Universidade Técnica de Lisboa
Instituto Superior Técnico

Finite automata over continuous time

Alexandre Paulo Lourenço Francisco

Applied Mathematics and Computation

Diploma Thesis

Supervisor:

Prof. José Félix Costa

June 2002

Acknowledgements

I would like to thank to my supervisor, Professor José Félix Costa, for his guidance during the elaboration of this dissertation. His support and motivation have contributed decisively to my future as a scientist and mathematician.

I am indebted to Dr. Alexander Rabinovich and Professor Boris Trakhtenbrot for their suggestions concerning this dissertation. Specially, I would like to thank to Professor Boris Trakhtenbrot for helping me with some bibliography and Dr. Alexander Rabinovich for his critical apreciation of chapter 7.

I would like to thank Professors Amilcar Sernadas and Cristina Sernadas for their help and suggestions. Specially, I want to thank Professor Amilcar Sernadas, as President of the Center for Logic and Computation, for inviting me to give a seminar about the topic of this dissertation.

Thanks also to all people in the Computer Science Section of the Department of Mathematics of IST for their support, specially to Dr. Miguel Dionísio and Dr. Paula Gouveia for their patience along the preparation of this dissertation.

I wish to thank to my colleagues Ricardo Silva, Pedro Adão, and Luís Russo for their friendship.

Thanks also to João Rasga, Hugo Lourenço, and Dr. Isabel Nunes for providing me some references I needed.

I am also indebted to my family for their support. I would like to thank Cátia Vaz for her patience and love during the preparation of this dissertation and for helping me with some details in the text.

Contents

List of symbols

T^+	Time set
[a,b[	Time intervals
Σ	Set of states of a channel
$Sig(\Sigma)$	Set of signals
$suf(x,t)$	Suffix of a signal x
$x\rfloor^t;z$	Concatenation of x and z
$Res(F,\vec{x},t)$	Residual of F
$nZSig(\Sigma)$	Set of non-Zeno signals over Σ
$Const_a$	Constant signal equal to a
ρ	Order preserving bijection
$x,y,x_1,x_2,...$	Signals
$\alpha,\beta,\alpha_1,\alpha_2,...$	ω-strings
$\perp$	Undefined
$out_{\vec{G}}$	Output function
$state_{\vec{G}}$	State function
Q	Set of states of a transducer
$[\mathcal{X};OP]$	Function algebra
$\mathcal{O}_{FMR}$	Set of finite memory retrospective operators
$LLim_a$	Left limit operator
$LJV^n_{k,a}$	Last jump value operator
$Rec(G)\vec{x}$	Feedback operator
$\vee$	Least upper bound
$\wedge$	Greatest lower bound

Chapter 1

Introduction

Nowadays control systems are commonly found in many devices such as industrial robots and airplanes. The study of these devices and their interaction with their environment leads us to the mathematical theory of control, which deals with the analysis and design of control systems.

As we can see in [Son90], there have been two main lines of work in control theory. One take us over the models in mathematics and physics and over their possible optimization with respect to a particular behavior. The other main line is based on the constraints of a specific object and the goal is to correct the deviations of such object from a desired behavior.

So and specially in what concerns the second main line, we need to study systems which involve interacting networks of digital and continuous systems, i.e., hybrid systems. These incorporate both discrete and continuous dynamics in which the continuous aspects may require incursions into calculus and differential equations. As we know, differential equations have nothing in common with existing and well understood tools of automata theory and logic.

The approximation between automata theory and continuous systems lead us to extensions of the basic finite automata paradigm. A first extension arise from the idea of interaction with environment seen as an oracle. A second one focuses on the use of continuous time instead of discrete time and ignores interaction with the oracle. As is put in [RT98], it is believed that these two orthogonal extensions may facilitate a structured formalization of hybrid systems and a lucid adaptation of basic automata theory to hybrid systems.

In this work we will be around the second extension and we will follow ideas discussed by B. A. Trakhtenbrot and A. Rabinovich, see e.g. [Rab97, RT98, Tra98, Tra99]. Their work include many definitions and formalizations with respect to automata over continuous time, namely with respect to lift concepts of the classical automata theory from discrete to continuous time; many of the results achieved by them will be discussed in this dissertation. We note also that this work is specially related to speed independent properties which rely on the order of real numbers, metric aspects which deal with the distance between real numbers are not considered because, as we will see, operators which rely on metric of reals have uncountable memory.

R. Alur and D. L. Dill (see, e.g., [AD94]) have proposed timed automata to model the behavior of real-time systems and in their approach metric properties of the reals are taken into account. In fact the work of R. Alur and D. L.

Dill deals with timed automata, finite automata which have a finite number of clocks associated, that accept words in which real-valued time of occurrence is associated with each symbol. I.e., a timed automaton accepts words as a classical automaton with a major difference: the time between the occurrence of two symbols is not necessarily constant and may be a real number. Note that, in the approach took by B. A. Trakhtenbrot and A. Rabinovich, signals over real-time are considered in place of words and automata compute operators on signals in place of accept words.

This dissertation is organized as follows. In chapter 2 we will discuss some definitions and concepts of classic automata theory and we will define in an axiomatic manner the behavior of finite state automata operating in continuous time. As is done in [Rab97], many postulates of automata theory will be analyzed and concepts of retrospective operators and finite memory operators will be studied with some improvements. Namely and with respect to [Rab97, RT98, PRT01], it will be formalized in detail the concept of time set, the extension to n-ary operators will be considered and proofs of propositions 2.2.5 and 2.2.7 will be given too.

Stability and speed independence will be object of study in chapter 3. With the lift to continuous time comes up some properties of signals invisible at discrete time, for example majority of signals will be sensible to expansion and compression of time space. Concepts such as stable operator and speed independent operator found, e.g., in [Rab97, PRT01] will be given and some generalizations will be done with respect to n-ary operators. We also prove some results about these properties, namely the characterization of speed independent operators over non-Zeno signals and over right open signals indicated in [Rab97] will be generalized and complete proofs of propositions 3.1.3 and 3.2.3 will be provided.

Chapter 4 gathers many examples found in [Rab97, RT98, PRT01] which ones we have studied here in detail. They will illustrate many of concepts and properties introduced in chapters 2 and 3.

In chapter 5 we will study closure properties of operators on signals and some properties of finite memory retrospective operators. These have been introduced in [Rab97] and we will contribute with some improvements, namely the generalization to n-ary operators and complete proofs of propositions 5.1.1, 5.1.2 and 5.2.8 will be given.

The representation of finite memory operators found in [Rab97] is discussed in chapter 6 where we generalize the notion of finite state transducer to n-ary operators, i.e., finite state transducers with multiple input channels are given. We also give illustrative examples of automata to finite memory retrospective operators already introduced in chapter 4, these examples will clarify some ideas about the relation between the states of a transducer and the residuals of an operator.

Our main contribution to this theory is given in chapter 7. A physical device, in which complex transformations are implemented, is usually an appropriate combination of elementary parts that interact as desired. These idea conducts us to the concept of circuit which appears many times in literature (see, e.g., [KT65]) and which permits, for example, describe a given automaton as the combination of elementary automata.

We introduce circuits of finite memory retrospective operators over signals, i.e., we choose a set of elementary finite memory operators and we study how to obtain all finite memory retrospective operators by constructing circuits with the elementary operators. In order to perform this study, we use the notion of function algebra in [Clo99] and we obtain an algebra of finite memory retrospective operators. The equivalence between this algebra of operators and the set of finite memory retrospective operators is stated in propositions 7.3.1 and 7.2.15. We note that the proof of proposition 7.3.1 is constructive and so, making use of the elementary operators and operations provided by this algebra, we can construct circuits of operators.

Chapter 7 includes also examples of circuits for the finite memory retrospective operators given in chapter 4.

Chapter 2

Postulates

In this chapter we pretend to lift the basic concepts of classical automata theory from discrete to continuous time. For that purpose we will follow the main ideas found in [Rab97, PRT01, RT98].

Automata Theory is commonly introduced as a study of sets of strings accepted by finite machines, see e.g. [HMU01]. However as is stated in [Tra99], the operators realized by these machines are more basic than the sets accepted by them. This is in accordance with the belief that in Automata Theory as well as in Computability Theory operators are more fundamental than sets and this point of view was followed consistently in [KT65].

Therefore our interest is the study of Automata Theory considering operators realized by some finite machines.

Before going on we observe that a machine is considered as a closed box with input and output channels. The user interacts with the machine through the input channels and gets the answers from the output channels.

With respect to [Rab97, PRT01, RT98] we provide the following extensions and more deep explanations: definition of time set, extension to n-ary operators and proofs of propositions 2.2.5 and 2.2.7.

2.1 Postulates of Automata Theory

The aim of this section is to define in an axiomatic way the behavior of finite state devices operating in continuous time. For decades, most of the following concepts have been employed for the behavior of finite state devices operating in discrete time. However, concerning continuous time, formalisms and notations are many times obscured by imprecise definitions.

Taking advantage of ideas presented in [Rab97], we will examine some postulates commonly followed in automata theory.

2.1.1 Nature of Time

Looking at discrete time systems, the underlying time set is discrete and we usually think that it is the set of integers $\mathbb{Z}$. When we consider a continuous time system, each instant is a real number and therefore we think in the time set as the set of real numbers $\mathbb{R}$. For both cases there are common assumptions and we consider the following postulates.

Postulate 2.1.1 (Linear Time) *The set of moments of time is a linearly ordered set.*

Postulate 2.1.2 (Discrete Time) *The set of moments of time is the set of natural numbers where* 0 *is the beginning of time.*

For current work we will consider the next postulate.

Postulate 2.1.3 (Continuous Time) *The set of moments of time is represented by the non-negative real numbers where* 0 *is the beginning of time.*

To treat both continuous and discrete time cases simultaneously, we use the notation followed for example in [Son90]. So we define time set.

Definition 2.1.4 A *time set* T is a subgroup of $(\mathbb{R}, +)$ with addition and the usual order relation $<$. For any such set, T^+ is the set of non-negative elements $\{t \in T : t \geq 0\}$.

Whenever the time set T is understood from the context, the intervals are restricted[1] to T and we have:

$$[a, b[= \{t \in T : a \leq t < b\}.$$

2.1.2 Finiteness Postulates

For our devices we adopt the following postulates too.

Postulate 2.1.5 (Finiteness of the Channels' number) *A machine has a finite number of input and output channels.*

Postulate 2.1.6 (Finiteness of Channels' states) *At a given instant of time a channel can be in one among a finite number of possible states.*

If Σ is the set of states, then an element of Σ is a possible state of a Σ-channel.

2.1.3 Input-Output Behavior

Given a channel c, the input to a machine through c is the state of that channel at each moment of time. We define the sequence of states of a channel along time as follows.

Definition 2.1.7 A *signal* over a channel is a function

$$s : T^+ \longrightarrow \Sigma$$

where T is the time set and Σ is the set of the channel's states.

[1] If $T = \mathbb{Z}$ and $k > 0$, then $[m, m + k[= \{m, m + 1, ..., m + k - 1\}$.

Let Σ be the set of channel's states for some channel c, we denote the set of signals defined through c as $Sig(\Sigma)$.

The following postulates are relative to the acceptable behavior of a machine.

Postulate 2.1.8 (Deterministic Behavior) *The output signals are completely determined by input signals.*

Postulate 2.1.9 (Causal Behavior) *The output at a moment t does not depend on future inputs.*

Postulate 2.1.10 (Strong Causal Behavior) *The output at a moment t only depends on the past inputs.*

Before continuing we observe that we did some generalizations with respect to [Rab97]. In [Rab97] only unary operators have been considered, but we know that a machine may have many input channels and the output through a given channel may depend on multiple input channels. So we will consider n-ary operators from signals to signals.

The following definition formalizes the concepts found in the last three postulates.

