs.ed.ac.uk (A. Rabinovich).

0304-3975/03/$ - see front matter
PII: S0304-3975(02)00331-6

classical works of Pitts–Mc.Culloch [11], Kleene [6] and it was consistently pursued by Trakhtenbrot [7,21]. Here is Scott's argumentation [16] in favor of this view:

> The author (along with many other people) has come recently to the conclusion that the functions computed by the various machines are more important—or at least more basic—than the sets accepted by these devices. The sets are still interesting and useful, but the functions are needed to understand the sets. In fact by putting the functions first, the relationship between various classes of sets becomes much clearer. This is already done in recursive function theory and we shall see that the same plan carriers over to the general theory.

Therefore, here our main interest will be in the functions realized by finite machines operating in continuous [1] time.

An obvious transition from discrete time to continuous time is as follows: instead of signals defined over a discrete sequence of time instants (i.e., strings or ω-strings), consider signals defined over the non-negative reals (i.e., the functions from $[0,\infty)$ into a finite alphabet). Also, instead of functions that map ω-strings into ω-strings, consider functions that manipulate continuous time signals. A more realistic approach would reject a 'signal' with value 1 on rational time instants and value 0 otherwise. Indeed, it is reasonable to confine with 'signals' that are piecewise constant functions of time (such functions are often called non-Zeno signals), and to formalize an appropriate notion of 'realistic' operators. Various formalizations are discussed in Section 2.

The paper is organized as follows.

Section 2—*Postulates of automata theory*. In recent years, many extensions of discrete time formalisms to continuous time have been suggested. Sometimes, the presentation of these continuous time formalisms is obscured by ad hoc definitions and notations. The aim of this section is to define in an axiomatic way the behavior of finite state devices operating in continuous time. We state explicitly the postulates of automata theory and lift them from discrete to continuous time. Basic terminology and notations of automata theory are extended to continuous time; *finite memory retrospective functions* are defined and it is argued that this is the class of functions which is realized by finite state devices.

Section 3—*Stability and speed-independence*. The shift to continuous time brings to surface properties of signals and functions over signals that are invisible at discrete time. For example, the unit delay is a finite memory function in the discrete case, whereas continuous time forces the delay to memorize an uncountable amount of information. Another important property of functions is 'speed-independence'. An operator is speed-independent if it is invariant under 'stretchings' of the time axis. In discrete time all operators are obviously speed-independent, because of the lack of non-trivial 'stretchings'. For continuous time, speed-independence is a nontrivial property; it fails for unit delays, however we show that finite memory functions are speed-independent (Theorem 21). Unfortunately, it turns out that finite state devices are unable to compute functions which deal with metrical aspects of real line. The metrical

[1] We use the word 'continuous' for the time domain of the reals. The phrase 'real time' is overloaded, so we prefer to use 'continuous time'.

properties are the most intriguing characteristic of real-time specifications which have led to a large number of highly interesting results and applications (see Proceedings of Hybrid and Real-Time Workshops, e.g. [9]).

In Section 4, the definitions are illustrated by numerous examples, which point to subtleties and warn against likely misjudgments.

In Section 5, we provide a faithful representation of speed-independent functions over 'realistic' signals by functions over ω-strings.

Section 6 states some closure properties of the finite memory functions, the speed-independent functions and the stable functions. In Section 7 properties of finite memory retrospective functions are investigated. The proof of the main technical proposition is deferred to the appendix.

In Section 8, representations of finite memory functions is discussed. It is shown there that (1) finite memory functions over piecewise constant signals can be represented by finite transition diagrams, however (2) no finite representation is possible for the finite memory functions over the general signals. We also show that for functions on signals the property of being finite memory implies speed independence.

In Section 9, related results are discussed.

2. Postulates of automata theory

In recent years many extensions of discrete time formalisms to continuous time have been suggested. Sometimes, the presentation of these continuous time formalisms is obscured by ad hoc definitions and notations. The aim of this section is to define in an axiomatic way the behavior of finite state devices operating in continuous time. Most of the ideas, concepts and notions we rely on have been employed for almost forty years for the description of the behavior of finite devices operating in discrete time. In particular, the same notions and terminology as in [17,21] are used in this section. Our contribution here is only in explicit formulation of all these assumptions.

A machine is considered as a black box with input and output channels. Over the time, the user acts to a machine through the input channels and the machine produces an output over its output channels. This is a very simple form of interaction between a machine and a user (environment). The output of a machine does not influence the behavior of the environment. In this paper only this simplest form of interaction is considered.

In the next three subsections we state explicitly the postulates which underly the classical automata theory and re-examine them.

2.1. Nature of time

The first group of postulates of classical automata theory deals with the nature of time.

Linear time: The set of moments of time is a linearly ordered set.

Discrete time: Every natural number represents a time moment and vice versa; the number zero represents the beginning of time [3].

In this paper discrete time postulate is replaced by

Continuous time: Time is continuous; every time moment is represented by a non-negative real and vice versa; the number zero represents the beginning of time.

2.2. Finiteness postulates on machines

The following postulates are also assumed.

Finiteness of number of channels: A machine has a finite number of input and output channels.

Finiteness of number of channel states: At any moment of time a channel can be in one of a finite number possible states.

If the set of possible states of a channel ch is Σ, we say that ch is a Σ-channel.

The last finiteness postulate deals with the finiteness of memory and will be explained in the next subsection.

2.3. Input–output Behavior

A *signal* over a channel is a function from time to the set of the channel's states. Hence, a continuous (respectively, discrete) time signal over a Σ-channel is a function from the non-negative reals (respectively, natural numbers) into Σ.

The postulates in this section deal with the input–output behavior of a machine.

Deterministic behavior: The output signals are completely determined by the input signals.

Hence, the input–output behavior of a machine is a function from the signals over the input channels to the signals over the output channels.

It is natural to assume that an input at a present moment cannot influence the output produced in the past (before the present moment). Hence, we require

Causal behavior: The output at a moment t does not depend on the inputs at later time.

Sometimes the causal behavior postulate is strengthened as follows:

Strong causal behavior: The output at a moment t does not depend on the inputs at moment t and at later moments.

The following definition formalizes these concepts.

Definition 1 (Retrospective and strongly retrospective functions [17]). Let F be a function from signals to signals.

- F is retrospective if for any signals x, y and time moment t the following condition holds: If x and y coincide in the interval $[0,t]$ then Fx and Fy coincide in the interval $[0,t]$.
- F is strongly retrospective if for any signals x, y and time moment t the following condition holds: If x and y coincide in the interval $[0,t)$ then Fx and Fy coincide in the interval $[0,t]$.

Hence, the above postulates imply that the input–output behavior of a machine is a (strongly) retrospective function.

The last postulate is a key postulate of finite automata theory.

> For a given machine M at a given time moment t we can imagine an infinite variety of possible signal histories that M has received priory to t. The one that actually occurred will determine the future behavior of M. [12]

Finite memory [12]: A (finite state) machine can distinguish by its present and future behavior between only a finite number of classes of possible signal histories.

In the rest of this section a formalization of this postulate is suggested. However, first some notations and terminology are introduced which will be used throughout the paper.

Notation and terminology: $\mathbf{R}^{\geqslant 0}$ is the set of non-negative reals; **BOOL** is the set of booleans and Σ is a finite set (alphabet). Letters t, τ will range over non-negative reals or integers, x, y, z will range over signals and F, G over functions from signals to signals, and a, b, c over elements of an alphabet. We use $Sig(\Sigma)$ for the set of signals over Σ.

The notation fv is used for the application of a function f to an element v, however, sometimes to improve the readability parenthesis will be used, and the application of f to v will be denoted by $f(v), (f)v$ or $(f)(v)$; application is left associative, so fvu will be an abbreviation for $(fv)u$; the notation $f \circ g$ is used for the composition of functions f and g, which is the function $\lambda x.g(fx)$; the notation f^{-1} is used for the inverse of a function f.

A *t-history* (over an alphabet Σ) is a function from the interval $[0, t]$ into Σ. A t-history h is a t-history of a signal x if $h(\tau) = x(\tau)$ for $\tau \leqslant t$.

The restriction of x to the interval $[0, t)$ is called *t-prefix* of x. The *suffix* of x at t (notation $suf(x, t)$) is the signal y defined as $y(t') = x(t+t')$, i.e., $suf(x, t) = \lambda t'.x(t+t')$.

We sometimes use $x\rfloor^t$ for the restriction of x to the interval $[0, t)$; similarly, we use $x]^t$ (respectively, $x\lfloor_t$ and $x[_t$) for the restriction of x to the interval $[0, t]$ (respectively, to the interval (t, ∞) and to the interval $[t, \infty)$). Let x and z be two signals. The *concatenation* of t-prefix of x and z (notation $x\rfloor^t; z$) is defined as:

$$(x\rfloor^t; z)(\tau) = \begin{cases} x(\tau) & \text{if } \tau < t, \\ z(\tau - t) & \text{if } \tau \geqslant t. \end{cases}$$

Now let us proceed with the formalization of finite memory.

First, we define when a t-history h_1 is indistinguishable from (or equivalent to) t-history h_2 and afterward define when histories over different time intervals are indistinguishable.

Let us imagine that we have two copies M_1 and M_2 of a "machine" M transforming signals into signals. Assume that two signals x_1 and x_2 pass over the inputs of M_1 and M_2 respectively. Assume further that x_1 and x_2 coincide on $[t, \infty)$ and that h_1 is the t-history of x_1 and h_2 is the t-history of x_2. If h_1 and h_2 are indistinguishable by the future behavior of M then at time moment t and after it, both M_1 and M_2 should produce the same output i.e., $\forall x_1 x_2.(h_1 = x_1\rfloor^t \wedge h_2 = x_2\rfloor^t \wedge suf(x_1, t) = suf(x_2, t)) \Rightarrow suf(Fx_1, t) = suf(Fx_2, t)$.

The preceding paragraph suggests the following:

Definition 2 (Residual [17]). Let F be a function on signals, x be a signal and t a time moment. The residual of F with respect to x and t is the function $\lambda z.\lambda t'.F(x\rfloor^t;z)(t+t')$.

Remark. The residual of F wrt x and t maps signal z on z' if and only if F maps $x\rfloor^t;z$ on $y\rfloor^t;z'$ for some y.

We use the notation $Res(F,x,t)$ for the residual of F wrt x and t. We say that G is a residual of F if for some x and t the function G is the residual of F wrt x and t.

Example (*Unit delays*). Let Σ be an alphabet. For $a\in\Sigma$ define:

$$Delay_a(x)(t) = \begin{cases} a & \text{if } t < 1. \\ x(t-1) & \text{otherwise.} \end{cases}$$

It is easy to see that over discrete time N, for every ω-string x and $t>0$ the residual of $Delay_a$ wrt x and t is $Delay_b$, where b is equal to $x(t-1)$.

Let h be a function from the real interval $[0,1)$ into Σ. Let

$$Delay^h(x)(t) = \begin{cases} h(t) & \text{if } t < 1, \\ x(t-1) & \text{otherwise.} \end{cases}$$

It is easy to see that over continuous time for every signal x and $t>1$ the residual of $Delay_a$ wrt x and t is $Delay^h$, where $h(\tau)=x(t-1+\tau)$ for $\tau\in[0,1)$.