Definition 2.1.11 ((Strong) Retrospective Operator) Let F be an operator from signals to signals,

$$F : Sig(\Sigma)^n \longrightarrow Sig(\Sigma).$$

- F is retrospective with respect to the i-th argument if for any $\vec{x}, \vec{y} \in Sig(\Sigma)^n$ such that $x_j = y_j$ for $i \neq j$ and $t \in T^+$ the following condition holds: If x_i and y_i coincide in the interval $[0, t]$ then $F\vec{x}$ and $F\vec{y}$ coincide in the interval $[0, t]$.

- F is strong retrospective with respect to the i-th argument if for any $\vec{x}, \vec{y} \in Sig(\Sigma)^n$ such that $x_j = y_j$ for $i \neq j$ and $t \in T^+$ the following condition holds: If x_i and y_i coincide in the interval $[0, t[$ then $F\vec{x}$ and $F\vec{y}$ coincide in the interval $[0, t]$.

Given a set $S \subset \{1, ..., n\}$ of components, an operator F is (strong) retrospective with respect to S when it is (strong) retrospective with respect to i for any $i \in S$. An operator F is (strong) retrospective when it verifies the above conditions for all components i.

As is indicated in [Rab97], the postulates 2.1.9 and 2.1.10 imply that the input-output behavior of a machine is a retrospective or strong retrospective operator.

The last postulate is a key postulate of finite automata theory [Rab97].

Postulate 2.1.12 (Finite Memory) *A machine can distinguish by its present and future behavior between only a finite number of classes of possible signals histories.*

Through the rest of this section we will try to formalize this postulate. In first place we define history of a signal, prefix and suffix of a signal and concatenation of signals.

Definition 2.1.13 A *t-history* over Σ is an operator from the interval $[0, t]$ to Σ. A t-history h is a t-history of a signal x if $h(\tau) = x(\tau)$ for $\tau \leq t$.

Definition 2.1.14 The restriction[2] of $x \in Sig(\Sigma)$ to the interval $[0, t[$ is called *t-prefix* of x. The *suffix* of x at t, $suf(x, t)$, is the signal $y \in Sig(\Sigma)$ defined as

$$y(t') = x(t + t').$$

Definition 2.1.15 Let $x, z \in Sig(\Sigma)$. The *concatenation* of t-prefix of x and z (notation $x\rfloor^t; z$) is defined as:

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

Suppose that we have two copies, M_1 and M_2, of the a machine M. Let $x_1, x_2 \in Sig(\Sigma)$ be two signals and h_1, h_2 be t-histories of x_1, x_2, respectively. Therefore, we say that h_1, h_2 are indistinguishable if at time t and after both machines M_1, M_2 produce the same output. We will say that a machine, or an operator over signals, has finite memory when given an instant t and some t-history it will assume a behavior among a finite number of possibilities.

In order to formalize this idea we introduce the concept of residual. As is discussed in [PRT01], given an ω-operator F, an ω-operator G is the residual of F with respect to a string u of length k if $y = G(x)$ and $z = F(ux)$ implies $\forall\ \tau \geq 0,\ y(\tau) = z(\tau + k)$. The extension of this notion to the case of signals is straightforward, given an operator F on signals, a signal operator G is the residual of F with respect to a t-history h of a signal x' if $y = G(x)$ and $z = F(x'\rfloor^t; x)$ implies $\forall\ \tau \geq 0,\ y(\tau) = z(\tau + t)$. Clearly, G is a residual of a signal operator F if there are a signal x and an instant t such that G is the residual of F with respect to the t-history h of x, i.e., with respect to x and t.

The following definition generalizes the notion of residual to n-ary operators and some notation is introduced.

Definition 2.1.16 (Residual [Rab97]) Let $F : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$ be an operator on signals, $\vec{x} \in Sig(\Sigma)^n$ a signal and $t \in T^+$ a time point. The residual of F with respect to $\vec{x}$ and t, $Res(F, \vec{x}, t)$, is the operator:

$$\lambda z_1 ... z_n . \lambda t' . F((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n))(t + t').$$

We note that if F is a retrospective operator, then $Res(F, \vec{x}, t)$ maps signal $\vec{z}$ on z' iff F maps $(x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n)$ on $y\rfloor^t; z'$ for some $y \in Sig(\Sigma)$.

Definition 2.1.17 (Finite Memory [Rab97]) An operator F is a finite memory operator if it has finitely many distinct residuals, i.e., the set

$$\{Res(F, \vec{x}, t) : \vec{x} \in Sig(\Sigma)^n, t \in T^+\}$$

is finite.

So the postulates 2.1.8, 2.1.9, 2.1.10 and 2.1.12 are summarized as:

[2] As is done in [Rab97] we will use $x\rfloor^t$, $x]^t$, $x\lfloor_t$ and $x[_t$ for the restriction of x to $[0, t[$, $[0, t]$, $]t, +\infty[$ and $[t, +\infty[$, respectively.

Postulate 2.1.18 (Input-output) *The input-output behavior of a machine is a finite memory retrospective operator.*

As we have done in definition 2.1.17, we may define also countable memory.

Definition 2.1.19 (Countable Memory) An operator F is a countable memory operator if it has countable many distinct residuals, i.e., the set

$$\{Res(F, \vec{x}, t) : \vec{x} \in Sig(\Sigma)^n, t \in T^+\}$$

is countable.

2.2 Non-Zeno Signals

As is stated in [Rab97], the piecewise constant signals are physically more realistic than general signals. So we define next the concept of *non-Zeno* signal.

Definition 2.2.1 (Non-Zeno Signal) A signal x is *non-Zeno* or piecewise constant if there exists an unbounded increasing ω-sequence $t_0 = 0 < t_1 < ... < t_n < ...$ such that x is constant in all sub-intervals $]t_i, t_{i+1}[$. $nZSig(\Sigma)$ will denote the set of non-Zeno signals over the alphabet Σ.

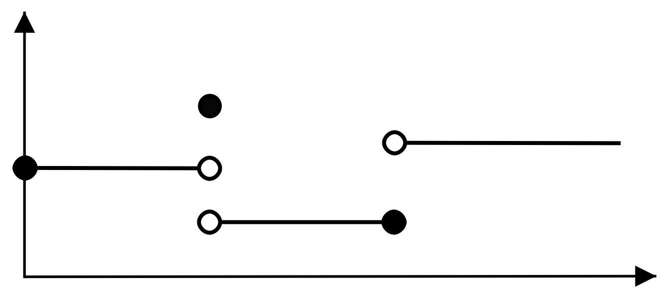

Figure 2.1: A non-Zeno signal.

However, we can get signals physically more realistic than non-Zeno signals if we impose the condition of *non-zero duration*.

Definition 2.2.2 (Non-zero Duration) A signal satisfies the requirement of *non-zero duration* if it does not have instantaneous jumps, i.e, if for every $t \in T^+$, x is constant in an interval with non empty interior that contains t.

Unfortunately and as is observed in example 2.2.3, the set of signals satisfying the above requirement is not closed under boolean operations.

Example 2.2.3 Let x and y be the following signals:

$$x(t) = \begin{cases} a & \text{if } 0 \leq t < 1 \\ b & \text{otherwise} \end{cases}$$

$$y(t) = \begin{cases} a & \text{if } 0 \leq t \leq 1 \\ b & \text{otherwise} \end{cases}$$

When we test their equality we obtain the following non-Zeno signal:

$$eq(t) = \begin{cases} False & \text{if } t = 1 \\ True & \text{otherwise} \end{cases}$$

with an instantaneous jump at $t = 1$.

But we can make more restrictions and get the class of *right open signals*.

Definition 2.2.4 (Right Open Signal) A signal x is *right open* if for every $t \in T^+$ there exists $t' \in T^+$ such that $t' > t$ and x is constant in $[t, t'[$.

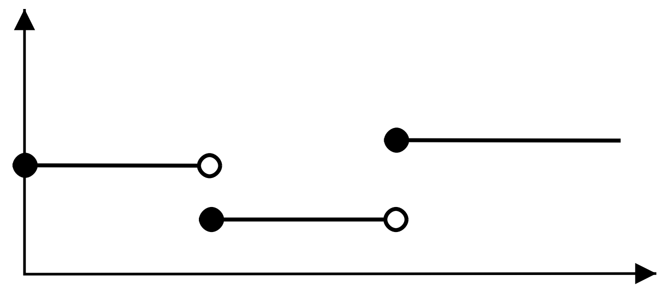

Figure 2.2: A right open signal.

It is easy to check that both the set of non-Zeno signals and the set of right-open signals are closed under suffix and concatenation. Constant signals are obviously included in these sets.

Proposition 2.2.5 *The set of non-Zeno signals and the set of right open signals are closed under suffix and concatenation.*

Proof: Let $t \in T^+$. If x is a non-Zeno signal, by definition 2.2.1, there exists an unbounded increasing ω-sequence $t_0 = 0 < t_1 < ... < t_n < ...$ such that x is constant in all sub-intervals $]t_i, t_{i+1}[$. Therefore, since $t_k \leq t < t_{k+1}$ for some k, we know that $suf(x, t)$ is constant in all sub-intervals $]s_j, s_{j+1}[$ for $s_0 = t$ and $s_j = t_{k+j}$ for $j \geq 1$, i.e., $suf(x, t)$ is a non-Zeno signal by definition 2.2.1.

Let y be another non-Zeno signal, by definition 2.2.1 there exists an unbounded increasing ω-sequence $s_0 = 0 < s_1 < ... < s_n < ...$ such that y is constant in all sub-intervals $]s_i, s_{i+1}[$. So $x\rfloor^t; y$ will be constant in all sub-intervals obtained from the ω-sequence $t_0 = 0 < t_1 < ... < t_k \leq t < s_1 + t < ... < s_n + t < ...$, i.e., it is a non-Zeno signal by definition 2.2.1.

Let x be a right open signal and $t \in T^+$, by definition 2.2.4 we know that for every $\tau \in T^+$ there exists $t' \in T^+$ such that $t' > t + \tau$ and x is constant in $[t + \tau, t'[$. Therefore, for every $\tau \in T^+$ there exists t'' such that $suf(x, t)$ is constant in $[\tau, t''[$, i.e., such that x is constant in $[t + \tau, t + t''[$. Take $t'' = t' - t$. By definition 2.2.4, $suf(x, t)$ is a right open signal.

Let y be another right open signal and $t, \tau \in T^+$, we want to prove that there exists $t' \in T^+$ such that $x\rfloor^t; y$ is constant in $[\tau, t'[$. The case $\tau \geq t$ is clear because y is a right open signal. If $\tau < t$ and since x is a right open signal, there exists t'' such that x is constant in $[\tau, t''[$. Then, $x\rfloor^t; y$ is constant in $[\tau, min(t'', t)[$ and we choose $t' = min(t'', t)$, i.e., $x\rfloor^t; y$ is a right open signal by definition 2.2.4. $\square$

In order to study what happen to signals and operators on signals when we stretch time, we give in the following definition the concept of order preserving bijection.

Definition 2.2.6 (Order Preserving Bijection) Let $\rho : T \longrightarrow T$ be a bijective function. Then ρ is an order preserving bijection if it also verifies:

- $\rho(0) = 0$;
- if $t < t'$, then $\rho(t) < \rho(t')$.

Clearly, we may restrict an order preserving bijection $\rho : T \longrightarrow T$ to T^+ and we get an order preserving bijection over T^+. We note also that, in the discrete time case, the identity is the unique order preserving bijection.

Let us suppose that ρ is an order preserving bijection. If C is the class of non-Zeno signals or the class of right open signals, then we prove that $x \circ \rho \in C$ for any $x \in C$.

Proposition 2.2.7 *Let C be the set of non-Zeno signals or the set of right open signals. If ρ is an order preserving bijection and $x \in C$, then $x \circ \rho \in C$.*

Proof: Let $t \in T^+$. If x is a non-Zeno signal, by definition 2.2.1, there exists an unbounded increasing ω-sequence $t_0 = 0 < t_1 < ... < t_n < ...$ such that x is constant in all sub-intervals $]t_i, t_{i+1}[$. Therefore, since ρ is an order preserving bijection, there are $t'_0, ..., t'_n, ... \in T^+$ such that $\rho(t'_i) = t_i$ and $t'_0 = 0 < t'_1 < ... < t'_n < ...$ and we also know that $x \circ \rho$ is constant in all sub-intervals $]t'_i, t'_{i+1}[$, i.e, $x \circ \rho$ is a non-Zeno signal by definition 2.2.1.