Definition 3 (Finite memory). A function F is a finite memory function if it has finitely many distinct residuals, i.e., the set $\{Res(F,x,t)\text{: } x \textit{ is a signal},\ t\in\mathbf{R}^{\geqslant 0}\}$ is finite.

Example (*Unit delays—continued*). In the discrete time case, the set of residuals of unit delays over an alphabet Σ has the same cardinality as Σ. In the continuous time case, the set of residuals of unit delays over Boolean alphabet is uncountable.

The postulates on the input–output behavior of finite state machines are summarized as follows:

Input–output postulates: The input–output behavior of a finite state machine is a finite memory retrospective function.

2.4. Non-Zeno signals

Let C be a set of signals which satisfies the following conditions:

(1) C is closed under suffix, i.e., if $x\in C$ and $t\in\mathbf{R}^{\geqslant 0}$ then $suf(x,t)\in C$,

(2) C is closed under concatenation, i.e., if $x,y\in C$ then $x\rfloor^t;y\in C$ for each $t\in\mathbf{R}^{\geqslant 0}$.

Consider the set of functions $C\to C$, where C satisfies the above requirements. The notions introduced in the previous sections can be relativized to such set of functions.

For example we say that $F\colon C \to C$ is retrospective if whenever signals x and x' in C coincide on the interval $[0,t]$ the signals Fx and Fx' coincide on $[0,t]$.

An important set of the signals which satisfy the above requirements is the set of piecewise constant signals. In the literature, piecewise constant signals are often named non-Zeno or finite variability signals.

A signal is *piecewise constant* (or non-Zeno) if there exists an unbounded increasing ω-sequence $t_0 = 0 < t_1 < \cdots < t_n < \cdots$ such that x is constant in all subintervals (t_i, t_{i+1}).

The piecewise constant (non-Zeno) signals are physically more realistic than (general) signals. For example, the signal that has the value 0 at all irrational time moments and the value 1 at the rational time moments is not piecewise constant. The signal $ONLY5$ which receives the value 0 at the moment 5 and all other time moments has the value 1 is piecewise constant. The following requirement is physically more realistic than non-Zeno requirement and excludes the signal $ONLY5$.

Non-zero duration: A non-Zeno signal satisfies *non-zero duration* requirement if for every t there exists an interval I of a non-zero length (duration) such that $t \in I$ and x is constant in I.

Unfortunately, the set of Boolean signals satisfying the non-zero duration requirement is not closed under Boolean operations and equality test. For example, if x and y satisfy the non-zero duration requirement, then the Boolean valued signal eq defined as $eq(t) = TRUE$ iff $x(t) = y(t)$ might violate non-zero duration requirement. It is easy to see that the closure of the set of non-zero duration signals under Boolean operations coincide with the set of non-Zeno Boolean signals.

Even more restricted set of signals is the set of *right open* signals. A non-Zeno signal x is right open if for every t there exists $t' > t$ such that x is constant in $[t, t')$.

It is easy to check that both the set of right open and the set of non-Zeno signals are closed under suffix and under concatenation. These sets also include everywhere constant signals. Note also that if C is the set of non-Zeno or the set of right open signals, then C satisfies the following requirement:

$$\text{If } \rho\colon [0,\infty) \to [0,\infty) \text{ is an order preserving bijection and } x \in C \text{ then} \rho \circ x \in C.$$

It is easy to see that the only proper subsets of non-Zeno signals which are closed under concatenation, suffix, the order preserving bijections and contain all constant signals are (1) the set of right open signals, (2) the set of non-Zeno signals that have only finitely many changes and (3) the set of right open signals that have only finitely many changes. These sets are also closed under the Boolean operations.

3. Speed-independence and stability

We say that a signal x is constant at t if there are t_1, t_2 such that $t_1 < t < t_2$ and x is constant in the interval (t_1, t_2). If x is not constant at t we say that x changes at t. We say that x has left limit c at t if there exists $t' < t$ such that $x(\tau) = c$ for $\tau \in [t', t)$. The right limit is defined in a similar way. We say that a signal x is continuous from the left (right) at moment t if the left (respectively, right) limit of x at t is equal to

$x(t)$. A signal is continuous at t if it is continuous from the left and from the right at t. It is clear that a signal is continuous at t if it is constant at t. Note that according to the above terminology 0 is a singularity point, in particular no signal is continuous at 0. Note also that in the discrete time case every signal is constant at $t>0$.

Definition 4 (Stability). A function F from signals to signals is stable if for every moment $t>0$ and a signal x the following implication holds: x constant at t implies Fx constant at t.

Remark. In the discrete time case, every function is stable.

The following proposition is straightforward.

Proposition 1. *A stable function maps non-Zeno signals to non-Zeno signals.*

Definition 5 (Speed independence). A function F from signals to signals is speed-independent if for every order-preserving bijective function ρ on time $\forall x . F(\rho \circ x) = \rho \circ (Fx)$.

Hence, the stretching (along time) of an input signal for a speed-independent function F by an order-preserving bijection ρ cause the stretching of the output produced by F by the same ρ.

Remark. Note that in the classical automata theory, due to the discrete time postulates, the only order preserving bijection is the identity. Hence, every function from the discrete time signals to the discrete time signals is speed-independent.

Proposition 2. *If F is speed-independent then F is stable.*

Proof. Assume that x is constant at $t>0$ then there exists $\tau_1<t<\tau_2$ such that x is constant in (τ_1,τ_2). Let t_1 be an arbitrary point in (τ_1,τ_2). Clearly there exists an order preserving bijection $\rho_1:(\tau_1,\tau_2)\rightarrow(\tau_1,\tau_2)$ such that $\rho_1(t)=t_1$. Let ρ be the bijection on non-negative reals defined as

$$\rho(\tau) = \begin{cases} \tau & \text{if } \tau \in [0,\tau_1] \text{ or } \tau \in [\tau_2,\infty), \\ \rho_1(\tau) & \text{otherwise.} \end{cases}$$

It is clear that ρ is an order preserving bijection on non-negative reals and that $\rho \circ x = x$. Therefore, $(F(x))(t_1) = (F(\rho \circ x))(\rho^{-1}(t_1)) = (F(x))(t)$. Therefore, Fx is constant in (τ_1,τ_2). Hence, F is stable. $\square$

4. Examples

In this section we provide many examples of functions on signals and classify these concrete functions according to the properties introduced in the previous sections (see

Functions	Properties						
	Speed independent	Stable	maps non-Zeno to non-Zeno	Strong Retro	Retro	Finite memory	Countable memory
$Jump_{a\to b}$	+	+	+	+	+	+	+
$Rational$	-	-	-	+	+	-	-
$\exists$	+	+	+	-	-	+	+
$Leftcont$	+	+	+	+	+	+	+
$Cont$	+	+	+	-	-	+	+
$LLIM$	+	+	+	+	+	+	+
$RLIM$	+	+	+	-	-	+	+
Pointwise	+	+	+	-	+	+	+
$Prime$	+	+	+	+	+	-	+
$Last_a$	+	+	+	+	+	+	+
$Timer$	-	-	+	+	+	-	-
$Delay$	-	-	+	+	+	-	-
F_{10}	-	-	+	+	+	-	-
F_{11}	-	+	+	+	+	-	-

Fig. 1. Properties of the functions from examples.

Fig. 1). Some of these examples point to subtleties and warn against likely misjudgments. Examples 3–10 are from [18].

Note that we can identify signals with 0-ary functions from signals to signals. The notions defined for the functions from signals to signals are extended to signals through this correspondence. For example, we say that a signal has finite memory if it has only a finite number of distinct suffixes.

(1) Signal *Jump* is a finite memory speed-independent signal defined as

$$Jump_{a\to b}(t) = \begin{cases} a & \text{if } t = 0, \\ b & \text{if } t > 0. \end{cases}$$

(2) Signal *Rational* is speed-dependent signal defined as

$$Rational(t) = \begin{cases} True & \text{if } t \text{ is a rational number,} \\ False & \text{otherwise.} \end{cases}$$

Note that if t and t' are rational numbers then the suffixes of *Rational* at t and at t' are equal to the signal *Rational*. However if t and t' are irrational then $suf(Rational, t)$ might be distinct from $suf(Rational, t')$. It is easy to see that the signal *Rational* has uncountable (memory) number of distinct suffixes. Indeed, if $t - t'$ is irrational then $suf(Rational, t)$ is distinct from $suf(Rational, t')$.

(3) The existential quantifier (notation $\exists$) maps Boolean signals to Boolean signals and it is defined as

$$\exists(x)(t) = \begin{cases} True & \text{if there exists } t' \text{ such that } x(t') = True, \\ False & \text{otherwise.} \end{cases}$$

$\exists$ is not a retrospective, however it is speed-independent and has finite memory.

(4) The function *Leftcont* tests the continuity of signals from the left. It is defined as

$$Leftcont(x)(t) = \begin{cases} True & \text{if } x \text{ is left continuous at } t, \\ False & \text{otherwise.} \end{cases}$$

It is clear that *Leftcont* is finite memory, retrospective and speed-independent.

(5) The function *Cont* tests the continuity of the signals. It is defined as

$$Cont(x)(t) = \begin{cases} True & \text{if } x \text{ is continuous at } t, \\ False & \text{otherwise.} \end{cases}$$

Cont is not retrospective because its output at time t depends on the value of its input immediately after t, however, it is finite memory and speed-independent.

(6) Left and right limit functions map signals over Σ to signals over $\Sigma \cup \{Undef\}$, where $Undef \notin \Sigma$.

$$LLIM(x)(t) = \begin{cases} a & \text{if } \exists t' < t,\ x(t') = a \wedge (u \in (t', t) \to x(u) = a), \\ Undef & \text{otherwise.} \end{cases}$$

$$RLIM(x)(t) = \begin{cases} a & \text{if } \exists t' > t,\ x(t') = a \wedge (u \in (t, t') \to x(u) = a), \\ Undef & \text{otherwise.} \end{cases}$$

Note that both *RLIM* and *LLIM* are finite memory and speed-independent. *LLIM* is strongly retrospective, but *RLIM* is not retrospective.

(7) Let g be a function from $\Sigma_1 \times \cdots \times \Sigma_k$ into Σ. Its pointwise extension $\bar{g}$ is defined as $\bar{g}(x_1, x_2, \ldots, x_k)(t) = g(x_1(t), x_2(t), \ldots, x_k(t))$. It is clear that a pointwise function is retrospective and has only one residual.

(8) The following retrospective function is speed-independent and has countable memory.

$$Prime(x)(t) = \begin{cases} True & \text{if } x \text{ changes a prime number of times in interval } [0, t), \\ False & \text{otherwise.} \end{cases}$$

(9) Timer and Delay functions are not speed-independent and are defined as follows

$$Timer(x)(t) = \begin{cases} True & \text{if } \exists \tau < t. \text{ such that } x \text{ is constant in } [\tau, t) \text{ and } t - \tau \geqslant 1, \\ False & \text{otherwise,} \end{cases}$$

$$Delay_a(x)(t) = \begin{cases} a & \text{if } t < 1, \\ x(t-1) & \text{otherwise.} \end{cases}$$

Both these functions are unstable, however, the output of the *Timer* cannot change more than twice in any interval of length one and therefore, a non-Zeno signal is always produced on the output of *Timer*.