Let x be a right open signal and $t \in T^+$, by definition 2.2.4 we know that for $\rho(t) \in T^+$ there exists $t' \in T^+$ such that $t' > \rho(t)$ and x is constant in $[\rho(t), t'[$. Since ρ is an order preserving bijection, there exists $t'' \in T^+$ such that $\rho(t'') = t'$ and therefore x is constant in $[\rho(\tau), \rho(t'')[$. Then $x \circ \rho$ is constant in $[\tau, t''[$, i.e., by definition 2.2.4, $x \circ \rho$ is a right open signal. $\square$

By previous propositions[3] and observations we know that the only proper subsets C of non-Zeno signals which verify the following conditions:

- C is closed under suffix, i.e., if $x \in C$, then $suf(x, t) \in C$ for any $t \in T^+$.
- C is closed under concatenation, i.e., if $x, y \in C$, then $x\rfloor^t; y \in C$ for any $t \in T^+$.
- if ρ is an order preserving bijection and $x \in C$, then $x \circ \rho \in C$.
- C contains all constant signals.

are the set of non-Zeno signals, the set of right open signals, the set of non-Zeno signals that have finitely many changes and the set of right open signals that have finitely many changes.

[3]The proof is easily generalized for these cases.

Results of this chapter

2.2.5 The set of non-Zeno signals and the set of right open signals are closed under suffix and concatenation.

2.2.7 Let C be the set of non-Zeno signals or the set of right open signals. If ρ is an order preserving bijection and $x \in C$, then $x \circ \rho \in C$.

Chapter 3

Speed Independence and Stability

The aim of this chapter is to study speed independent and stable operators, namely we will give a characterization of these operators whenever they are defined over right open signals and non-Zeno signals. Beforehand we give some necessary definitions.

Definition 3.0.1 A signal x is *constant* at $t \in T^+$ if there are $t_1, t_2 \in T^+$ such that $t_1 < t < t_2$ and x is constant in the interval $]t_1, t_2[$.

In the discrete case, $T = \mathbb{Z}$, it is clear that every signal is constant at every moment t.

Definition 3.0.2 (Left limit) Let x be a signal, x has *left limit* c at $t \in T^+$ if there exists $t' \in T^+$ such that $t' < t$ and $x(\tau) = c$ for $\tau \in]t', t[$.

Definition 3.0.3 (Right limit) Let x be a signal, x has *right limit* c at $t \in T^+$ if there exists $t' \in T^+$ such that $t < t'$ and $x(\tau) = c$ for $\tau \in]t, t'[$.

Definition 3.0.4 Let x be a signal, x is *continuous from the left (right)* at $t \in T^+$ if the left (respectively right) limit of x at t is equal to $x(t)$. The signal x is continuous at $t \in T^+$ if it is continuous from the left and from the right at t.

Note that in discrete time every signal is continuous for a given instant t. As we can see, the conditions in definitions 3.0.2 and 3.0.3 are always verified with an arbitrary c because the sets $]t', t[$ and $]t, t'[$ may be empty. Therefore, by definition 3.0.4, it follows that every signal is continuous.

Definition 3.0.5 (Stability [Rab97]) A total operator F from signals to signals is *stable* if for every instant $t > 0$ and signal $\vec{x} \in Sig(\Sigma)^n$ the following implication holds: x_i is constant at t for all i implies $F\vec{x}$ is constant at t.

It is clear that in the discrete case every operator is stable since every signal is constant at every instant as we observed above.

Next proposition, stated but not proved in [Rab97], states an interesting property of stable operators and follows straightforward from what we have seen in previous definition.

Proposition 3.0.6 *A stable operator maps non-Zeno signals to non-Zeno signals.*

Proof: Let $F : Sig(\Sigma) \longrightarrow Sig(\Sigma')$ be a stable operator and $\vec{x} \in nZSig(\Sigma)^n$. By definition 2.2.1, there exists an unbounded increasing ω-sequence $t_0 = 0 < t_1 < ... < t_n < ...$ such that $\vec{x}$ is constant in all sub-intervals $]t_i, t_{i+1}[$ and, since F is stable, $F(\vec{x})$ is also constant in all sub-intervals $]t_i, t_{i+1}[$. Therefore, by definition 2.2.1, $F(\vec{x})$ is a non-Zeno signal. $\square$

We also have an interesting property that relates countable memory operators that map non-Zeno signals into non-Zeno signals and stability. These property was conjectured in [Rab97] for arbitrary countable memory operators, however it was not proved. (Countable memory was defined in 2.1.19.)

Proposition 3.0.7 *Every countable memory operator that maps non-Zeno signals to non-Zeno signals is stable.*

Proof: Let F be a countable memory operator that maps non-Zeno signals into non-Zeno signals and suppose that F is not stable. Then, by definition 3.0.5, there exists a non-Zeno signal $\vec{x}$ and an instant $t \in T^+$ such that x_i is constant at t, for every $i \in \{1, ..., n\}$, and $F\vec{x}$ is not constant at t, i.e., for every $\epsilon > 0$, there exists $\tau \in [t-\epsilon, t+\epsilon]$ such that $F\vec{x}\tau \neq F\vec{x}t$. Since $\vec{x}$ is constant at t, there exists $t_1 < t$ and $t_2 > t$ such that $\vec{x}$ is constant in $[t_1, t_2]$, i.e., $\vec{x}(t') = (a_1, ..., a_n)$, for $t' \in [t_1, t_2]$. We know that $F\vec{x}$ is a non-Zeno signal, therefore it will be constant in $[t'_1, t[$ and $]t, t'_2]$ for some $t_1 < t'_1 < t$ and $t < t'_2 < t_2$ such that $t - t'_1 = t'_2 - t$. Also $F\vec{x}\tau \neq F\vec{x}t$, for $\tau \in [t'_1, t[$, or $F\vec{x}\tau \neq F\vec{x}t$ for $\tau \in]t, t'_2]$.

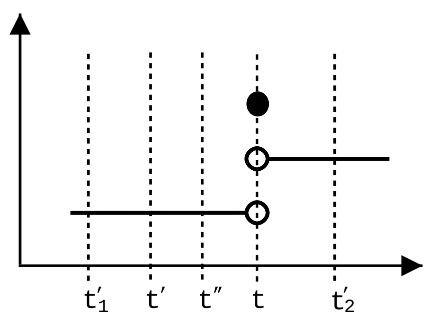

Figure 3.1: $F(x)$

Suppose that $F\vec{x}\tau \neq F\vec{x}t$ for $\tau \in [t'_1, t[$ (see figure 3.2) and let $t', t'' \in [t'_1, t[$ such that $t' < t''$, F_1 be the residual of F with respect to $\vec{x}$ and t' and F_2 be the residual of F with respect to $\vec{x}$ and t'', then by definition 2.1.16 we have

$$\begin{aligned}
F_1(Const_{a_1}, ..., Const_{a_n})(t - t'') &= F(\vec{x})(t' + t - t'') \\
&\neq F(\vec{x})(t) \\
&= F(\vec{x})(t'' + t - t'') \\
&= F_2(Const_{a_1}, ..., Const_{a_n})(t - t'').
\end{aligned}$$

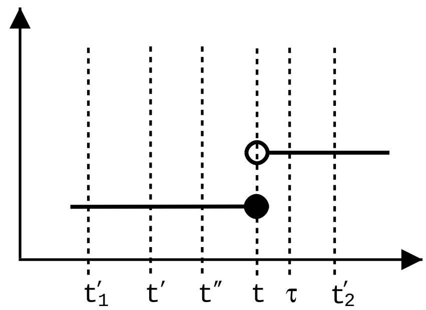

Figure 3.2: $F(x)$

Suppose now that $F\vec{x}\tau \neq F\vec{x}t$, for $\tau \in \,]t, t'_2]$, and that $F\vec{x}\tau' = F\vec{x}t$, for $\tau' \in [t'_1, t[$ (see figure 3.2). Let $t', t'' \in [t'_1, t[$ such that $t' < t''$, F_1 be the residual of F with respect to $\vec{x}$ and t' and F_2 be the residual of F with respect to $\vec{x}$ and t''. By definition 2.1.16 and for $\tau \in \,]t, t'_2]$ such that $\tau - t < t'' - t'$ we have

$$\begin{aligned}
F_1(Const_{a_1}, ..., Const_{a_n})(\tau - t'') &= F(\vec{x})(t' + \tau - t'') \\
&= F(\vec{x})(\tau - (t'' - t')) \\
&\neq F(\vec{x})(\tau) \\
&= F(\vec{x})(t'' + \tau - t'') \\
&= F_2(Const_{a_1}, ..., Const_{a_n})(\tau - t'').
\end{aligned}$$

By the above, given any $t' \in [t'_1, t[$, we know that the residual of F with respect to $\vec{x}$ and t' is different from the residual of F with respect to $\vec{x}$ and t'', for all $t'' \in \,]t'_1, t[$ such that $t' \neq t''$, because $t' > t''$ or $t' < t''$. Then, for each $t' \in [t'_1, t[$ we get a different residual of F with respect to $\vec{x}$ and t', obtaining an uncountable number of residuals of F, i.e., F cannot have countable memory. Hence, if F has countable memory, it is stable. □

We will now study another property of operators over signals, speed independence. We will say that a given operator has this property when it is not sensible to stretching of time.

Definition 3.0.8 (Speed Independence) An operator F from signals to signals is *speed independent* if for every order preserving bijection ρ on time:

$$\forall\, \vec{x},\; F((x_1 \circ \rho), ..., (x_n \circ \rho)) = F(\vec{x}) \circ \rho.$$

The following proposition relates stability and speed independence.

Proposition 3.0.9 *If $F : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$ is speed independent then F is stable.*

Proof [1]: Assume that $\vec{x} \in Sig(\Sigma)$ is constant at $t \in T^+$. Then there exists $\tau_1, \tau_2 \in T^+$ such that $\vec{x}$ is constant in $]\tau_1, \tau_2[$ and $\tau_1 < t < \tau_2$. Let t_1 be an arbitrary point in $]\tau_1, \tau_2[$. Clearly there exists an order preserving bijection $\rho_1 :]\tau_1, \tau_2[\longrightarrow]\tau_1, \tau_2[$ such that $\rho_1(t) = t_1$. Let ρ be the bijection on T^+ defined as

[1] This proof is a slate corrected version of the proof of proposition 2 in [Rab97].

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

It is clear that ρ is an order preserving bijection and that $x_i \circ \rho = x_i$ for $i = 1, ..., n$. Therefore,

$$\begin{aligned} F\vec{x})(t_1) &= F(\vec{x})(\rho \circ \rho^{-1}(t_1)) \\ &= F((x_1 \circ \rho), ..., (x_n \circ \rho))(\rho^{-1}(t_1)) \\ &= F(\vec{x})(t). \end{aligned}$$

Therefore, $F\vec{x}$ is constant in $]\tau_1, \tau_2[$ and F is stable. □

3.1 Speed Independent Operators over Right Open Signals

Let x be a right open signal, by definition 2.2.4 we can see that x is defined univocally by an ω-sequence $\alpha = < a_i : i \in \mathbb{N} >$ over Σ and an unbounded increasing ω-sequence $\tau = < t_i : i \in \mathbb{N} >$ such that $t_i \in T^+$, $t_0 = 0$ and

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

Suppose now that $\vec{x} \in Sig(\Sigma)^n$ is a right open signal, i.e., x_i is a right open signal, for $i = 1, ..., n$. By the above, each x_i is characterized by ω-sequences α_i and τ_i. Let τ be an unbounded increasing ω-sequence such that $\tau_i \subseteq \tau$ and let α_i' be the ω-sequence obtained from α_i by inserting the value of x_i in α_i at each new instant of τ. It is clear that α_i', τ characterize x_i and we note that α_i' may have repetitions. Thus, we can obtain an unbounded increasing ω-sequence $\tau = < t_i : i \in \mathbb{N} >$ such that $\tau_j \subseteq \tau$ for $j = 1, ..., n$, $t_i \in T^+$, $t_0 = 0$ and

$$\forall\ j,\ \forall\ i,\ \forall\ t \in [t_i, t_{i+1}[,\ x_j(t) = a_i^j,$$

where $\alpha = < (a_i^1, ..., a_i^n) : i \in \mathbb{N} >$ is an ω-sequence over Σ^n such that $< a_i^j : i \in \mathbb{N} >$ and τ characterize x_j, for $j = 1, ..., n$.