(10) The function $Last_a$ is a version of non-metric delay operator.

$$Last_a(x)(t) = \begin{cases} b & \text{if } \exists\tau_1\tau_2.\tau_1 < \tau_2 < t \text{ and } \forall\tau \in (\tau_1,\tau_2).x(\tau) = b \text{ and} \\ & \quad x \text{ changes at every point in } (\tau_2, t), \\ \mathrm{a} & \text{otherwise.} \end{cases}$$

This function is retrospective finite memory and speed-independent.

(11) Our last example is two functions F_{10} and F_{11}. Both these functions are stable, however they are not speed-independent. The output of F_{11} is always non-Zeno. F_{10} maps non-Zeno signals to non-Zeno signals, however, it can map Zeno signal to Zeno signal.

$$F_{10}(x)(t) = \begin{cases} True & \text{if } x \text{ changes a finite number of times in } [0,t) \text{ or} \\ & \quad \text{if } t \text{ is rational,} \\ False & \text{otherwise,} \end{cases}$$

$$F_{11}(x)(t) = \begin{cases} True & \text{if there is irrational } t_0 \leqslant t \text{ such that } x \text{ is constant} \\ & \quad \text{in } [0,t_0) \text{ and } x(t_0) \neq x(0), \\ False & \text{otherwise.} \end{cases}$$

Note that if t is rational and u maps $[0,t)$ to $\{True, False\}$ then the residual of F_{11} wrt u and t is either F_{11} or the constant operator that outputs *False*. Hence F_{11} has only two distinct residuals wrt function over the rational length intervals. Nevertheless, it is easy to see that F_{11} has an uncountable number of distinct residuals. In [10], a retrospective function which has countable memory and is not speed-independent was constructed.

5. Speed-independent functions over right open and non-Zeno signals

In this section descriptions of speed-independent functions over right open signals and speed-independent functions over non-Zeno signals are provided. We will show that such functions can be faithfully represented by functions over ω-strings.

Recall that a Σ-signal x is right open if there exist an ω-sequence $\alpha = \langle a_i\colon\ i \in \mathbf{N}\rangle$ over Σ and an unbounded increasing ω-sequence $\tau = \langle t_i\colon\ i \in \mathbf{N}\rangle$ of reals such that $t_0 = 0$ and

$$\forall i \forall t \in [t_i, t_{i+1}).x(t) = a_i.$$

If the above conditions hold we say that (the pair) α, τ *characterizes* x or x *is characterized by* α, τ (see Fig. 2).

Terminology and notations: An unbounded increasing sequence $t_0 < t_1 < \cdots$ of reals with $t_0 = 0$ is called time scale. Throughout this section letters τ, τ' range over time

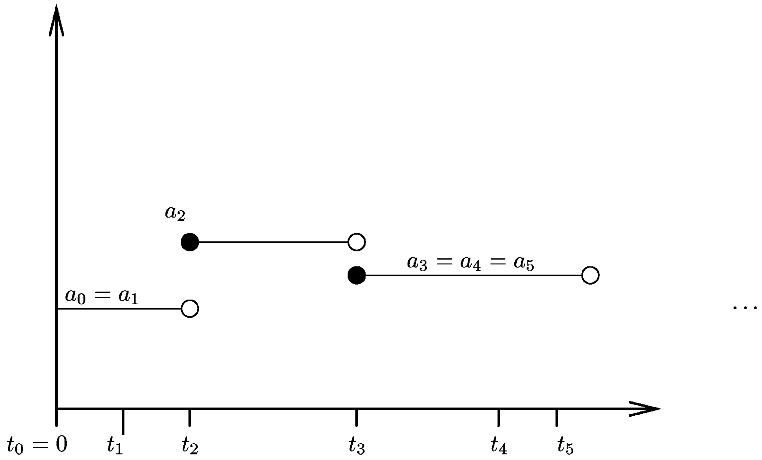

Fig. 2. A right open signal characterized by $\langle a_0, a_1, \ldots \rangle, \langle t_0, t_1, \ldots \rangle$.

scales. For a time scale $\tau = \langle t_0, t_1, \ldots, t_i, \ldots \rangle$ we sometimes use $\tau(i)$ to denote t_i. Letters α, β denote ω-sequences (ω-strings) over an alphabet Σ. We use Σ^ω for the set of all ω-strings over the alphabet Σ.

Assume that ρ is an order preserving bijection on non-negative reals. Let τ be a time scale and let $\tau'(i) = \rho(\tau(i))$ for all natural i. Then τ' is a time scale; moreover, α, τ characterizes x if and only if α, τ' characterizes $\rho \circ x$. It is clear that for every time scales τ and τ' there exists an order preserving bijection ρ such that α, τ characterizes x if and only if α, τ' characterizes $\rho \circ x$.

Note that (1) if x is characterized by α, τ and x is not constant at t then t appears in τ and (2) if τ contains all points at which x is not constant then there exists α such that α, τ characterizes x. Hence, if F is stable function from right open signals to right open signals and α, τ characterizes x then there exists β such that β, τ characterizes Fx.

Let F be a speed-independent function from right open signals to right open signals. By Proposition 2, F is stable. Assume that α, τ characterizes x and let β be such that β, τ characterizes $y = Fx$ (such β exists by (2) above). Since F is speed-independent, it follows that for any τ' and for the x' characterized by α, τ' the signal Fx' is characterized by β, τ'.

Hence, with every speed-independent function F one can associate a function G from ω-strings to ω-strings such that

$$\forall \alpha \forall \tau. \text{ if } \alpha, \tau \text{ characterizes } x \text{ then } G\alpha, \tau \text{ characterizes } Fx$$

Such G is said to be a (discrete) *characterization* of F.

Not every G on ω-strings characterizes a function on right open signals. Indeed, if G characterizes a function then whenever α, τ and α', τ' characterize the same signal then $G\alpha, \tau$ and $G\alpha', \tau'$ should also characterize the same signal. Many distinct α, τ may characterize the same signal. For example, assume that $\alpha = \langle a_0, \ldots, a_i, a_{i+1} \ldots \rangle$ and $\tau = \langle t_0, \ldots, t_i,\ t_{i+1} \ldots \rangle$. Let $t \in (t_i, t_{i+1})$ and let α' and τ' be defined as $\langle a_0, \ldots, a_i, a_i,$

$a_{i+1}\dots\rangle$ and $\langle t_0,\dots,t_i,t,t_{i+1}\dots\rangle$ respectively. Then α,τ characterize x if and only if α',τ' characterize x. Therefore, if G characterizes a function on right open signals it should satisfy the following:

SI condition: For any $\langle a_0,\dots,a_i,a_{i+1}\dots\rangle$ and $\langle b_0,\dots,b_i,b_{i+1}\dots\rangle$

$$G(\langle a_0,\dots,a_i,a_{i+1}\dots\rangle)=\langle b_0,\dots,b_i,b_{i+1}\dots\rangle$$

if and only if

$$G(\langle a_0,\dots,a_i,a_i,a_{i+1}\dots\rangle)=\langle b_0,\dots,b_i,b_i,\ b_{i+1}\dots\rangle.$$

In Appendix A, it is shown that if a function G on ω-strings satisfies SI condition then there exists a speed-independent F on right open signals such that G characterizes F. Note that if F is characterized by G then F is retrospective iff G is retrospective and F and G have the same number of distinct residuals. These observations are summarized in

Proposition 3 (Characterization of speed-independent functions on right open signals).
(1) *Every speed-independent function F from right open signals to right open signals is characterized by a function G that satisfies SI condition.*
(2) *Every function G that satisfies SI condition characterizes a speed-independent function F from right open signals to right open signals.*
(3) *If G characterizes a function F from right open signals to right open signals then*
   (a) *G is retrospective iff F is retrospective.*
   (b) *G and F have the same number of distinct residuals and hence*,
   (c) *G has finite memory iff F has finite memory.*

Since every retrospective function on ω-strings has at most countable memory (i.e., countable number of distinct residuals) we obtain

Corollary 4. *Every speed-independent retrospective function on right open signals has at most countable memory.*

Below we provide a similar description for speed-independent functions over non-Zeno signals.

A non-Zeno signal x over an alphabet Σ (see Fig. 3) is said to be characterized by α,α',τ if (1) $\alpha=\langle a_i\colon\ i\in\mathbf{N}\rangle$ and $\alpha'=\langle a'_i\colon\ i\in\mathbf{N}\rangle$ are ω-strings over Σ, (2) $\tau=\langle t_i\colon\ i\in\mathbf{N}\rangle$ is a time scale and (3) $x(t_i)=a_i$ and $x(t)=a'_i$ for every i and every $t\in(t_i,t_{i+1})$.

Observe that for every non-Zeno signal x there exists a triple α,α',τ that characterizes x and that every α,α',τ characterizes a non-Zeno signal.

A function F from non-Zeno signals over Σ_1 to non-Zeno signals over Σ_2 is said to be characterized by a function $G\colon(\Sigma_1^\omega\times\Sigma_1^\omega)\to(\Sigma_2^\omega\times\Sigma_2^\omega)$ if whenever α,α',τ characterize x then $G(\alpha,\alpha'),\tau$ characterize Fx.

Every speed-independent function is characterized by a function on ω-strings. However, not every function $G\colon(\Sigma_1^\omega\times\Sigma_1^\omega)\to(\Sigma_2^\omega\times\Sigma_2^\omega)$ characterizes a speed-

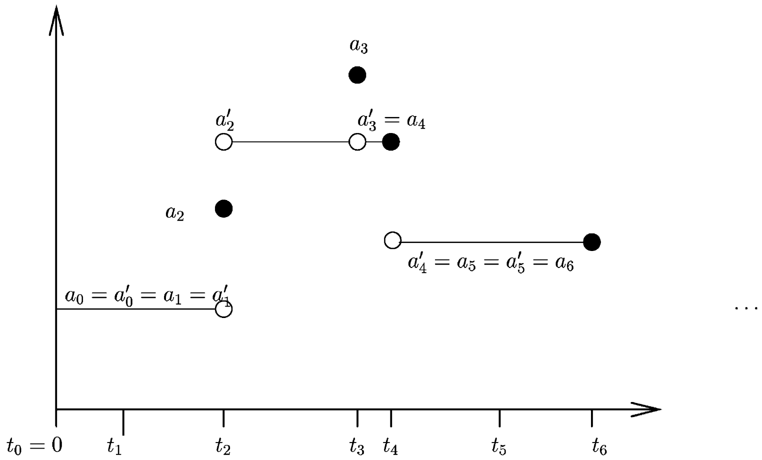

Fig. 3. A non-Zeno signal characterized by $\langle a_0, a_1, \ldots \rangle, \langle a'_0, a'_1, \ldots \rangle, \langle t_0, t_1, \ldots \rangle$.

independent function. In order to describe the functions on ω-strings that characterize speed-independent functions on non-Zeno signals, it is useful to define insertion operation on ω-sequences. We say that ω-sequence γ' is obtained from an ω-sequence γ by inserting c after a position i if (1) $\gamma'(k) = \gamma(k)$ for $k \leqslant i$, (2) $\gamma'(i+1) = c$ and (3) $\gamma'(k) = \gamma(k-1)$ for $k > i+1$. Hence, the insertion of c into $\gamma = \langle c_0, \ldots, c_i, c_{i+1}, \ldots \rangle$ after i is the ω-sequence $\langle c_0, \ldots, c_i, c, c_{i+1}, \ldots \rangle$.

Let α_1, α'_1 be ω strings, τ_1 be a time scale and let a'_i be equal to $\alpha'_1(i)$. Assume that (1) α_2, and α'_2 are obtained from α_1 and α'_1 by inserting a'_i after i and (2) τ_2 is obtained from τ by inserting any t from the interval $(\tau_1(i), \tau_1(i+1))$ after i. Then $\alpha_1, \alpha'_1, \tau_1$ characterize x iff $\alpha_2, \alpha'_2, \tau_2$ characterize x.

Hence, if G characterizes a speed-independent function it should satisfy the following *Generalized SI conditions*: Let α_1 α'_1 be ω-strings and let i be a natural number; let α_2 and α'_2 be obtained from α_1 and α'_1 by inserting $\alpha'_1(i)$ after i. Similarly, let β_2 and β'_2 be obtained from β_1 and β'_1 by inserting $\beta'_1(i)$ after i. Then for every $\alpha_j, \alpha'_j, \beta_j, \beta'_j$ as above $(j = 1, 2)$

$$G(\alpha_1, \alpha'_1) = (\beta_1, \beta'_1)$$

if and only if

$$G(\alpha_2, \alpha'_2) = (\beta_2, \beta'_2).$$

If G satisfies generalized SI condition then G characterizes a speed-independent function (the proof is similar to the proof of Proposition 3, given in Appendix A).

Assume that $\alpha_1, \alpha'_1, \tau$ characterize x_1 and $\alpha_2, \alpha'_2, \tau$ characterize x_2. Then x_1 is equal to x_2 in $[0, t]$ if either (1) $t \in (\tau(i), \tau(i+1))$ and $\alpha_1 = \alpha_2$ in $[0, i]$ and $\alpha'_1 = \alpha'_2$ in $[0, i]$ or (2) $t = \tau(i)$ and $\alpha_1 = \alpha_2$ in $[0, i]$ and $\alpha'_1 = \alpha'_2$ in $[0, i-1]$. Hence, if G characterizes a retrospective function F then G should satisfy the following

Generalized retrospective conditions:
(1) G is retrospective, i.e. if $G(\alpha_1,\alpha_1')=(\beta_1,\beta_1')$, $G(\alpha_2,\alpha_2')=(\beta_2,\beta_2')$, $\alpha_1=\alpha_2$ in $[0,i]$ and $\alpha_1'=\alpha_2'$ in $[0,i]$ then $\beta_1=\beta_2$ in $[0,i]$ and $\beta_1'=\beta_2'$ in $[0,i]$.
(2) if $G(\alpha_1,\alpha_1')=(\beta_1,\beta_1')$, $G(\alpha_2,\alpha_2')=(\beta_2,\beta_2')$, $\alpha_1=\alpha_2$ in $[0,i]$ and $\alpha_1'=\alpha_2'$ in $[0,i-1]$ then $\beta_1=\beta_2$ in $[0,i]$. (Note that condition 1 implies that $\beta_1'=\beta_2'$ in $[0,i-1]$.)

Actually, this condition is sufficient to ensure that the function F characterized by G is retrospective.

Finally, observe that if G characterizes F then F has finite (respectively, countable) memory if and only if G has finite (respectively countable) memory. The following proposition summarizes all these observations.

Proposition 5 (Characterization of speed-independent functions on non-Zeno signals).
(1) *Every speed-independent function F from non-Zeno signals to non-Zeno signals is characterized by a function G that satisfies generalized SI condition.*
(2) *Every function G that satisfies generalized SI condition characterizes a speed-independent function F from non-Zeno signals to non-Zeno signals.*
(3) *If G characterizes a function F from non-Zeno signals to non-Zeno signals then*
  (a) *F is retrospective if and only if G satisfies generalized retrospective condition.*
  (b) *F has finite memory iff G has finite memory.*
  (c) *F has countable memory iff G has countable memory.*

Since every function G on ω-strings has at most countable memory, we obtain

Corollary 6 (Trakhtenbrot [18]). *Every speed-independent function on non-Zeno signals has at most countable memory.*

In Section 8 we will show (Theorem 21) that every finite memory retrospective function on non-Zeno (right open) signals is speed-independent. Therefore, it can be characterized by a function on ω-strings. However, in order to prove this property of functions on non-Zeno signals we have to investigate in Sections 6 and 7 functions on general signals.

6. Closure properties of functions on signals

The next proposition follows from the definitions.

Proposition 7. *The following sets of functions on signals are closed under taking residual and are closed under composition.*
(1) *The set of retrospective functions.*
(2) *The set of strongly retrospective functions.*
(3) *The set of stable functions.*
(4) *The set of speed independent functions.*
(5) *The set of finite memory retrospective functions.*

Proof.

Closure under taking residual: We only show (A.5), i.e., a residual of a finite memory retrospective function is a finite memory retrospective function. The proofs of (1)–(4) are left to the reader.

Assume that G is a residual of a finite memory retrospective function F wrt x and t. Then the residual of G wrt y and τ is the residual of F wrt $x\rfloor^{t}; y$ and $t+\tau$. Hence the set of residuals of G is a subset of the set of residuals of F, and hence it is finite.

Closure under composition:

(1) Let us show that if $F: Sig(\Sigma_1) \rightarrow Sig(\Sigma_2)$ and $G: Sig(\Sigma_2) \rightarrow Sig(\Sigma_3)$ are retrospective functions, then their composition $F \circ G$ is retrospective.

Assume that x and x' coincide in $[0,t]$. Then Fx and Fx' coincide in $[0,t]$ because F is retrospective. Therefore, $G(Fx)$ and $G(Fx')$ coincide in $[0,t]$ because G is retrospective. This shows that $F \circ G$ is retrospective.

We omit the proofs for (2)–(4); they are similar to the proof of (1).

In order to show (A.5), observe that if F_0 is the residual of F wrt x and t and G_0 is the residual of G wrt Fx and t then $F_0 \circ G_0$ is the residual of $F \circ G$ wrt x and t.

From this observation it follows that if n_1 (respectively, n_2) is the number of distinct F residuals (respectively, G residuals) then the number of distinct $F \circ G$ residuals is less than or equal to $n_1 \times n_2$. $\square$

Remark (*Relativizing results*). All the results from this section hold when we replace everywhere 'functions over signals' by 'functions from non-Zeno signals to non-Zeno signals' or by 'functions from right open signals to right open signals'

Analyzing the proofs of this section one can check that they hold for the set of functions over any set C of signals which is closed under concatenation, suffix and the order preserving bijections. In particular, the following meta-theorem holds:

Proposition 8. *Let C be a set of signals which is closed under concatenation, suffix and the order preserving bijections. The following sets of functions over C are closed under taking residual and are closed under composition.*
(1) *The set of retrospective functions over C.*
(2) *The set of strongly retrospective functions over C.*
(3) *The set of stable functions over C.*
(4) *The set of speed-independent functions over C.*
(5) *The set of finite residual retrospective functions over C.*

7. Properties of finite memory retrospective functions

In this section we investigate properties of finite memory functions on general signals. We deal with functions on general signals not only for the sake of generality. The representation of finite memory functions on non-Zeno signals (see Section 8) will rely on the results about functions on general signals.

7.1. Finite memory signals

The following proposition is the key technical proposition which is needed for the finite representation of the finite memory retrospective functions on non-Zeno signals.

Proposition 9. *A (general) signal x is finite memory if and only if x is constant on the positive reals.*

Proof. See Appendix B.

Remark (*Contrast with discrete case*). Note that in a discrete time case a signal is an ω-sequence of states. Such a signal x is finite memory iff it is quasiperiodic, i.e., $x = uv^{\omega}$.

Remark. Note that a signal is speed-independent iff it is constant on the positive reals. In the discrete case every signal is speed-independent.

Remark. Proposition 9 is easily proved for the non-Zeno and for the right open signals. However, even if we want to deal with functions over non-Zeno signals many of our proofs will be based on this proposition which deals with (general) finite memory signals.

7.2. Some consequences of Proposition 9

Recall that $Jump_{c\to d}$ is the signal that has value c at 0 and value d at $t>0$.

Proposition 10. *If F is a finite memory retrospective function then $F(Jump_{a\to b}) = Jump_{c\to d}$ for some c and d.*

Proof. F is a finite memory retrospective function and $Jump_{a\to b}$ is a finite memory signal, therefore, $F(Jump_{a\to b})$ is a finite memory signal, by Proposition 7(5), and therefore, by Proposition 9, it is constant on the positive reals, hence it has the form $Jump_{c\to d}$. $\square$

Proposition 11. *Every finite memory retrospective function is stable.*

Proof. Assume that F is a finite memory retrospective function. We have to show that if x is constant at $t>0$ then Fx is constant at t.

Assume that x is constant at $t>0$. Then there exists $\varepsilon>0$ such that

$$x(\tau) = x(t) = b \quad \text{for } \tau \in [t-\varepsilon, t+\varepsilon]. \tag{1}$$

Let G be the residual of F wrt x and $t-\varepsilon$. From (1) it follows that $G(Const_b)\tau = Fx(t-\varepsilon+\tau)$, where $\tau \in [t-\varepsilon, t+\varepsilon]$ and $Const_b$ is the signal which is equal to b everywhere.