When the above conditions hold we say that the pair α, τ characterizes $\vec{x}$ or $\vec{x}$ is characterized by α, τ.

Notation 3.1.1 An unbounded increasing sequence $t_0 < t_1 < ...$ such that $t_i \in T^+$ and $t_0 = 0$ is called time scale. Throughout this section τ, τ' will denote time scales and α, β will denote ω-sequences over an alphabet Σ^n. $(\Sigma^n)^\omega$ will denote the set of all ω-sequences over the alphabet Σ^n.

Given an order preserving bijection $\rho : T^+ \to T^+$ and a time scale τ, we state that if $\tau_i' = \rho(\tau_i)$, for all $i \in \mathbb{N}$, then τ' is a time scale. Moreover, α, τ characterizes $\vec{x}$ if and only if α, τ' characterizes $\vec{x} \circ \rho = (x_1 \circ \rho, ..., x_n \circ \rho)$. Clearly, 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 $\vec{x} \circ \rho$.

It is also clear that, if $\vec{x}$ is characterized by α, τ and $\vec{x}$ is not constant at t, then t appears in τ and if τ contains all points at which x is not constant then there exists α such that α, τ characterizes $\vec{x}$. So, if F is a stable operator from right open signals to right open signals and α, τ characterizes $\vec{x}$, then there exists β such that β, τ characterizes $F\vec{x}$.

Assume that F is a speed independent operator from right open signals to right open signals, by proposition 3.0.9 we know that F is stable. Suppose that α, τ characterizes $\vec{x}$ and let β be such that β, τ characterizes $y = F(\vec{x})$, which exists as we saw in the previous paragraph. Since F is speed independent, it follows that for any τ' and for the $\vec{x}'$ characterized by α, τ' the signal $F\vec{x}'$ is characterized by β, τ'.

Therefore, we can associate with every speed independent operator F an operator G from ω-strings over Σ^n to ω-strings over Σ such that, for all α and τ, if α, τ characterizes $\vec{x}$, then $G(\alpha), \tau$ characterizes $F\vec{x}$. In this case G is known as a discrete characterization of F.

However not every G on ω-strings characterizes an operator on right open signals. Indeed, if G characterizes an operator, then, whenever α, τ and α', τ' characterize the same signal, $G\alpha, \tau$ and $G\alpha', \tau'$ should also characterize the same signal. Many distinct α, τ may characterize the same signal. For example, assume that $\alpha = < a_0, ..., a_i, a_{i+1}, ... >$ and $\tau = < t_0, ..., t_i, t_{i+1}, ... >$. Let $t \in]t_i, t_{i+1}[$ and let α' and τ' be defined as $< a_0, ..., a_i, a_i, a_{i+1}, ... >$ and $< t_0, ..., t_i, t, t_{i+1}, ... >$ respectively. Then α, τ characterize $\vec{x}$ if and only if α', τ' characterize $\vec{x}$. Therefore, if G characterizes an operator on right open signals it should satisfy the *SI* [2] *condition*.

Definition 3.1.2 (SI Condition [Rab97]) An operator G on ω-strings satisfies the *SI condition* if for any $< a_0, ...a_i, a_{i+1}, ... >$ and $< b_0, ...b_i, b_{i+1}, ... >$ with $a_i \in \Sigma^n$ and $b_i \in \Sigma$:

$$G(< a_0, ..., a_i, a_{i+1}, ... >) = < b_0, ..., b_i, b_{i+1}, ... >$$
$$\text{if and only if}$$
$$G(< a_0, ..., a_i, a_i, a_{i+1}, ... >) = < b_0, ..., b_i, b_i, b_{i+1}, ... >.$$

The following proposition resume some relevant observations and the given proof is a completion of the proof found in [Rab97].

Proposition 3.1.3 .

1. *Every speed independent operator F that maps right open signals into right open signals is characterized by a function G on ω-strings that satisfies the SI condition.*

2. *Every function G on ω-strings that satisfies the SI condition characterizes a speed independent operator F that maps right open signals into right open signals.*

3. *If G, a function on ω-strings, characterizes F, a speed independent operator that maps right open signals into right open signals, then*

   (a) *G is retrospective if and only if F is retrospective.*

   (b) *G and F have the same number of distinct residuals.*

   (c) *G has finite memory if and only if F has finite memory.*

Proof: **1.** Let F be a speed independent operator that maps right open signals into right open signals. We define $G : (\Sigma^n)^\omega \longrightarrow \Sigma^\omega$ as follows:

---

2 Speed Independence

$$G < (a_i^1, ..., a_i^n) : i \in \mathbb{N} > = < b_i : i \in \mathbb{N} >,$$

where $< b_i : i \in \mathbb{N} >, \tau$ characterizes $F\vec{x}$, for some time scale τ and for $\vec{x}$ characterized by $< (a_i^1, ..., a_i^n) : i \in \mathbb{N} >, \tau$. G is clearly well defined, i.e., given an ω-sequence $< (a_i^1, ..., a_i^n) : i \in \mathbb{N} >, \tau$, the ω-sequence $< b_i : i \in \mathbb{N} >$ is univocally determined because $< (a_i^1, ..., a_i^n) : i \in \mathbb{N} >, \tau$ is a right open signal and, since F is speed independent, $< b_i : i \in \mathbb{N} >$ will not depend on τ (we may choose any time scale). As we have seen in this section, it is clear that G is a characterization of F by definition of G.

Let $< a_0, ..., a_i, a_{i+1}, ... > \in (\Sigma^n)^\omega$ and $< b_0, ..., b_i, b_{i+1}, ... > \in \Sigma^\omega$. If

$$< a_0, ..., a_i, a_{i+1}, ... >, < t_0, ..., t_i, t_{i+1}, ... >$$

characterizes a right open signal x, being $\tau = < t_0, ..., t_i, t_{i+1}, ... >$ a time scale, then

$$< a_0, ..., a_i, a_i, a_{i+1}, ... >, < t_0, ..., t_i, t, t_{i+1}, ... >,$$

with $t_i < t < t_{i+1}$, will characterize also x. As F is an operator, $F\vec{x}$ is univocally determined and we have that

$$G < a_0, ..., a_i, a_{i+1}, ... > \text{ and } G < a_0, ..., a_i, a_i, a_{i+1}, ... >$$

will both characterize the right open signal $F\vec{x}$. Therefore, if

$$G < a_0, ..., a_i, a_{i+1}, ... > = < b_0, ..., b_i, b_{i+1}, ... >,$$

$G < a_0, ..., a_i, a_i, a_{i+1}, ... >$ must be $< b_0, ..., b_i, b_i, b_{i+1}, ... >$ otherwise the sequences will not characterize the same signal for the chosen time scales. The converse implication in definition 3.1.2 follows similarly. Thus, we conclude that G verifies the SI condition.

2. Let G be a function from ω-strings over Σ^n to ω-strings over Σ which satisfies the SI condition. We will consider only ω-strings $< a_i : i \in \mathbb{N} > \in (\Sigma^n)^\omega$ such that

$$\begin{array}{c} \forall\ i,\ a_i \neq a_{i+1} \\ \text{or} \\ \exists\ k,\ \forall\ i < k,\ a_i \neq a_{i+1} \text{ and } \exists\ a \in \Sigma^n,\ \forall\ i \geq k,\ a_i = a. \end{array} \quad (3.1)$$

As G verifies the SI condition, its values on ω-strings that verify the condition 3.1 determine completely its values on any ω-string. Let $\vec{x}$ be a right open signal and ξ the set of instants in which $\vec{x}$ is not continuous, i.e., in which x_i is not continuous for some $1 \leq i \leq n$. We have two possible cases:

1. $\xi = < t_i : i \in \mathbb{N} >$, i.e., ξ is a time scale;
2. $\xi = < t_1, ..., t_k >$.

In first case we may consider the ω-string $\alpha = < (x_1(t_i), ..., x_n(t_i)) : i \in \mathbb{N} >$ which clearly characterizes $\vec{x}$ with the time scale $\tau = \xi$ and verifies the condition 3.1. In second case we extend ξ to the time scale $\tau = < t_0, ..., t_k, t_k+1, t_k+2, ... >$ and we consider the ω-string $\alpha = < (x_1(t), ..., x_n(t)) : t \in \tau >$. It is clear that in this case α also verifies the condition 3.1 and that it also characterizes x with τ as time scale. Let y be the unique right open signal characterized by $G\alpha, \tau$ and we define F as follows:

$$F\vec{x} = y$$

As we have seen, if ρ is an order preserving bijection, then $< \rho(t) : t \in \tau >$ is also a time scale and by definition of F it follows that it is speed independent and we will show now that G characterizes F. Assuming that α', τ' characterize $\vec{x}$, it does not satisfy 3.1 necessarily but we clearly may obtain α, τ from this which characterizes also $\vec{x}$ and verifies 3.1. $G\alpha, \tau$ will characterize $F\vec{x}$ by definition of F and, as G verifies the SI condition, $G\alpha', \tau'$ will also characterize $F\vec{x}$. Therefore, G characterizes F.

3. Suppose that G characterizes F. Let $\alpha, \alpha' \in (\Sigma^n)^\omega$ such that $\alpha(i) = \alpha'(i)$, for $i \leq k$ and τ, τ' be two time scales such that $\tau(i) = \tau'(i)$, for $i \leq k+1$, therefore $\vec{x}$ characterized by α, τ and $\vec{x}'$ characterized by α', τ' are right open signals and they will verify $x_i(t) = x'_i(t)$, for $0 \leq t \leq \tau(k+1)$ and for all i. If F is retrospective by definition 2.1.11,

$$F(\vec{x})(t) = F(\vec{x}')(t), \text{ for } 0 \leq t \leq \tau(k+1),$$

and, since G is a characterization of F, we have

$$(G\alpha)(i) = (G\alpha')(i), \text{ for } i \leq k,$$

i.e., G is retrospective.

Conversely, let $\vec{x}$ and $\vec{x}'$ be two right open signals such that $x_i(t') = x'_i(t')$, for $t' \leq t$ with $t \in T^+$ and for all i. If G is retrospective and these signals are characterized by α, τ and α', τ', then $\alpha(i) = \alpha'(i)$, for $i \leq k$, and $\tau(i) = \tau'(i)$, for $i < k$, with k such that t_{k-1} is the last instant before t in which $\vec{x}$ and $\vec{x}'$ change and we have

$$(G\alpha)(i) = (G\alpha')(i), \text{ for } i \leq k.$$

Since G is a characterization of F, $G\alpha, \tau$ characterizes $F\vec{x}$ and $G\alpha', \tau'$ characterizes $F\vec{x}'$ and by the previous observations we get

$$F(\vec{x})(t') = F(\vec{x}')(t'), \text{ for } t' \leq t,$$

i.e., F is retrospective by definition 2.1.11. Therefore, F is retrospective if and only if G is retrospective.

As we saw in 1 and 2, exists an univocally correspondence between speed independent operators on right open signals and functions on ω-strings that verify the SI condition. Therefore, since the residuals of F still are speed independent and the residuals of G still verify the SI condition, G and F will have the same number of distinct residuals. Clearly, by definition 2.1.17, G has finite memory if and only if F has finite memory. □

Since every retrospective function on ω-strings has at most countable memory and, as we saw in last proposition, all speed independent retrospective operators on right open signals are characterized by retrospective functions on ω-strings, we obtain the following corollary.