Therefore,

$$Fx \text{ is constant at } t \text{ if and only if } G(Const_b) \text{ is constant at } \varepsilon. \tag{2}$$

Since G is a residual of F, it is a finite memory retrospective function, by Proposition 7. Therefore, by Proposition 9, the signal $G(Const_b)$ is constant on the positive reals and therefore it is constant at ε. Hence, by (2), the signal Fx is constant at t. □

Note that Propositions 11 and 1 imply

Corollary 12. *A finite memory retrospective function maps non-Zeno signals to non-Zeno signals.*

The *restriction* of F to non-Zeno signals is a function $Rest(F)$ defined as $Rest(F) = \lambda x \in \text{non-Zeno}.Fx$. Note that $Rest(F)$ might map a non-Zeno signal to a general signal. However, from Corollary 12 we obtain

Proposition 13. *If F is a finite memory retrospective function on signals then $Rest(F)$ is a function from non-Zeno signals to non-Zeno signals. Moreover, $Rest(F)$ is a finite memory retrospective function over the non-Zeno signals.*

Proof. It is clear that if F is a finite memory retrospective function then $Rest(F)$ is retrospective wrt non-Zeno signals and $Rest(F)$ has a finite number of distinct residuals wrt non-Zeno signals. The rest follows from Corollary 12. □

7.3. State function

Definition 6 (State function). Let G_0 be a finite memory retrospective function from $Sig(\Sigma)$ to $Sig(\Sigma')$ and let $\vec{G} = \langle G_0, G_1, \ldots, G_n \rangle$ be a sequence of all its residuals. It is clear that any residual of G_i is a residual of G_0. Define functions $out_{\vec{G}} : \Sigma \times \{0, \ldots, n\} \to \Sigma'$ and $state_{\vec{G}}$ from $Sig(\Sigma)$ to $Sig(\{0, \ldots, n\} \to \{0, \ldots, n\})$ as follows:

$$out_{\vec{G}}(a, i) = G_i cons_a 0, \text{ where } cons_a \text{ is the constant signal } \lambda t.a$$

$$\big(state_{\vec{G}}(x)\big)(t)i = j \text{ if } G_j \text{ is the residual of } G_i \text{ wrt } x \text{ and } t.$$

From the definition it follows

Proposition 14 (Properties of the state function). (1) *$state_{\vec{G}}(x)(0) = id$—the identity permutation.*

(2) *$state_{\vec{G}}$ is a strongly retrospective function.*
(3) *$state_{\vec{G}}(x_1\rfloor^{t_1}; x_2)(t_1 + t_2) = (state_{\vec{G}} x_1 t_1) \circ (state_{\vec{G}} x_2 t_2)$.*
(4) *$G_0 xt = out_{\vec{G}}(x(t), state_{\vec{G}} xt0)$.*

Remark. Actually for the above proposition there is no need in the assumption that G has a finite number of residuals.

Proposition 14 implies

Proposition 15. *Let G_0 be a finite memory retrospective function from $Sig(\Sigma)$ to $Sig(\Sigma')$ and let $\vec{G} = \langle G_0, G_1, \ldots, G_n \rangle$ be a sequence of all its residuals. Let $state_{\vec{G}}$ be defined as in Definition 6. Then $state_{\vec{G}}$ is a finite memory strongly retrospective functions on signals. Moreover, there exists $\delta_{\vec{G}} : (\Sigma \times \Sigma) \to (\{0, \ldots, n\} \to \{0, \ldots, n\})$ such that*
(1) *$\delta_{\vec{G}}(a, b) = state_{\vec{G}}\, Jump_{a \to b} t$ for every $t > 0$.*
(2) *$\delta(a, b) = \delta(a, b) \circ \delta(b, b)$ for any $a, b \in \Sigma$.*

Proof. From Proposition 14(2) it follows that $state_{\vec{G}}$ is strongly retrospective and from Proposition 14(3) we obtain that the number of the residuals of $state_{\vec{G}}$ is bounded by the number of the functions from $\{0, \ldots, n\}$ to $\{0, \ldots, n\}$.

Proposition 10 implies that $state_{\vec{G}}\, Jump_{a \to b}$ is constant on the positive real, hence $\delta_{\vec{G}}(a, b)$ can be defined as the value of $state_{\vec{G}}\, Jump_{a \to b}$ at any positive real and Proposition 15(1) holds.

Finally, note that

$$Jump_{a \to b} = Jump_{a \to b}\rfloor^{t/2}; \quad Jump_{b \to b} \text{ for any } t > 0 \tag{3}$$

Therefore,

$$\begin{aligned}
\delta(a, b) &= state_{\vec{G}}\, Jump_{a \to b} t, \text{ by Proposition 15(1)} \\
&= (state_{\vec{G}}\, Jump_{a \to b}(t/2)) \circ (state_{\vec{G}}\, Jump_{a \to b}(t/2)), \text{ by Proposition 14(3)} \\
&\quad \text{and Eq. (3)} \\
&= \delta(a, b) \circ \delta(b, b), \text{ by Proposition 15(1).} \qquad \square
\end{aligned}$$

7.3.1. Relativizing to functions over non-Zeno signals

Let $G = G_0$ be a function from non-Zeno signals over Σ to non-Zeno signals over Σ' which is retrospective and finite memory. Let $\vec{G} = \langle G_0, \ldots G_n \rangle$ be a sequence of all its residuals. The state functions $state_{\vec{G}}$ is defined exactly as in Definition 6.

Theorem 16. *The function $state_{\vec{G}}$ maps non-Zeno signals to non-Zeno signals. Moreover, Propositions 14 and 15 hold whenever all notions are relativized to non-Zeno signals. In particular, there exist $\delta : \Sigma \times \Sigma \to (\{0, \ldots, n\} \to \{0, \ldots, n\})$ and $out : \Sigma \times \{0, \ldots, n\} \to \{0, \ldots, n\}$ such that*
(1) *$\delta(a, b) \circ \delta(b, b) = \delta(a, b)$.*
(2) *$Gxt = out(x(t), state_{\vec{G}} xt0)$.*
(3) *$state_{\vec{G}}$ is strongly retrospective function from non-Zeno signals over Σ to non-Zeno signals over $(\{0, \ldots, n\} \to \{0, \ldots, n\})$.*
(4) *$state_{\vec{G}} x0 = id$—the identity permutation.*
(5) *$(state_{\vec{G}}\, Jump_{a \to b})(t) = \delta(a, b)$ for every $t > 0$.*
(6) *$state_{\vec{G}}(Jump_{a \to b}\rfloor^{t'}; x)(t' + t) = \delta(a, b) \circ (state_{\vec{G}} xt)$.*

Proof. Consider the extensions $Ext(state_{\vec{G}})$ of $state_{\vec{G}}$ to all signals defined as

$$Ext(state_{\vec{G}})(x)(\tau) = \begin{cases} state_{\vec{G}}x(\tau) & \text{if } x \text{ changes a finite number of times} \\ & \quad \text{in } [0,\tau) \text{ and} \\ Undef & \text{otherwise.} \end{cases}$$

(Here, *Undef* is any symbol not in $\{0,\dots,n\}$.)

The function $Ext(state_{\vec{G}})$ is a retrospective finite memory function. Moreover, $state_{\vec{G}}x = Ext(state_{\vec{G}})x$ for any non-Zeno signal x. Therefore, all the equations from Propositions 14 and 15 hold when x is restricted to the non-Zeno signals. Moreover, since $state_{\vec{G}}$ is the restriction of $Ext(state_{\vec{G}})$ to non-Zeno signal by Proposition 13 we obtain that $state_{\vec{G}}$ is a function from non-Zeno signals to non-Zeno signals. □

Remark. Note that Theorem 16 deals with functions over non-Zeno signals. However, the proof of this theorem relies on the consequences of Proposition 9 (namely, on Propositions 13 and 15) which deal with (general) signals. Hence, Proposition 9 plays a crucial role in our proof.

Motivated by Theorem 16(3–6) we introduce the following

Definition 7 (Definability). Let Σ and Q be finite sets and let $\delta:(\Sigma\times\Sigma)\to(Q\to Q)$. A function F from non-Zeno signals over Σ to non-Zeno signals over $Q\to Q$ is *definable* by δ if it satisfies the following conditions:
(1) F is a strongly retrospective function.
(2) $Fx0 = id_Q$.
(3) $F\,Jump_{a\to b}\,t = \delta(a,b)$ for every $t>0$ and $a,b\in\Sigma$.
(4) $F(Jump_{a\to b}\rfloor^{t'};x)(t'+t) = \delta(a,b)\circ Fxt$.

Proposition 17. *Let δ be a function in $\Sigma\times\Sigma\to(Q\to Q)$.* Then there exists at most one function definable by δ.

Proof. We have to show that If F_1 and F_2 are definable by δ then $F_1 = F_2$.

Let x be a non-Zeno signal and let t be a real number. Then x changes a finite number n of times in $(0,t)$. Therefore, there are sequences $t_0 = 0 < t_1 < \cdots < t_{n+1} = t$ and $a_0\dots a_n$ and $b_0\dots b_n$ such that
(1) $x(t_i) = a_i$ for $i\leqslant n$.
(2) $x(u) = b_i$ for $u\in(t_i,t_{i+1})$ and $i\leqslant n$.
(3) $b_i\neq a_{i+1}$ or $a_{i+1}\neq b_{i+1}$ for $0\leqslant i<n$.
By the induction of the number of changes of a signal x in $(0,t)$ we show that $F_1xt = F_2xt$.

Basis: (x does not change in $(0,t)$.) If $t=0$ then $F_1xt = F_2xt$ by condition (2) of Definition 7. If $t>0$ and x is constant in $(0,t)$ then $F_ixt = F_iJump_{a_0\to b_0}t$ by the strongly retrospectivity of F_i and therefore $F_1xt = F_2xt = \delta(a_0,b_0)$ by condition (3) of Definition 7.

Inductive step: Assume that $\forall x \forall t F_1xt = F_2xt$ whenever x changes in $(0,t)$ at most n times. Let x be a non-Zeno signal and assume that x changes $n+1$ times in $(0,t)$. Let $t_1>0$ be the first changes of x in $(0,t)$ and let $x_1 = suf(x,t_1)$. Observe that x_1 changes at most n times in $(0, t-t_1)$ and $x = Jump_{a\to b}\rfloor^{t_1}; x_1$ for some $a,b \in \Sigma$. By condition (4) of Definition 7.

$$F_ixt = \delta(a,b) \circ F_ix_1(t-t_1). \tag{4}$$

By the inductive hypothesis $F_1x_1(t-t_1) = F_2x_1(t-t_1)$. Therefore, $F_1xt = F_2xt$, by (4). This completes the inductive step. □

Remark. If the requirement that a function is over non-Zeno signals is dropped from the definition of definability, then the conclusion of Proposition 17 will be that all functions definable by δ coincide on non-Zeno signals.

Proposition 18. *If $\delta(a,b)\circ\delta(b,b) = \delta(a,b)$ then there exists a finite memory speed-independent function definable by δ.*

Proof. For every non-Zeno signal x and every $t>0$ there exist sequences $t_0 = 0 < t_1 \cdots < t_{n+1} = t$, $a_0, \ldots a_n$ and $b_0 \ldots b_n$ such that
(1) $x(t_i) = a_i$ for $i \leqslant n$.
(2) $x(u) = b_i$ for $u \in (t_i, t_{i+1})$ and $i \leqslant n$.
(3) $b_i \neq a_{i+1}$ or $a_{i+1} \neq b_{i+1}$ for $0 \leqslant i < n$.
Define F and a sequence of $\rho_i \in (Q \to Q)$ as follows:

$$\rho_0 = \delta(a_0, b_0),$$

$$\rho_{i+1} = \rho_i \circ \delta(a_{i+1}, b_{i+1}),$$

$$Fx0 = id\text{—the identity permutation},$$

$$Fxu = \rho_i \text{ for } u \in (t_i, t_{i+1}].$$

It is immediate that F satisfies conditions (1)–(3) of Definition 7. F satisfies condition (4) of Definition 7 because $\delta(a,b)\circ\delta(b,b) = \delta(a,b)$. The speed independence of F follows immediately from its definition. F is a finite memory function because the number of its residuals is bounded by the number of functions from Q to Q. □

From Proposition 18, and from Proposition 17 we obtain the following corollary.

Corollary 19. *If $\delta(a,b)\circ\delta(b,b) = \delta(a,b)$ then there exists a unique function definable by δ. Moreover, the function definable by δ is finite memory strongly retrospective and speed-independent.*

Remark (*Failure of the relativization to right open signals*). Recall that a signal x is right open if it is non-Zeno and for every t there exist $t' > t$ such that x is constant in $[t, t')$.

Even if F is a finite memory retrospective function from right open signals to right open signals its corresponding state function might map a right open signal into a not right open signal. The following example illustrates this:

Example. Let F_0 and F_1 be two functions over right open signals defined as follows

$$(F_i x)t = \begin{cases} i & \text{if } x \text{ is constant in } [0,t], \\ \text{left limit of } x \text{ at } \tau & \text{if } x \text{ changes at } \tau \text{ and } x \text{ is constant in } [\tau,t]. \end{cases}$$

It is easy to see that F_1 is a residual of F_0 and *state* maps a constant signal $\lambda\tau.1$ to a signal that is not right open.

Observe also that the constant functions are the only strongly retrospective functions over the right open signals.

8. Representation of finite memory retrospective function

In the first subsection, a set of labeled transition diagrams which is called a finite state transducer is defined. Every finite state transducer describes (computes) a finite memory retrospective function over non-Zeno signals. We show that the inverse also holds, namely, every finite memory retrospective function is computable by a finite state transducer. In this sense, the finite state transducers provide a finite description for the set of finite memory retrospective functions over non-Zeno signals. The result of the second subsection implies that it is impossible to find finite descriptions for all finite memory retrospective functions over (general) signals because the number of such function is at least uncountable.

8.1. Finite state transducers over non-Zeno signals

Definition 8. A finite state transducer over non-Zeno signal has the following components:
- A finite set of states Q,
- An initial state $q_0 \in Q$,
- An input alphabet Σ_{in} and output alphabet Σ_{out},
- An output function $out: Q \times \Sigma_{in} \to \Sigma_{out}$ and
- A transition function $\delta: \Sigma_{in} \times \Sigma_{in} \to (Q \to Q)$ such that $\delta(a,b) \circ \delta(b,b) = \delta(a,b)$.

It is convenient to use a graphical representation for transducers. On the picture, the states will be represented by nodes and the functions δ and *out* will be represented by labels on the arcs and the nodes of the graph (see Fig. 4). The initial state will be indicated by $\Rightarrow$.

If $\delta(a,b)q = q'$ we will draw an arc labeled by $\langle a,b\rangle$ from q to q'; note that in this case $\delta(b,b)q'$ should be equal to q', therefore in such case we can abbreviate the graph by dropping the arc $\langle b,b\rangle$ from q' to q'.

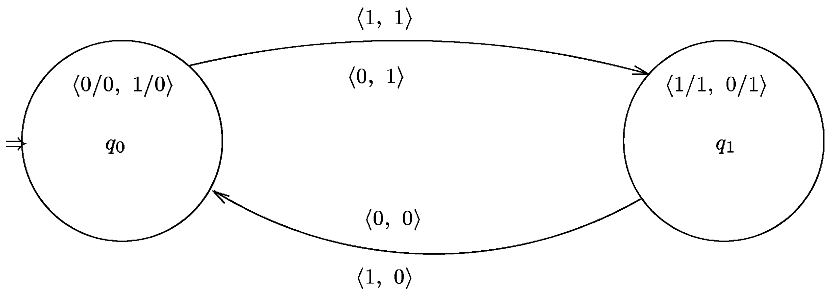

Fig. 4. Transducer for left limit.

Note that for every q the function $\lambda a.out(q,a)$ maps Σ_{in} to Σ_{out}, Therefore, we can represent out by labeling the nodes; we will label q by $\langle a_1/b_1,\dots,a_n/b_n\rangle$ if $out(q,a_i)=b_i$.

Definition 9 (The function computable by a transducer). Let $\mathscr{A}=\langle Q,q_0,\Sigma_{in},\Sigma_{out}, out,\delta\rangle$ be a transducer. Note that by Proposition 17 there exists a unique function F^δ definable by δ. The function $fun_{\mathscr{A}}$ *computable* by $\mathscr{A}$ is defined as $fun_{\mathscr{A}}xt = out(F^\delta xtq_0,xt)$.

Example. In Fig. 4 a transducer is presented. The function F computable by this transducer is defined as follows: $y=F(x)$ if $y(0)=0$ and if $t>0$ then $y(t)$ is the left limit of x at t (i.e., $y(t)=a$ iff there is $\varepsilon>0$ such that y is equal to a in the interval $[t-\varepsilon,t)$).

Theorem 20. *A function over non-Zeno signals is a finite memory retrospective function if and only if it is computable by a transducer.*

Proof. Let $\mathscr{A}=\langle Q,q_0,\Sigma_{in},\Sigma_{out},out,\delta\rangle$ be a transducer. Note that F^δ is a finite memory retrospective function. Therefore, the function $fun_{\mathscr{A}}$ is a finite memory because it is defined as the composition of the pointwise functions out and F^δ (F^δ is finite memory by Corollary 19).

The other direction follows from Theorem 16. □

Note that by Corollary 19, the function F^δ definable by δ, is speed-independent. Hence, the function computable by a transducer is speed-independent. Therefore, Theorem 20 implies

Theorem 21. *Every finite memory retrospective function over non-Zeno signals is speed-independent.*

We will conclude this subsection by providing a description of the function computable by a transducer in terms of ω-languages. Let $\mathscr{A}=\langle Q,q_0,\Sigma_{in},\Sigma_{out},out,\delta\rangle$ be a transducer. Consider $\Delta\subset Q\times\Sigma_{in}\times\Sigma_{in}\times\Sigma_{out}\times\Sigma_{out}\times Q$ defined as follows: $\langle q,a,a',b, b',q'\rangle\in\Delta$ iff (1) $q'=\delta(a,a')q$ and (2) $b=out(q_1,a)$ and $b'=out(q',a')$.

Let $L_{\mathscr{A}} \subset (\Sigma_{in} \times \Sigma_{in} \times \Sigma_{out} \times \Sigma_{out})^{\omega}$ be the set of ω-strings defined as:

$$\langle a_0, a_0', b_0, b_0' \rangle \langle a_1, a_1', b_1, b_1' \rangle \cdots \langle a_i, a_i', b_i, b_i' \rangle \ldots \in L_{\mathscr{A}}$$

iff there exist $q_1, q_2, \ldots q_n \ldots \in Q$ such that $\langle q_i, a_i, a_i', b_i, b_i', q_{i+1} \rangle \in \Delta$

(for $i = 0, 1 \ldots$).

Let $L_{\mathscr{A}}^{sig} \subset Sig(\Sigma_{in}) \times Sig(\Sigma_{out})$ be the set of pairs of non-Zeno signals defined as: $\langle x, y \rangle \in L_{\mathscr{A}}^{sig}$ iff there exists an increasing divergent ω-sequence $0 = t_0 < t_1 < \cdots t_n \cdots$ of reals and an ω-string $\langle a_0, a_0', b_0, b_0' \rangle \langle a_1, a_1', b_1, b_1' \rangle \cdots \langle a_i, a_i', b_i, b_i' \rangle \ldots$ in $L_{\mathscr{A}}$ such that $x(t_i) = a_i$, $y(t_i) = b_i$ and $\forall t \in (t_i, t_{i+1}).x(t) = a_i' \wedge y(t) = b_i'$.

From the proof of Proposition 17 and Definition 8, it follows that for every non-Zeno signal x there exists a unique y such that $\langle x, y \rangle \in L_{\mathscr{A}}^{sig}$, moreover $L_{\mathscr{A}}^{sig}$ is the graph of the function $fun_{\mathscr{A}}$ computable by the transducer $\mathscr{A}$.

8.2. The cardinality of the set of finite memory functions

The following theorem implies that there exists no finite representation for all finite memory retrospective functions over (general) signals.

Theorem 22. *The set of finite memory speed-independent retrospective functions is uncountable.*

The proof of the theorem is based on the notion of 'homogeneous language' due to Trakhtenbrot. An ω-language L is said to be homogeneous [16] if the set of languages $\{L/w : w$ is a finite string$\}$ is finite, where $L/w = \{s : ws \in L\}$.

Theorem 23 (Trakhtenbrot [17]). *The set of homogeneous ω-languages is uncountable.*

Proof of Theorem 22. Let L be an ω-language over the alphabet $\{0, 1\}$. Let F_L be the function from signals over $\{0, 1, 2\}$ into signals over $\{0, 1\}$ defined as $F_L xt = 1$ iff there exists $t' \leqslant t$, $s = \langle a_0 \ldots a_n \ldots \rangle \in L$ and an ω-sequence $t_0 = 0 < t_1 < \cdots < t_n < \cdots$ such that

(1) $\lim t_i = t'$.
(2) $x(u) = 2$ for $u \in (t_i, t_{i+1})$.
(3) $x(t_i) = a_i$.

It is clear that F_L is a strongly retrospective speed-independent function. Moreover, if $L_1 \neq L_2$ then $F_{L_1} \neq F_{L_2}$.

It is clear that if an ω-language L is homogeneous then the function F_L has finite memory. The set of homogeneous ω-languages is uncountable [17], therefore, the set of finite memory speed-independent retrospective functions is at least uncountable. □

9. Conclusion and related work

Let us first re-examine the contents, results and techniques of the paper.

In Section 2 the behavior of finite devices operating in continuous time was formalized. The formalization is a smooth adaptation of the notions employed for the description of finite devices operating in discrete time. In Section 3 speed-independent and stable functions were introduced. Stability and speed-independence are invisible in discrete time, however, are important in continuous time. Speed-independent functions over non-Zeno signals were investigated in Section 5. It turns out that they are very similar to the functions over discrete time signals. The main technical efforts of Sections 6–8 were directed to the proof (of Theorem 21) that finite memory implies speed-independence for finite memory retrospective functions over non-Zeno signals. However, it turns out that in order to establish this result one has to leave the world of non-Zeno signals and to deal with functions over general signals. Our investigation of functions over general signals were needed for the proof of Theorem 16 which insures that the function *state* which produces the (names of) residuals of a finite memory retrospective function maps non-Zeno signals only to non-Zeno signals.

Though we have considered the time domain of non-negative reals, only the following properties of a time domain T are used in our proofs:

- T is a linear order with a minimal element and with no maximal element.
- There exists an associative function $+ : T \times T \to T$ such that for every $t \in T$ the function $\lambda\tau.t+\tau$ is an order preserving bijection from T to $\{t' : t' \geqslant t\}$.

One can see that the domain $\mathbf{Q}^{\geqslant 0}$ of non-negative rationals has also the above properties. Therefore, all notions, results and their proofs are immediately extended to $\mathbf{Q}^{\geqslant 0}$. The main notions and results can be adapted to time domains that do not have the above stated properties, e.g., to the time domain of $\{0\}\cup$ positive irrationals. However, such extensions are not immediate.