Corollary 3.1.4 *Every speed independent retrospective operator on right open signals has at most countable memory.*

3.2 Speed Independent Operators over Non-Zeno Signals

Now we provide a similar description for speed independent operators over non-Zeno signals.

A non-Zeno signal x over an alphabet Σ is said to be characterized by α, α', τ if $\alpha = < a_i : i \in \mathbb{N} >$ and $\alpha' = < a'_i : i \in \mathbb{N} >$ are ω-strings over Σ^n, $\tau = < t_i : i \in \mathbb{N} >$ is a time scale and $x(t_i) = a_i$ and $x(t) = a'_i$ for every $i \in \mathbb{N}$ and every $t \in]t_i, t_{i+1}[$. We 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 by definition 2.2.1.

Suppose now that $\vec{x} \in Sig(\Sigma)^n$ is a non-Zeno signal, i.e., x_i is a non-Zeno signal, for $i = 1, ..., n$. By the above, each x_i is characterized by ω-sequences α_i, α'_i and τ_i. Let τ be an unbounded increasing ω-sequence such that $\tau_i \subseteq \tau$ and let β_i and β'_i be the ω-sequences obtained from α_i and α'_i by inserting the necessary values of x_i in α_i and α'_i for each new instant of τ. It is clear that β, β'_i, τ characterize x_i and we note that β_i and β'_i may have repetitions. Thus, we can obtain an unbounded increasing ω-sequence $\tau = < t_i : i \in \mathbb{N} >$ such that $\tau_j \subseteq \tau$ for $j = 1, ..., n$, $t_i \in T^+$, $t_0 = 0$ and

$$\forall\ j,\ \forall i,\ x_j(t_i) = a_i^j \text{ and } \forall\ t \in]t_i, t_{i+1}[,\ x_j(t) = a_i'^j,$$

where $\beta = < (a_i^1, ..., a_i^n) : i \in \mathbb{N} >$ and $\beta' = < (a_i'^1, ..., a_i'^n) : i \in \mathbb{N} >$ are ω-sequences over Σ^n such that $< a_i^j : i \in \mathbb{N} >, < a_i'^j : i \in \mathbb{N} >, \tau$ characterize x_j, for $j = 1, ..., n$.

When the above conditions hold we say that the pair α, τ characterizes $\vec{x}$ or $\vec{x}$ is characterized by α, τ.

An operator F from non-Zeno signals over Σ_1 to non-Zeno signals over Σ_2 is said to be characterized by a function $G : (\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega \longrightarrow \Sigma_2^\omega \times \Sigma_2^\omega$ if whenever α, α', τ characterizes $\vec{x}$ then $G(\alpha, \alpha'), \tau$ characterize $F\vec{x}$.

As we will see, every speed independent operator is characterized by a function on ω-strings but not every function $G : (\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega \longrightarrow \Sigma_2^\omega \times \Sigma_2^\omega$ characterizes a speed independent operator. Let $\alpha = < a_0, ..., a_i, a_{i+1}, ... >$ and $\alpha' = < a'_0, ..., a'_i, a'_{i+1}, ... >$ with $a_i, a'_i \in \Sigma_1^n$ be ω-strings and $\tau = < t_0, ...t_i, t_{i+1}, ... >$ be a time scale. Assume that β, β' are obtained from α, α' by inserting a'_i after the i position, i.e., $\beta = < a_0, ..., a_i, a'_i, a_{i+1}, ... >$ and $\beta' = < a'_0, ..., a'_i, a'_i, a'_{i+1}, ... >$, and that τ' is obtained from τ by inserting any $t \in]\tau_i, \tau_{i+1}[$ after the i position, i.e., $\tau' = < t_0, ..., t_i, t, t_{i+1}, ... >$. Then α, α', τ characterize $\vec{x}$ if and only if β, β', τ' characterize $\vec{x}$.

Hence, if G characterizes a speed independent operator it should satisfy the following condition.

Definition 3.2.1 (Generalized SI Condition [Rab97]) Let α and β be ω-strings and let $i \in \mathbb{N}$. Let β and β' be obtained from α and α' by inserting a'_i after i as above. Similarly, let ζ and ζ' be obtained from ξ and ξ' by inserting x'_i after i. Then:

$$G(\alpha, \alpha') = (\xi, \xi')$$
$$\text{if and only if}$$
$$G(\beta, \beta') = (\zeta, \zeta').$$

Assume that α, α', τ characterizes $\vec{x}_1$ and that β, β', τ characterizes $\vec{x}_2$. Then $\vec{x}_1$ is equal to $\vec{x}_2$ in $[0, t]$ if either $t \in]t_i, t_{i+1}[$ and $a_j = b_j$ and $a'_j = b'_j$ for $j \in 0, ..., i$ or $t = t_i$ and $a_j = b_j$ for $j \in \{0, ..., i\}$ and $a'_j = b'_j$ for $j \in \{0, ..., i-1\}$. Hence, if G characterizes a retrospective operator F, then G should satisfy the following condition.

Definition 3.2.2 (Generalized Retrospective Condition [Rab97]) Let G be a function on $(\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega$ and let

$$\begin{aligned}
\alpha &= < a_i : i \in \mathbb{N} >, \\
\alpha' &= < a'_i : i \in \mathbb{N} >, \\
\beta &= < b_i : i \in \mathbb{N} >, \\
\beta' &= < b'_i : i \in \mathbb{N} >, \\
\xi &= < x_i : i \in \mathbb{N} >, \\
\xi' &= < x'_i : i \in \mathbb{N} >, \\
\zeta &= < z_i : i \in \mathbb{N} >, \\
\zeta' &= < z'_i : i \in \mathbb{N} >
\end{aligned}$$

be ω-strings. Then G satisfies the *generalized retrospective condition* if

1. G is retrospective;
2. if $G(\alpha, \alpha') = (\xi, \xi')$, $G(\beta, \beta') = (\zeta, \zeta')$, $a_j = b_j$ for $j \in \{0, ..., i\}$ and $a'_j = b'_j$ for $j \in \{0, ..., i-1\}$, then $x_j = z_j$ for $j \in \{0, ..., i\}$ and $x'_j = z'_j$ for $j \in \{0, ..., i-1\}$.

The following proposition resume some relevant observations and it was stated (but not proved) in [Rab97].

Proposition 3.2.3 ([Rab97]) .

1. *Every speed independent operator F over non-Zeno signals is characterized by a function G on $(\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega$ that satisfies the generalized SI condition.*
2. *Every function G on $(\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega$ that satisfies the generalized SI condition characterizes a speed independent operator F over non-Zeno signals.*
3. *If G, a function on $(\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega$, characterizes F, a speed independent operator over non-Zeno signals, then:*
   (a) *F is retrospective is and only if G satisfies the generalized retrospective condition;*
   (b) *F has finite memory if and only if G has finite memory;*
   (c) *F has countable memory if and only if G has countable memory.*

Proof: **1.** Let F be a speed independent operator that maps non-Zeno signals into non-Zeno signals. We define $G : (\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega \longrightarrow \Sigma_2^\omega \times \Sigma_2^\omega$ as follows:

$$G(\alpha, \alpha') = (\beta, \beta'),$$

where α, α', τ characterizes a signal $\vec{x}$ and β, β', τ characterizes $F\vec{x}$, for some time scale τ. G is clearly well defined, i.e, given ω-sequences α, α' and τ, the ω-sequences β and β' are univocally determined because α, α', τ is a non-Zeno signal and, since F is speed independent, β and β' will not depend on τ. As we have seen in this section, it is clear that G is a characterization of F by definition of G.

Let

$$< a_0, ...a_i, a_{i+1}, ... >, < a'_0, ...a'_i, a'_{i+1}, ... > \in (\Sigma_1^n)^\omega$$

and

$$< b_0, ...b_i, b_{i+1}, ... >, < b'_0, ...b'_i, b'_{i+1}, ... > \in \Sigma_2^\omega.$$

If $< a_0, ...a_i, a_{i+1}, ... >, < a'_0, ...a'_i, a'_{i+1}, ... >, < t_0, ..., t_i, t_{i+1}, ... >$ characterizes a non-Zeno signal x, being $\tau = < t_0, ..., t_i, t_{i+1}, ... >$ a time scale, then

$$< a_0, ...a_i, a'_i, a_{i+1}, ... >, < a'_0, ...a'_i, a'_i, a'_{i+1}, ... >, < t_0, ..., t_i, t, t_i + 1, ... >,$$

being $t_i < t < t_{i+1}$, will characterize also x. Since F is an operator, $F\vec{x}$ is univocally determined and we have that

$$\begin{gathered} G(< a_0, ...a_i, a_{i+1}, ... >, < a'_0, ...a'_i, a'_{i+1}, ... >) \\ \text{and} \\ G(< a_0, ...a_i, a'_i, a_{i+1}, ... >, < a'_0, ...a'_i, a'_i, a'_{i+1}, ... >) \end{gathered}$$

will both characterize the non-Zeno signal $F\vec{x}$. Therefore, if

$$\begin{gathered} G(< a_0, ...a_i, a_{i+1}, ... >, < a'_0, ...a'_i, a'_{i+1}, ... >) \\ = \\ (< b_0, ...b_i, b_{i+1}, ... >, < b'_0, ...b'_i, b'_{i+1}, ... >), \end{gathered}$$

we must have

$$\begin{gathered} G(< a_0, ...a_i, a'_i, a_{i+1}, ... >, < a'_0, ...a'_i, a'_i, a'_{i+1}, ... >) \\ = \\ < b_0, ...b_i, b'_i, b_{i+1}, ... >, < b'_0, ...b'_i, b'_i, b'_{i+1}, ... > \end{gathered}$$

otherwise they will not characterize the same signal for the chosen time scales. The converse implication in definition 3.2.2 follows similarly. Thus, we conclude that G verifies the generalized SI condition.

2. Let $G : (\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega \longrightarrow \Sigma_2^\omega \times \Sigma_2^\omega$ be a function over ω-strings which satisfies the generalized SI condition. We will consider only ω-strings

$$(< a_i : i \in \mathbb{N} >, < a'_i : i \in \mathbb{N} >) \in (\Sigma^n)^\omega \times (\Sigma^n)^\omega$$

such that

$$\begin{gathered} \forall\ i,\ (a_i, a'_i) \neq (a_{i+1}, a'_{i+1}) \\ \text{or} \\ \exists\ k,\ \forall\ i < k,\ (a_i, a'_i) \neq (a_{i+1}, a'_{i+1}) \\ \text{and } \exists\ a, a' \in \Sigma^n,\ \forall\ i \geq k,\ (a_i, a'_i) = (a, a'). \end{gathered} \tag{3.2}$$

As G verifies the generalized SI condition, its values on ω-strings that verify the condition 3.2 determine completely its values on any ω-string. Let $\vec{x}$ be non-Zeno signal and ξ the set of instants in which $\vec{x}$ is not continuous, i.e., in which x_i is not continuous for some $1 \leq i \leq n$. We have two possible cases:

1. $\xi = < t_i : i \in \mathbb{N} >$, i.e., ξ is a time scale;

2. $\xi = < t_1, ..., t_k >$.

In first case we may consider the ω-strings

$$\alpha = < (x_1(t_i), ..., x_n(t_i)) : i \in \mathbb{N} >$$
$$\alpha' = < (x_1(t), ..., x_n(t)) : i \in \mathbb{N} \textit{ and } t_i < t < t_{i+1} >$$

which clearly characterizes $\vec{x}$ with the time scale $\tau = \xi$ and verifies the condition 3.2. In second case we extend ξ to the time scale $\tau = < t_0, ..., t_k, t_k{+}1, t_k{+}2, ... >$ and we consider the ω-strings

$$\alpha = < (x_1(t), ..., x_n(t)) : t \in \tau >$$
$$\alpha' = < (x_1(t), ..., x_n(t)) : t_i \in \tau \textit{ and } t_i < t < t_{i+1} >.$$

It is clear that in this case α, α' also verifies the condition 3.2 and that it also characterizes x with τ as time scale. Let y be the unique right open signal characterized by $G(\alpha, \alpha'), \tau$ and we define F as follows:

$$F\vec{x} = y.$$

As we have seen, if ρ is an order preserving bijection, $< \rho(t) : t \in \tau >$ is also a time scale and by definition of F it follows that it is speed independent and we will show now that G characterizes F. Assumes that β, β', τ' characterizes $\vec{x}$, it does not satisfy 3.2 necessarily but we clearly may obtain α, α', τ from this which characterizes also $\vec{x}$ and verifies 3.2. $G(\alpha, \alpha'), \tau$ will characterize $F\vec{x}$ by definition of F and, as G verifies the generalized SI condition, $G(\beta, \beta'), \tau'$ will also characterize $F\vec{x}$. Therefore, G characterizes F.

3. Suppose that G characterizes F. Let $\alpha, \alpha', \beta, \beta' \in (\Sigma^n)^\omega$ such that $\alpha(i) = \beta(i)$, for $i \leq k$, and $\alpha'(i) = \beta'(i)$, for $i < k$, and τ, τ' be two time scales such that $\tau(i) = \tau'(i)$, for $i \leq k$, therefore $\vec{x}$ characterized by α, α', τ and $\vec{x}'$ characterized by β, β', τ' are non-Zeno signals and they will verify $x_i(t) = x'_i(t)$, for $0 \leq t \leq \tau(k)$ and for all i. If F is retrospective by definition 2.1.11,

$$F(\vec{x})(t) = F(\vec{x}')(t), \text{ for } 0 \leq t \leq \tau(k)$$

and as G is a characterization of F, for ξ, ξ' and ζ, ζ' such that $G(\alpha, \alpha') = (\xi, \xi')$ and $G(\beta, \beta') = (\zeta, \zeta')$, we have

$$\xi(i) = \zeta(i), \text{ for } i \leq k, \text{ and } \xi'(i) = \zeta'(i), \text{ for } i < k,$$

i.e., G is retrospective.

Let $\vec{x}$ and $\vec{x}'$ be two non-Zeno signals such that $x_i(t') = x'_i(t')$ for $t' \leq t$ with $t \in T^+$ and all i. If G is retrospective and these signals are characterized by α, α', τ and β, β', τ', then $\alpha(i) = \beta(i)$, for $i \leq k$, and $\alpha'(i) = \beta'(i)$, for $i < k$, with k such that $t_k \geq t$, and so, for ξ, ξ' and ζ, ζ' such that $G(\alpha, \alpha') = (\xi, \xi')$ and $G(\beta, \beta') = (\zeta, \zeta')$, we have

$$\xi(i) = \zeta(i), \text{ for } i \leq k, \text{ and } \xi'(i) = \zeta'(i), \text{ for } i < k.$$

Since G is a characterization of F, $G(\alpha, \alpha'), \tau$ characterizes $F\vec{x}$ and $G(\beta, \beta'), \tau'$ characterizes $F\vec{x}'$ and by the previous observations we get that

$$F(\vec{x})(t') = F(\vec{x}')(t'), \text{ for } t' \leq t,$$

i.e., F is retrospective by definition 2.1.11. Therefore, F is retrospective if and only if G is retrospective.

As we saw in 1 and 2, there exists an univocally correspondence between speed independent operators on non-Zeno signals and functions on ω-strings that verify the generalized SI condition. Therefore, since the residuals of F still be speed independent and the residuals of G still verify the generalized SI condition, G and F will have the same number of distinct residuals. Clearly, by definition 2.1.17, G has finite memory if and only if F has finite memory. □

Since every retrospective function on ω-strings has at most countable memory and, as we saw in the last proposition, all speed independent retrospective operators on non-Zeno signals are characterized by retrospective functions on ω-strings, we obtain the following corollary.

Corollary 3.2.4 *Every speed independent retrospective operator on non-Zeno signals has at most countable memory.*

Results of this chapter

3.0.6 A stable operator maps non-Zeno signals to non-Zeno signals.

3.0.7 Every operator that maps non-Zeno signals to non-Zeno signals with countable memory is stable.

3.0.9 If $F : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$ is speed independent then F is stable.

3.1.3
1. Every speed independent operator F that maps right open signals into right open signals is characterized by a function G on ω-strings that satisfies the SI condition.
2. Every function G on ω-strings that satisfies the SI condition characterizes a speed independent operator F that maps right open signals into right open signals.
3. If G, a function on ω-strings, characterizes F, a speed independent operator that maps right open signals into right open signals, then
   (a) G is retrospective if and only if F is retrospective.
   (b) G and F have the same number of distinct residuals.
   (c) G has finite memory if and only if F has finite memory.

3.1.4 Every speed independent retrospective operator on right open signals has at most countable memory.

3.2.3
1. Every speed independent operator F over non-Zeno signals is characterized by a function G on $(\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega$ that satisfies the generalized SI condition.
2. Every function G on $(\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega$ that satisfies the generalized SI condition characterizes a speed independent operator F over non-Zeno signals.
3. If G, a function on $(\Sigma_1^n)^\omega \times (\Sigma_1^n)^\omega$, characterizes F, a speed independent operator over non-Zeno signals, then:
   (a) F is retrospective is and only if G satisfies the generalized retrospective condition;
   (b) F has finite memory if and only if G has finite memory;
   (c) F has countable memory if and only if G has countable memory.

3.2.4 Every speed independent retrospective operator on non-Zeno signals has at most countable memory.

Chapter 4

Examples of Operators

In this chapter we gather a considerable number of examples of operators on signals[1] found in [Rab97, PRT01]. However discussed there, these examples were not studied in detail and some mistakes have been committed. We will study all examples carefully, each one will be classified according to the properties studied in previous chapters, namely we will verify if each one is:

1. speed independent;
2. stable;
3. mapping non-Zeno signals into non-Zeno signals;
4. strong retrospective;
5. retrospective;
6. finite memory;
7. countable memory.

We note that in these examples the time set T will be considered as $\mathbb{R}$, i.e., we will study operators that receive real-time signals.

Example 4.0.1 (Signal $Jump_{a\to b}$ [Rab97]) $Jump_{a\to b}$ is the signal, i.e, the 0-ary operator on signals, defined as follows:

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

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection, therefore:

$$Jump_{a\to b}(t) = Jump_{a\to b}(\rho(t)),$$

for every $t \in T^+$, i.e, $Jump_{a\to b}$ is speed independent by definition 3.0.8.

2. By 1 and by proposition 3.0.9, we know that $Jump_{a\to b}$ is stable.

3. Since $Jump_{a\to b}$ is a 0-ary operator on signals, we only need to verify if it is a non-Zeno signal, which is true by definition 2.2.1.

[1] Not necessarily non-Zeno signals.

4. We know that $Jump_{a \to b}$ has no arguments, therefore it is strong retrospective by definition 2.1.11.

5. $Jump_{a \to b}$ is retrospective by 4.

6. Accordingly with definition 2.1.16, $Jump_{a \to b}$ has the following residuals:

$$\{suf(Jump_{a \to b}, t) : t \in T^+\} = \{Jump_{a \to b}, Const_b\}.$$

By definition 2.1.17, $Jump_{a \to b}$ has finite memory.

7. By 6, it has countable memory.

Example 4.0.2 (Signal *Rational* [Rab97]) *Rational* is the signal, i.e, the 0-ary operator on signals, defined as follows:

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

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection such that

$$\rho(1) = \sqrt{2},$$

therefore:

$$Rational(1) = True \neq False = Rational(\sqrt{2}) = Rational(\rho(1)),$$

i.e.,

$$Rational \neq Rational \circ \rho.$$

So, *Rational* is not speed independent by definition 3.0.8.

2. *Rational* is not constant for any $t > 0$, therefore it is not stable by definition 3.0.5.

3. Since *Rational* is a 0-ary operator on signals, we only need to verify if it is a non-Zeno signal, which is false by definition 2.2.1 and because *Rational* is not constant for any $t > 0$.

4. By definition 2.1.11, *Rational* is clearly a strong retrospective operator.

5. *Rational* is retrospective by 4.

6. Since *Rational* is a signal, the set of residuals is the set of the possible suffixes. Let $q, q' \in T^+$ be irrationals and $r \in T^+$ be a rational such that $r > q$, $q' > q$ and $q' - q$ is not rational, therefore

$$\begin{aligned} Rational(q) &= False, \\ Rational(q') &= False, \\ Rational(r) &= True \end{aligned}$$

and we will see that $suf(Rational, q)$ and $suf(Rational, q')$ are different for $r - q$. Clearly:

$$suf(Rational, q)(r - q) = Rational(r - q + q) = True,$$

however,

$$suf(Rational, q')(r - q) = Rational(r - q + q') = False,$$

where $r - q + q'$ is irrational because $q' - q$ is irrational. Therefore, since the set of irrationals is uncountable, the set of irrationals q, q' such that $q' - q$ is irrational is also uncountable and then the set of suffixes is uncountable, i.e., *Rational* has not finite memory by definition 2.1.17.

7. As we have seen in 6, the set of prefixes is uncountable and then *Rational* has not countable memory.

Example 4.0.3 (Operator $\exists$ [Rab97, PRT01]) The unary operator $\exists$ over boolean signals is defined as follows:

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

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection. Thus, if $x(t) = True$ for some $t \in T^+$, there exists t' such that $\rho(t') = t$ and therefore $(x \circ \rho)(t') = True$. Then

$$\exists(x \circ \rho)(t) = \exists(x)(\rho(t)),$$

for every $t \in T^+$. If $x(t) = False$, for every $t \in T^+$, then

$$\exists(x \circ \rho)(t) = False = \exists(x)(\rho(t)),$$

for every $t \in T^+$. Therefore, $\exists$ is speed independent by definition 3.0.8.

2. By 1 and by proposition 3.0.9, we know that $\exists$ is stable.

3. By 2 and by proposition 3.0.6, we know that $\exists$ maps non-Zeno signals into non-Zeno signals.

4. Looking at definition of $\exists$, we see that the value at t may depend on t' with $t' > t$. Therefore $\exists$ is not strong retrospective by definition 2.1.11.

5. By what we have seen in 4, $\exists$ can not be retrospective.

6. Accordingly with definition 2.1.16, if x is a boolean signal and $t \in T^+$, the residual of $\exists$ with respect to x and t is one of the following operators:

- if $x(t') = False$ for t' such that $0 \le t' < t$,

$$\lambda z \lambda \tau.\ \exists(x)(\tau);$$

- if exists t' such that $0 \le t' < t$ and $x(t') = True$,

$$\lambda z \lambda \tau.\ True.$$

Therefore, $\exists$ has finite memory by definition 2.1.17.

7. By 6, it has countable memory.

Example 4.0.4 (Operator *LeftCont* [Rab97]) *LeftCont* is an unary operator on signals defined as follows:

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

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection,

$$\begin{aligned}
LeftCont(x \circ \rho)(t) &= \begin{cases} True & \text{if } x \circ \rho \text{ is left continuous at } t \\ False & \text{otherwise} \end{cases} \\
&= \begin{cases} True & \text{if } x \text{ is left continuous at } \rho(t) \\ False & \text{otherwise} \end{cases} \\
&= LeftCont(x)(\rho(t)) \\
&= (LeftCont(x) \circ \rho)(t).
\end{aligned}$$

Therefore, $LeftCont$ is speed independent by definition 3.0.8.

2. By 1 and by proposition 3.0.9, we know that $LeftCont$ is stable.

3. By 2 and by proposition 3.0.6, we know that $LeftCont$ maps non-Zeno signals into non-Zeno signals.

4. Given a signal x, we need to know its value at t in order to decide about the left continuity of x at t. Therefore, $LeftCont$ is not strong retrospective by definition 2.1.11.