In the next subsection the relationships among stability, speed-independence and size of the memory are summarized. The other subsections describe some results related to the Trakhtenbrot's program [19,13,15] for lifting the classical trinity: monadic logic, nets and automata from discrete to continuous time. In this trinity monadic second-order logic of order represents a powerful specification formalism, the formalization of hardware via logical nets represents a lower level implementation formalism and finite transition diagrams represent an intermediate level formalism. In Sections 2–4 we recall some basic facts and state their extension to continuous time. We refer the reader to [15], where extensions to continuous time of the fundamental theorems of classical automata theory are provided.

9.1. Memory, speed-independence and stability

In Fig. 5, the inclusion relation among the properties of retrospective functions on non-Zeno signals is summarized. The inclusion Finite Memory⊂Speed-Independent was

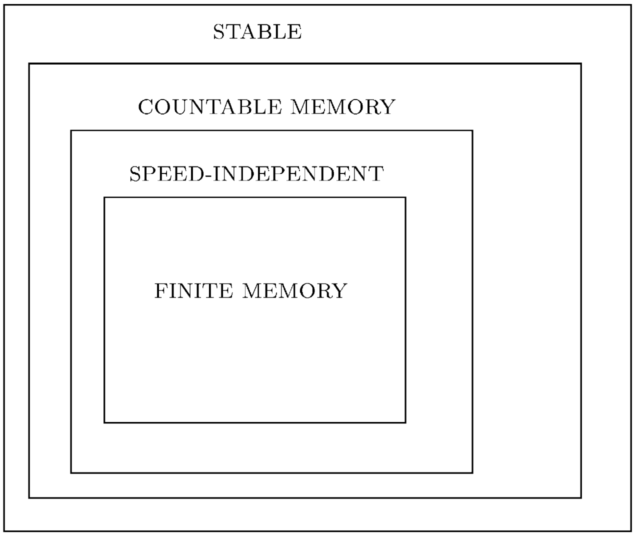

Fig. 5. Properties of retrospective functions.

proved in Theorem 21; the function *Prime* (see Section 4) shows that the inclusion is proper.

The inclusion Speed-Independent $\subset$ Countable Memory was proved in Corollary 6; a function which demonstrates that the inclusion is proper was given in [10].

The proof of inclusion Countable Memory $\subset$ Stable will be given elsewhere; this inclusion is proper, since the function F_{11} (see Section 4) is stable and has uncountable memory.

9.2. Canonical equations

Let $\delta : \Sigma_{in} \times Q \to Q$ and $out : \Sigma_{in} \times Q \to \Sigma_{out}$ be two functions, where Q, Σ_{in} and Σ_{out} are sets (non necessary finite). Let q_0 be an element of Q.

Consider the following system of equations:

$$q(t+1) = \delta(x(t), q(t)),$$

$$y(t) = out(x(t), q(t)),$$

$$q(0) = q_0.$$

In [17] such systems of equations are called canonical; the functions δ and *out* are said to be the conversion functions of a system. For finite Q, Σ_{in} and Σ_{out}, canonical systems were studied by Church [4] under the name restricted recursive arithmetic definitions.

It is easy to see that for every $x : \mathbf{N} \to \Sigma_{in}$ there exists a unique $q : \mathbf{N} \to Q$ and a unique $y : \mathbf{N} \to \Sigma_{out}$ such that the triple $\langle x, q, y \rangle$ satisfies given canonical system. Hence, we can define a function $G:(\mathbf{N}\to\Sigma_{in})\to(\mathbf{N}\to\Sigma_{out})$ and function $ST:(\mathbf{N}\to\Sigma_{in})\to Q$ such that for every $x\in\mathbf{N}\to\Sigma_{in}$ the triple $\langle x, G(x), ST(x)\rangle$ satisfies the system. These functions G and ST are said to be defined by the system. Observe that (1) G is retrospective and ST is strongly retrospective; (2) the cardinality of the sets of distinct residuals of G and of ST is bounded by the size of Q.

It is well known that for every retrospective function G from discrete signals over Σ_{in} (i.e., from the set $\mathbf{N}\to\Sigma_{in}$) into discrete signals over Σ_{out} there exists a canonical system of equations Sys such that G is definable by Sys.

Below a similar description of speed-independent retrospective functions over non-Zeno signals by systems of equations is provided [18].

We use $x(t^{+0})$ for the right limit of a non-Zeno signal x at t. Given functions $\delta:\Sigma_{in}\times\Sigma_{in}\to(Q\to Q)$ and $out:\Sigma_{in}\times Q\to\Sigma_{out}$ such that $\delta(b,a)\circ\delta(a,a)=\delta(b,a)$.

Consider the system of equations

$$q(t^{+0}) = \delta(x(t), x(t^{+0}))(q(t)),$$

$$y(t) = out(x(t), q(t)),$$

$$q(0) = q_0.$$

Observe that for every non-Zeno signal x there exists a unique non-Zeno signal y and a unique non-Zeno signal q such that $\langle x, y, q\rangle$ satisfies the system. Hence, such a system defines functions G (from non-Zeno signals over Σ_{in} to non-Zeno signals over Σ_{out}) and function ST (from non-Zeno signals over Σ_{in} to non-Zeno signals over Q) such that for every x the triple $\langle x, G(x), ST(x)\rangle$ satisfies the system. Note that G is retrospective and ST is strongly retrospective. Moreover, for every retrospective G there exists a system of equations of the above form that defines G. The corresponding conversion functions δ and out are defined like in Proposition 15, and Theorem 16. Though Proposition 15, and Theorem 16 deal with finite memory, the finite memory assumption can be replaced by the speed-independence requirement (see also the remark after Proposition 15).

The speed-independent retrospective functions over right open signals can be described in a similar way. Namely, let $\delta:\Sigma_{in}\to(Q\to Q)$ be such that $\delta(a)\circ\delta(a)=\delta(a)$ and let $out: Q\times\Sigma_{in}\to\Sigma_{out}$.

Consider the system

$$q(t^{+0}) = \delta(x(t^{+0}))(q(t)),$$

$$y(t) = out(x(t), q(t)),$$

$$q(0) = q_0.$$

Then for every right open signal x there exists a unique right open signal y and a unique non-Zeno signal q such that $\langle x, y, q\rangle$ satisfies the system. Hence, such a system defines functions G (from right open signals over Σ_{in} to right open signals over Σ_{out}) and function ST (from right open signals over Σ_{in} to non-Zeno signals over Q) such that for every x the triple $\langle x, G(x), ST(x)\rangle$ satisfies the system. Observe that

G is retrospective and ST is strongly retrospective. Note also that the only strongly retrospective functions over right open signals are constant functions, therefore in the above characterization of functions over right open signals, it is necessary that ST outputs non-Zeno signals.

It can be shown that for every retrospective speed-independent function G there exists a system of equations that defines G.

9.3. Monadic second-order theory of order

Recall that the language of monadic second-order theory of order (see e.g. [5,21]) has individual variables, monadic second-order variables, a binary predicate $<$, the usual propositional connectives and first and second-order quantifiers. The atomic formulas are formulas of the form: $t<v$ and $x(t)=b$, where t, v are individual variables and x is a monadic second-order variable and b is an element of a finite set Σ. The formulas are constructed from atomic formulas by logical connectives and first and second-order quantifiers.

In the standard discrete time interpretation of monadic logic, the individual variables range over natural numbers and the monadic variables range over the functions in $\mathbf{N}\rightarrow\Sigma$ (this set is isomorphic to the set of discrete time signals over Σ and to the set of ω-strings over Σ). A set of signals (i.e., ω-language) L is definable by $\phi(x)$ if L is the set of all $\mathbf{x}$ that satisfy $\phi(x)$. A function F over discrete time signals is definable by a formula $\phi(x, y)$ if the set of $\{\langle\mathbf{x},\mathbf{y}\rangle : \phi(\mathbf{x},\mathbf{y})\}$ is the graph of F.

Recall

Fact 1 (Trakhtenbrot [17]). *A retrospective function over ω-strings is definable in monadic second-order logic of order if and only if it is finite memory.*

Fact 1 holds with the following changes (1) replace functions over ω-strings by the functions over non-Zeno signals (2) as an interpretation for the individual variable (respectively, monadic variables) of the second-order monadic logic of order consider non-negative reals (respectively, non-Zeno signals over Σ [14,15]).

There are finite memory functions over (general) signals which are not definable in monadic-second-order logic of order because the set of such functions is uncountable (see Theorem 22) and the set of formulas is countable. However,

Fact 2. *If a function (over general signals) is definable in monadic second-order logic then it is speed-independent and finite memory.*

Proof (*Sketch*). Speed-independence follows from the observation that order preserving bijections are isomorphisms for the structures for monadic logic of order.

Assume that F is definable by formula $\phi(x, y)$ of quantifier rank k. Then every residual F' of F is definable by a formula $\phi'(x, y)$ of quantifier rank k. There are only finitely many semantically distinct formulas of quantifier rank k with free variables x, y. Therefore, F has a finite number of residuals. □

Finally, recall that in the discrete time case a language (set of ω-strings) L is definable by a monadic formula iff there exists a finite memory retrospective function $F:\Sigma_{in}^{\omega}\to\Sigma_{out}^{\omega}$ and $\Sigma\subset\Sigma_{out}$ such that $\mathbf{x}\in L$ iff $\exists t\in\mathbf{N}.\ \forall t'>t.(F\mathbf{x})(t')\in\Sigma\wedge\forall a\in\Sigma.\exists t''>t'.x(t'')=a$. Similar characterization holds for non-Zeno signals languages. Namely, a set L of non-Zeno signals over Σ_{in} is definable by a monadic formula iff there exists a finite memory retrospective function F and sets $\Sigma\subset\Sigma_{out}$ such that (1) F maps non-Zeno signals over Σ_{in} to non-Zeno signals over Σ_{out}; (2) $\mathbf{x}\in L$ iff $\exists t\in\mathbf{N}.\forall t'>t.(F\mathbf{x})(t')\in\Sigma\wedge\forall a\in\Sigma.\exists t''>t'.x(t'')=a$.

9.4. Real time

Many formalisms for specification of real-time behavior were suggested in the literature. Some of these formalisms (e.g., timed automata [1]) extend discrete time formalisms by introducing metrical real-time constraints, others (e.g., temporal logic of reals [2]) are defined by providing continuous (or dense) time interpretation for the modalities studied in the discrete cases, yet others (e.g., duration calculus [22]) are based on ideas that were not widely used among the formalisms for the specification of discrete time behavior.

It is worthwhile to distinguish two aspects of real time specifications: (A) Metric aspects which deal with the distance between moments of real time. (B) Speed-independent properties which rely only on the order of real numbers.

In this paper metric aspects of specification were ignored because the functions which rely on metric have uncountable memory. In [13,15] the extension of finite automata theory to hybrid and timed formalisms are suggested. In these extensions metrical properties of the reals are taken into account. Metrical properties are the most intriguing characteristic of real-time specifications which have led to a large number of highly interesting results and applications (see Proceedings of Hybrid and Real-Time Workshops, e.g. [9]).

Acknowledgements

B.A. Trakhtenbrot invited me to participate in his project for lifting classical automata theory from discrete to continuous time and to interaction paradigms. The results of this paper were obtained as a part of this more general program. I am grateful to B.A. Trakhtenbrot for many valuable ideas and suggestions.

Appendix A. Proof of Proposition 3(2)

In this appendix, we use standard notations for ω-strings. In particular $\langle a_0^{n_0}a_1^{n_1}\dots a_i^{n_i}\dots\rangle$ denotes the ω-string $\langle\underbrace{a_0\dots a_0}_{n_0}\underbrace{a_1\dots a_1}_{n_1}\dots\underbrace{a_i\dots a_i}_{n_i}\dots\rangle$.

Let G be a function that satisfies SI condition. Then

$$G(a_0a_1\dots a_i\dots) = b_0^{n_0}b_1^{n_1}\dots b_i^{n_i}\dots \quad \text{for some } b_i. \tag{A.1}$$

A string is α *stuttering free* [8] if either $\alpha = a_0^{n_0}a_1^{n_1}\dots a_i^{n_i}\dots$ and $\forall i.a_i \neq a_{i+i}$ or $\alpha = a_0a_1 \dots a_i \dots a_j^{\omega}$ and $\forall i<j.a_i \neq a_{i+1}$.

From (A.1) it follows that a function which satisfies SI condition is completely determined by its values on stuttering free strings, i.e., if $G_1\alpha = G_2\alpha$ for all stuttering free α and both G_1 and G_2 satisfy SI condition then $G_1 = G_2$.

Let x be a right open signal. Assume that there exists an ω-sequence of point $t_0 = 0 < t_1 < \cdots$ where x is not constant. Then $\tau = \langle t_0, \dots, t_i, \dots\rangle$ is a time scale. Let $\alpha(i)$ be defined as $x(t_i)$ (for $i = 0, 1, \dots$). Then α is a stuttering free string and α, τ characterizes x. Moreover, if α', τ' characterizes x and α' is stuttering free then $\alpha = \alpha'$ and $\tau = \tau'$.

Assume that there exists only a finite sequence $t_0 = 0 < t_1 < \cdots < t_j$ of points where a right open signal x is not continuous. Let $\alpha(i)$ be defined as $x(t_i)$ for $i<j$ and as $x(t_j)$ for $i \geqslant j$. Then α is stuttering free. Let τ be any time scale such that $\tau(i) = t_i$ for $i \leqslant j$. The α, τ characterizes x, moreover if α', τ' characterizes x and α' is stuttering free then $\alpha = \alpha'$ and $\tau(i) = \tau'(i)$ for $i \leqslant j$.

Let us define a binary relation R on right open signals as follows: $\langle x, y\rangle \in R$ if there exist α, β and τ such that (1) α is stuttering free and (2) $\beta = G\alpha$ and (3) α, τ characterizes x and β, τ characterizes y. From two preceding paragraphs it follows that for every x there exists a unique y such that $\langle x, y\rangle \in R$. Hence, R is the graph of a function F. Moreover, from the definition of R it follows that F is speed-independent. Hence, in order to complete the proof, it is sufficient to show that G characterizes F.

Assume that α', τ' characterizes x. We have to show that $G\alpha', \tau'$ characterizes Fx. The proof proceeds by two cases:

Case 1: α' has the form $\langle a_0^{n_0}a_1^{n_1}\dots a_i^{n_i}\dots\rangle$ where $a_i \neq a_{i+1}$ and $n_i > 0$ for all i.

Case 2: α' has the form $\langle a_0^{n_0}a_1^{n_1}\dots a_{j-1}^{n_{j-1}}a_j^{\omega}\rangle$ where $a_i \neq a_{i+1}$ and $n_i > 0$ for all $i<j$.

Consider the first case and define α as the stuttering free string $\langle a_0a_1\dots a_i\dots\rangle$; let m_i be defined as $\sum_{k<i} n_k$ and let $\tau(i)$ be defined as $\tau'(t_{m_i})$. Then α, τ and α', τ' characterize the same x. Let y be characterized by $G\alpha, \tau$. Then $y = Fx$ by the definition of F. From (A.1) and the construction of α, τ it follows that $G\alpha', \tau'$ and $G\alpha, \tau$ characterize the same signal. Hence, $G\alpha', \tau'$ characterizes Fx and this completes the proof of the first case.

The second case is proved similarly. □

Appendix B. Proof of Proposition 9

Recall that the suffix of a signal y at t notation ($suf(y, t)$) is the signal defined by $\lambda\tau.y(t + \tau)$.

Assume that a signal x has only finitely many distinct suffices. Let us denote the set of its suffices by Q. Let us define the function $F : \mathbf{R}^{\geqslant 0} \to (Q \to Q)$ as follows:

$$q' = F(t)q \quad \text{iff } q' = suf(q,t). \tag{B.1}$$

It is clear that

$$F(t_1 + t_2) = F(t_1) \circ F(t_2) = F(t_2) \circ F(t_1). \tag{B.2}$$

Therefore,

$$\forall k. F(t) = \underbrace{F(t/k) \circ F(t/k) \circ \cdots \circ F(t/k)}_{k}. \tag{B.3}$$

Observe that if $g : Q \to Q$ and size of Q is less than n then $g^{n!} = g^{n!} \circ g^{n!}$ (we use g^r for $\underbrace{g \circ g \cdots \circ g}_{r}$). This observation and (B.3) imply

$$\forall t. F(t) \circ F(t) = F(t) \tag{B.4}$$

and therefore,

$$F(t/k) = F(t) = F(mt) \quad \text{for every positive natural numbers } k \text{ and } m. \tag{B.5}$$

Lemma B.1. *If* $suf(x,t_1) = suf(x,t_2) = q$ *then* $suf(x,t) = q$ *for all* $t \in (t_1,t_2)$.

Proof. Assume that $t_1 < t_2$ (the case $t_1 \geqslant t_2$ is trivial.). Let $t \in (t_1,t_2)$ and let $p = suf(x,t)$ we are going to show that $q = p$.

Define $\delta_1 \triangleq t_2 - t_1$, $\delta_2 = t - t_1$ and $\delta_3 \triangleq t_2 - t$. It is clear that $\delta_i > 0$.

Observe that from (B.1) and (B.2) and from our assumptions it follows:

$$F(\delta_1)(q) = q, \tag{B.6}$$

$$F(\delta_2)(q) = p, \tag{B.7}$$

$$F(\delta_3)(p) = q. \tag{B.8}$$

From (B.4) and (B.8) we obtain

$$F(\delta_3)(q) = F(\delta_3)(F(\delta_3)(p)) = F(\delta_3)(p) = q. \tag{B.9}$$

Note that $\delta_1 = \delta_3 + \delta_2$, therefore from (B.2), (B.7) and (B.9) we obtain

$$F(\delta_1)(q) = F(\delta_2 + \delta_3)(q) = F(\delta_2)(F(\delta_3)(q)) = F(\delta_2)(q) = p. \tag{B.10}$$

From Eqs. (B.10) and (B.6) we obtain that $p = q$. This completes the proof of the lemma.

Let us proceed now with the proof of Proposition 9.
Take any $t_0 > 0$ and let q be $suf(x, t_0)$.
By (B.1),

$$q = F(t_0)x. \tag{B.11}$$

Therefore, by (B.5),

$$suf(x, t_0/k) = q = suf(x, mt_0), \\ \text{where } k \text{ and } m \text{ are positive natural numbers.} \tag{B.12}$$

Hence by Lemma B.1,

$$q = suf(x, t) \text{ for every } t \in (t_0/k, mt_0), \\ \text{where } k \text{ and } m \text{ are positive natural.} \tag{B.13}$$

Hence, $q = suf(x, t)$ for every positive t. Therefore, $\forall t > 0.q(0) = x(t)$. This establishes that x is constant on the positive reals.

References

[1] R. Alur, D. Dill, A theory of timed automata, Theoret. Comput. Sci. 126 (1994) 183–235.
[2] H. Baringer, R. Kuiper, A. Pnueli, A really abstract concurrent model and its fully abstract semantics, in: Proc. 13th ACM Symp. on Principles of Programming Language, 1986, pp. 173–183.
[3] A.W. Burks, H. Wang, The logic of Automata. J. ACM, 4, 1957 193–218, 279–297.
[4] A. Church, Logic, arithmetic and automata, in: Proc. Internat. Congr. on Mathematics, Moscow, 1963, pp. 23–35.
[5] Y. Gurevich, Monadic second-order theories, in: J. Barwise, S. Feferman (Eds.), Handbook of Model-Theoretical Logics, Springer, Berlin, 1985, pp. 479–506.
[6] S. Kleene, Representation of events in nerve nets and finite automata, Automata Studies, Princeton, 1956, pp. 3–41.
[7] N. Kobrinski, B.A. Trakhtenbrot, Introduction to the Theory of Finite Automata, North-Holland, Amsterdam, 1965. (English translation) (in Russian 1962)
[8] L. Lamport, The temporal logic of actions, ACM Trans. Programming Languages Systems 16 (3) (1994) 872–923.
[9] O. Maler (Ed.), Hybrid and real-time system, Internat. Workshop HART97, Lecture Notes in Computer Science, Vol. 1201, Springer, Berlin, 1997.
[10] O. Maler, L. Staiger, Unpublished manuscript, 1998.
[11] R. McCulloch, W. Pitts, A logical calculus of the ideas immanet in nervous activity, Bull. Math. Biophys. 5 (1943) 115–133.
[12] M. Minsky, Computation: Finite and Infinite Machines, Prentice-Hall, Englewood Cliffs, NJ, 1967.
[13] D. Pardo, A. Rabinovich, B.A. Trakhtenbrot, On synchronous circuits over continuous time, Technical Report, Tel Aviv University, 1997.
[14] A. Rabinovich, On translation of temporal logic of actions into monadic second-order logics, Theoret. Comput. Sci. 193 (1998) 197–214.
[15] A. Rabinovich, B.A. Trakhtenbrot, From finite automata toward hybrid systems, Proc. Fundamentals of Computation Theory, Lecture Notes in Computer Science, Vol. 1450, Springer, Berlin, 1998, pp. 411–422.
[16] D. Scott, Some definitional suggestions for automata theory, J. Computer System Sci. 1 (1967) 187–212.

[17] B.A. Trakhtenbrot, in: Finite Automata and the Logic of One-place Predicates, American Mathematical Society Translation, Vol. 59, American Mathematical Society, Providence, RI, 1966, pp. 23–55 (in Russian 1961).
[18] B.A. Trakhtenbrot, Lecture Notes of a Course on Verification of Software and Hardware, Tel Aviv University, Fall 1994.
[19] B.A. Trakhtenbrot, Origins and metamorphosis of the trinity: logics, nets, automata, in: Proc. LICS, San Diego, CA, 1995.
[20] B.A. Trakhtenbrot, Automata and hybrid systems, Notes on a Course at Uppsala University, Fall 1997.
[21] B.A. Trakhtenbrot, Y.M. Barzdin, Finite Automata, North-Holland, Amsterdam, 1973. (English Translation) (in Russian 1970)
[22] C. Zhou, C.A.R. Hoare, A.P. Ravn, A calculus of duration, Inform. Process. Lett. 40 (5) (1991) 269–279.