5. As we see in 4, we need to know the value of x at t and, clearly, its value before t. Thus, since we do not need future values, $LeftCont$ is retrospective by definition 2.1.11.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$, the residual of $LeftCont$ with respect to x and t is one of the following operators:

- if $t = 0$,

$$\lambda z \lambda \tau.\ LeftCont(z)(\tau);$$

- if $t > 0$ and $\exists\, t' < t,\ \forall\, u \in [t', t[,\ x(u) = a$,

$$\lambda z \lambda \tau. \begin{cases} True & \text{if } \tau = 0 \text{ and } z(0) = a \\ False & \text{if } \tau = 0 \text{ and } z(0) \neq a \\ LeftCont(z)(\tau) & \text{if } \tau > 0; \end{cases}$$

- if $t > 0$ and no $\exists\, t' < t,\ \forall\, u \in [t', t[,\ x(u) = a$,

$$\lambda z \lambda \tau. \begin{cases} False & \text{if } \tau = 0 \\ LeftCont(z)(\tau) & \text{if } \tau > 0. \end{cases}$$

Since x is defined over a finite alphabet, $LeftCont$ has a finite number of residuals, i.e., it has finite memory by definition 2.1.17.

7. By 6, it has countable memory.

Example 4.0.5 (Operator *Cont* [Rab97]) The unary operator $Cont$ on signals is defined as follows:

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

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection,

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

Therefore, *Cont* is speed independent by definition 3.0.8.

2. By 1 and by proposition 3.0.9, we know that *Cont* is stable.

3. By 2 and by proposition 3.0.6, we know that *Cont* maps non-Zeno signals into non-Zeno signals.

4. Given a signal x, we need to know its value at t in order to decide about the continuity of x at t. Therefore, *Cont* is not strong retrospective by definition 2.1.11.

5. As we see in 4, we need to know the value of x at t and, clearly, its value before and after t. Thus, since we need future values, *Cont* is not retrospective by definition 2.1.11.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$, the residual of *Cont* with respect to x and t is one of the following operators:

- if $t = 0$,

$$\lambda z \lambda \tau.\ Cont(z)(\tau);$$

- if $t > 0$ and $\exists\, t' < t,\ \forall\, u \in [t', t[,\ x(u) = a$,

$$\lambda z \lambda \tau. \begin{cases} True & \text{if } \tau = 0,\ a = z(0) \text{ and} \\ & z \text{ is right continuous at } 0 \\ False & \text{if } \tau = 0 \text{ and} \\ & (a \neq z(0) \text{ or } z \text{ is not right continuous at } 0) \\ Cont(z)(\tau) & \text{if } \tau > 0; \end{cases}$$

- if $t > 0$ and no $\exists\, t' < t,\ \forall\, u \in [t', t[,\ x(u) = a$,

$$\lambda z \lambda \tau. \begin{cases} False & \text{if } \tau = 0 \\ Cont(z)(\tau) & \text{if } \tau > 0. \end{cases}$$

Since x is defined over a finite alphabet, *Cont* has a finite number of residuals, i.e., *Cont* has finite memory by definition 2.1.17.

7. By 6, *Cont* has countable memory.

Example 4.0.6 (Operator *LLim* [Rab97]) The unary operator *LLim* on signals is defined as follows:

$$LLim(x)(t) = \begin{cases} a & \text{if } \exists\, t' < t,\ \forall\, u \in [t', t[,\ x(u) = a \\ \perp & \text{otherwise} \end{cases}$$

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection,

$$\begin{aligned} LLim(x \circ \rho)(t) &= \begin{cases} a & \text{if } \exists\, t' < t,\ \forall\, u \in [t', t[,\ x(\rho(u)) = a \\ \perp & \text{otherwise} \end{cases} \\ &= \begin{cases} a & \text{if } \exists\, t',\ \rho(t') < \rho(t) \\ & \quad \text{and } \forall\, v \in [\rho(t'), \rho(t)[,\ x(v) = a \\ \perp & \text{otherwise} \end{cases} \\ &= \begin{cases} a & \text{if } \exists\, t'' < \rho(t),\ \forall\, v \in [t'', \rho(t)[,\ x(v) = a \\ \perp & \text{otherwise} \end{cases} \\ &= LLim(x)(\rho(t)) \\ &= (LLim(x) \circ \rho)(t). \end{aligned}$$

Therefore, $LLim$ is speed independent by definition 3.0.8.

2. By 1 and by proposition 3.0.9, we know that $LLim$ is stable.

3. By 2 and by proposition 3.0.6, we know that $LLim$ maps non-Zeno signals into non-Zeno signals.

4. Given a signal x and an instant $t \in T^+$, by the definition of $LLim$, we conclude that its value at t depends only on instants before t. Therefore, $LLim$ is strong retrospective by definition 2.1.11.

5. By 4 and by definition 2.1.11, $LLim$ is retrospective.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$, the residual of $LLim$ with respect to x and t is one of the following operators:

- if $t = 0$,

$$\lambda z \lambda \tau.\ LLim(z)(\tau);$$

- if $t > 0$ and $\exists\ t' < \ t,\ \forall\ u \in [t', t[,\ x(u) = a$,

$$\lambda z \lambda \tau. \begin{cases} a & \text{if } \tau = 0 \\ LLim(z)(\tau) & \text{if } \tau > 0; \end{cases}$$

- if $t > 0$ and no $\exists\ t' < \ t,\ \forall\ u \in [t', t[,\ x(u) = a$,

$$\lambda z \lambda \tau. \begin{cases} \bot & \text{if } \tau = 0 \\ LLim(z)(\tau) & \text{if } \tau > 0. \end{cases}$$

Since x is defined over a finite alphabet, $LLim$ has a finite number of residuals, i.e., it has finite memory by definition 2.1.17.

7. By 6, it has countable memory.

Example 4.0.7 (Operator $RLim$ [Rab97]) The operator $RLim$ on signals is defined as follows:

$$RLim(x)(t) = \begin{cases} a & \text{if } \exists\ t' > \ t,\ \forall\ u \in]t, t'],\ x(u) = a \\ \bot & \text{otherwise.} \end{cases}$$

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection,

$$\begin{aligned} RLim(x \circ \rho)(t) &= \begin{cases} a & \text{if } \exists\ t' > \ t,\ \forall\ u \in]t, t'],\ x(\rho(u)) = a \\ \bot & \text{otherwise} \end{cases} \\ &= \begin{cases} a & \text{if } \exists\ t', \rho(t') > \rho(t) \\ & \quad \text{and } \forall\ v \in]\rho(t), \rho(t')],\ x(v) = a \\ \bot & \text{otherwise} \end{cases} \\ &= \begin{cases} a & \text{if } \exists\ t'' > \rho(t),\ \forall\ v \in]\rho(t), t''],\ x(v) = a \\ \bot & \text{otherwise} \end{cases} \\ &= RLim(x)(\rho(t)) \\ &= (RLim(x) \circ \rho)(t). \end{aligned}$$

Therefore, $RLim$ is speed independent by definition 3.0.8.

2. By 1 and by proposition 3.0.9, we know that $RLim$ is stable.

3. By 2 and by proposition 3.0.6, we know that $RLim$ maps non-Zeno signals into non-Zeno signals.

4. Given a signal x and an instant $t \in T^+$, by the definition of $RLim$, we conclude that its value at t depends on instants after t. Therefore, $RLim$ is not strong retrospective by definition 2.1.11.

5. Given a signal x and an instant $t \in T^+$, by the definition of $RLim$, we conclude that its value at t depends on instants after t. Therefore, $RLim$ is not retrospective by definition 2.1.11.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$, the residual of $RLim$ with respect to x and t is itself. Therefore $RLim$ has a finite number of residuals, i.e., $RLim$ has finite memory by definition 2.1.17.

7. By 6, $RLim$ has countable memory.

Example 4.0.8 (Pointwise extension [Rab97, PRT01]) Given a function $g : \Sigma_1 \times ... \times \Sigma_n \longrightarrow \Sigma$, its pointwise extension G is the n-ary operator on signals defined as follows:

$$G(x_1, ..., x_n)(t) \ = \ g(x_1(t), ..., x_n(t)).$$

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection,

$$\begin{aligned} G(x_1 \circ \rho, ..., x_n \circ \rho)(t) &= g(x_1(\rho(t)), ..., x_n(\rho(t))) \\ &= G(x_1, ..., x_n)(\rho(t)) \\ &= (G(x_1, ..., x_n) \circ \rho)(t). \end{aligned}$$

Therefore, G is speed independent by definition 3.0.8.

2. By 1 and by proposition 3.0.9, we know that G is stable.

3. By 2 and by proposition 3.0.6, we know that G maps non-Zeno signals into non-Zeno signals.

4. Given a signal $\vec{x}$ and an instant $t \in T^+$, by the definition of G, we conclude that its value at t depends on the value of $\vec{x}$ at t. Therefore, G is not strong retrospective by definition 2.1.11.

5. Given a signal $\vec{x}$ and an instant $t \in T^+$, by the definition of G, we conclude that its value at t depends only on the value of $\vec{x}$ at t. Therefore, G is retrospective by definition 2.1.11.

6. Accordingly with definition 2.1.16, if $\vec{x}$ is a signal and $t \in T^+$, the residual of G with respect to $\vec{x}$ and t is itself. Therefore G has a finite number of residuals, i.e., G has finite memory by definition 2.1.17.

7. By 6, G has countable memory.

Example 4.0.9 (Operator *Prime* [Rab97]) The unary operator *Prime* on signals is defined as:

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

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection,

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

Therefore, *Prime* is speed independent by definition 3.0.8.

2. By 1 and by proposition 3.0.9, we know that *Prime* is stable.

3. By 2 and by proposition 3.0.6, we know that *Prime* maps non-Zeno signals into non-Zeno signals.

4. Given a signal x and an instant $t \in T^+$, by the definition of *Prime*, we conclude that its value at t depends only on instants before t. Therefore, *Prime* is strong retrospective by definition 2.1.11.

5. By 4 and by definition 2.1.11, *Prime* is retrospective.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$, the residual of *Prime* with respect to x and t is one of the following operators:

- if $t = 0$,

$$\lambda z \lambda \tau.\ Prime(z)(\tau);$$

- if $t > 0$ and n is the number of changes of x in $[0, t[$,

$$\lambda z \lambda \tau. \begin{cases} True & \text{if } x\rfloor^t; z \text{ changes at } t \text{ and} \\ & \exists\ p \text{ prime, } z \text{ changes } p - n - 1 \text{ times in } [0, t'[\\ True & \text{if } x\rfloor^t; z \text{ does not change at } t \text{ and} \\ & \exists\ p \text{ prime, } z \text{ changes } p - n \text{ times in } [0, t'[\\ False & \text{otherwise.} \end{cases}$$

Since $\mathbb{N}_0$ is countable, *Prime* has not a finite number of residuals, i.e., *Prime* has not finite memory by definition 2.1.17.

7. By 6, *Prime* has countable memory since $\mathbb{N}_0$ is countable.

Example 4.0.10 (Operator *Timer* [Rab97, PRT01]) The unary operator *Timer* on signals is defined as follows:

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

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection and x a signal such that

$$\rho(1) = \tfrac{1}{2} \text{ and } x(t) = \begin{cases} 0 & \text{if } 0 \leq t < \frac{1}{2} \\ 1 & \text{otherwise.} \end{cases}$$

Therefore,

$$\begin{aligned} Timer(x \circ \rho)(1) &= True \\ &\neq False \\ &= Timer(x)(\frac{1}{2}) \\ &= (Timer(x) \circ \rho)(1), \end{aligned}$$

i.e., $Timer$ is not speed independent by definition 3.0.8.

2. Clearly

$$Timer(Const_a)(t) = False,$$

for every $t < 1$, and

$$Timer(Const_a)(t) = True,$$

for every $t \geq 1$, therefore it is not constant at $t = 1$ while $Const_a$ is constant at $t = 1$. By definition 3.0.5, we know that $Timer$ is not stable.

3. $Timer$ change at most twice in any interval of length one. Therefore, $Timer$ maps non-Zeno signals into non-Zeno signals.

4. Given a signal x and an instant $t \in T^+$, by the definition of $Timer$, we conclude that its value at t depends only on instants before t. Therefore, $Timer$ is strong retrospective by definition 2.1.11.

5. By 4 ad by definition 2.1.11, $Timer$ is retrospective.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$ such that x is constant in $[0, 1[$ and $t < 1$, the residual of $Timer$ with respect to x and t is one of the following operators:

- if $t = 0$,

$$\lambda z \lambda \tau.\ Timer(z)(\tau);$$

- if $t > 0$ and with $a = x(0)$,

$$\lambda z \lambda \tau. \begin{cases} True & \text{if } z(t') = a \text{ for } t' \in [0, \tau[\text{ and } \tau \geq 1 - t \\ Timer(z)(\tau) & \text{otherwise.} \end{cases}$$

As the number of such operators is uncountable (for each $t \in [0, 1[$, there exists a different operator), it follows that $Timer$ has not a finite number of residuals, i.e., $Timer$ has not finite memory by definition 2.1.17.

7. By 6, $Timer$ also has not countable memory.

Example 4.0.11 (Operator $Delay_a$ [Rab97, PRT01]) The unary operator $Delay_a$ on signals is defined as follows:

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

1. Let $a, b \in \Sigma$ such that $a \neq b$ and let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection such that

$$\rho(\tfrac{1}{2}) = 1.$$

Therefore,

$$\begin{aligned}
Delay_a(Const_b \circ \rho)(\frac{1}{2}) &= a \\
&\neq b \\
&= Delay_a(Const_b)(1) \\
&= (Delay_a(Const_b) \circ \rho)(\frac{1}{2}),
\end{aligned}$$

i.e., $Delay_a$ is not speed independent by definition 3.0.8.

2. Clearly $Delay_a(Const_b)$ is not constant at $t = 1$ where $Const_b$ is the signal constant everywhere. Therefore, $Delay_a$ is not stable by definition 3.0.5.

3. If x is a non-Zeno signal, it is also clear that $Delay_a(x)$ is a non-Zeno signal. Thus, $Delay_a$ maps non-Zeno signals into non-Zeno signals.

4. Given a signal x and an instant $t \in T^+$, by the definition of $Delay_a$, we conclude that its value at t depends only on instants before t. Therefore, $Delay_a$ is strong retrospective by definition 2.1.11.

5. By 4 and by definition 2.1.11, $Delay_a$ is retrospective.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$ such that $t < 1$, the residual of $Delay_a$ with respect to x and t is one of the following operators:

- if $t = 0$,

$$\lambda z \lambda \tau.\ Delay_a(z)(\tau);$$

- if $t > 0$,

$$\lambda z \lambda \tau. \begin{cases} a & \text{if } \tau < 1 - t \\ x\rfloor^t ; z(\tau - 1 + t) & \text{otherwise.} \end{cases}$$

As the number of such operators is uncountable (for each $t \in [0, 1[$, there exists a different operator), it follows that $Delay_a$ has not a finite number of residuals, i.e., $Delay_a$ has not finite memory by definition 2.1.17.

7. By 6, $Delay_a$ also has not countable memory.

Example 4.0.12 (Operator $PTimer$ [PRT01]) The unary operator $PTimer$ on signals is defined as:

$$PTimer(x)(t) = \begin{cases} False & \text{if } t < 1 \\ changes\ value & \text{if } x \text{ has changed value at } t - k \\ & \text{and } x \text{ was constant in }]t - k, t[\\ & \text{for some } k \in \mathbb{N} \text{ s.t. } k \leq t. \end{cases}$$

1. Let $a, b \in \Sigma$ such that $a \neq b$ and let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection such that

$$\rho(\tfrac{1}{2}) = 1.$$

Therefore,

$$\begin{array}{rcl} PTimer(Jump_{b\to c} \circ \rho)(\frac{1}{2}) & = & False \\ & \neq & True \\ & = & PTimer(Jump_{b\to c})(1) \\ & = & (PTimer(Jump_{b\to c}) \circ \rho)(\frac{1}{2}), \end{array}$$

i.e., $PTimer$ is not speed independent by definition 3.0.8.

2. Clearly $PTimer(Jump_{b\to c})$ is not constant at $t = 1$ where $Jump_{b\to c}$ is a signal constant for $t > 0$. Therefore, $PTimer$ is not stable by definition 3.0.5.

3. By its definition, it is clear that $PTimer$ maps non-Zeno signals into non-Zeno signals.

4. Given a signal x and an instant $t \in T^+$, by the definition of $PTimer$, we conclude that its value at t depends only on instants before t. Therefore, $PTimer$ is strong retrospective by definition 2.1.11.

5. By 4 and by definition 2.1.11, $PTimer$ is retrospective.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$ such that $t < 1$, the residual of $PTimer$ with respect to x and t is one of the following operators:

- if $t = 0$,

$$\lambda z \lambda \tau.\ PTimer(z)(\tau);$$

- if $t > 0$,

$$\lambda z \lambda \tau. \begin{cases} a & \text{if } \tau < 1 - t \\ \textit{changes value} & \text{if } x\rfloor^t; z \text{ has changed value at } \tau - k \\ & \text{and } x\rfloor^t; z \text{ was constant in }]\tau - k, \tau[\\ & \text{for some } k \in \mathbb{N} \text{ s.t. } k \leq \tau. \end{cases}$$

As the number of such operators is uncountable (for each $t \in [0, 1[$, there exists a different operator), it follows that $PTimer$ has not a finite number of residuals, i.e., $PTimer$ has not finite memory by definition 2.1.17.

7. By 6, $PTimer$ also has not countable memory.

Example 4.0.13 (Operator $Filter_a$ [PRT01]) The unary operator $Filter_a$ on signals is defined as follows:

$$Filter_a(x)(t) = \begin{cases} a & \text{if } t < 1 \\ x(t-1) & \text{if } t \geq 1 \text{ and } x \text{ is constant in } [t-1, t[\\ \textit{does not change} & \text{otherwise.} \end{cases}$$

1. Let $a, b \in \Sigma$ such that $a \neq b$ and let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection such that

$$\rho(\tfrac{1}{2}) = 1.$$

Therefore,

$$\begin{aligned}
Filter_a(Const_b \circ \rho)(\frac{1}{2}) &= a \\
&\neq b \\
&= Filter_a(Const_b)(1) \\
&= (Filter_a(Const_b) \circ \rho)(\frac{1}{2}),
\end{aligned}$$

i.e., $Filter_a$ is not speed independent by definition 3.0.8.

2. Clearly $Filter_a(Const_b)$ is not constant at $t = 1$ where $Const_b$ is the signal constant everywhere. Therefore, $Filter_a$ is not stable by definition 3.0.5.

3. By its definition, it is clear that $Filter_a$ maps non-Zeno signals into non-Zeno signals.

4. Given a signal x and an instant $t \in T^+$, by the definition of $Filter_a$, we conclude that its value at t depends only on instants before t. Therefore, $Filter_a$ is strong retrospective by definition 2.1.11.

5. By 4 and by definition 2.1.11, $Filter_a$ is retrospective.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$ such that $t < 1$, the residual of $Filter_a$ with respect to x and t is one of the following operators:

- if $t = 0$,

$$\lambda z \lambda \tau.\ Filter_a(z)(\tau);$$

- if $t > 0$,

$$\lambda z \lambda \tau. \begin{cases} a & \text{if } \tau < 1 - t \\ x\rfloor^t; z(\tau - 1 + t) & \text{if } \tau \geq 1 - t \text{ and} \\ & x\rfloor^t; z \text{ is constant in } [\tau - 1, \tau[\\ \textit{does not change} & \text{otherwise.} \end{cases}$$

As the number of such operators is uncountable (for each $t \in [0, 1[$, there exists a different operator), it follows that $Filter_a$ has not a finite number of residuals, i.e., $Filter_a$ has not finite memory by definition 2.1.17.

7. By 6, $Filter_a$ also has not countable memory.

Example 4.0.14 (Operator $Last_a$ [Rab97]) The unary operator $Last_a$ on signals is defined as:

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

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection,

$$\begin{array}{rcl} Last_a(x \circ \rho)(t) & = & \left\{ \begin{array}{ll} b & \text{if } \exists\ \tau_1, \tau_2,\ \tau_1 < \tau_2 < t \text{ and} \\ & \forall\ \tau \in]\tau_1, \tau_2[,\ (x \circ \rho)(\tau) = b \\ & \text{and } x \circ \rho \text{ changes at every point in }]\tau_2, t[\\ a & \text{otherwise} \end{array} \right. \\ & = & \left\{ \begin{array}{ll} b & \text{if } \exists\ \tau_1', \tau_2',\ \tau_1' < \tau_2' < \rho(t) \text{ and} \\ & \forall\ \tau' \in]\tau_1', \tau_2'[,\ x(\rho(\tau')) = b \\ & \text{and } x \text{ changes at every point in }]\rho(\tau_2'), \rho(t)[\\ a & \text{otherwise} \end{array} \right. \\ & = & Last_a(\rho(t)) \\ & = & (Last_a(x) \circ \rho)(t). \end{array}$$

Therefore, $Last_a$ is speed independent by definition 3.0.8.

2. By 1 and by proposition 3.0.9, we know that $Last_a$ is stable.

3. By 2 and by proposition 3.0.6, we know that $Last_a$ maps non-Zeno signals into non-Zeno signals.

4. Given a signal x and an instant $t \in T^+$, by the definition of $Last_a$, we conclude that its value at t depends only on instants before t. Therefore, $Last_a$ is strong retrospective by definition 2.1.11.

5. By 4 and by definition 2.1.11, $Last_a$ is retrospective.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$, the residual of $Last_a$ with respect to x and t is one of the following operators:

- if $t = 0$,

$$\lambda z \lambda \tau.\ Last_a(z)(\tau);$$

- if $t > 0$ and $x(\tau) = b$ for $\tau \in]t', t[$ with $t' < t$,

$$\lambda z \lambda \tau. \left\{ \begin{array}{ll} b & \text{if } \exists\ \tau_2,\ 0 \leq \tau_2 < \tau \\ & \text{ and } \forall\ \tau' \in\ [0, \tau_2[,\ z(\tau') = b \\ & \text{ and } z \text{ changes at every point in }]\tau_2, \tau[\\ Last_a(z)(\tau) & \text{otherwise;} \end{array} \right.$$

- if $t > 0$ and $x(\tau) = b$ for $\tau \in]t', t''[$ and x changes at every point in $]t'', t[$ with $t' < t'' < t$,

$$\lambda z \lambda \tau. \left\{ \begin{array}{ll} b & \text{if } z \text{ changes at every point in } [0, \tau[\\ Last_a(z)(\tau) & \text{otherwise.} \end{array} \right.$$

Therefore, $Last_a$ has a finite number of residuals, i.e., $Last_a$ has finite memory by definition 2.1.17.

7. By 6, $Last_a$ has countable memory.

Example 4.0.15 (Operator F [Rab97]) The unary operator F on signals is defined as follows:

$$F(x)(t) = \left\{ \begin{array}{ll} True & \text{if } x \text{ changes a finite number of times} \\ & \text{in } [0, t[\text{ or } t \text{ is rational.} \\ False & \text{otherwise.} \end{array} \right.$$

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection such that

$$\rho(1) = \sqrt{2},$$

therefore:

$$\begin{aligned} F(Rational)(1) &= True \\ &\neq False \\ &= F(Rational)(\sqrt{2}) \\ &= F(Rational)(\rho(1)), \end{aligned}$$

i.e.,

$$F(Rational) \circ \rho \neq F(Rational \circ \rho).$$

Thus, F is not speed independent by definition 3.0.8.

2. Let x be a signal such that $x = Rational\rfloor^1; Const_{False}$. x is constant at $t = 2$, however $F(x)(2) = True$ and therefore F is not constant at 2 because, if t is irrational, $F(x)(t) = False$. Clearly, F is not stable by definition 3.0.5.

3. Let x be a non-Zeno signal, it is clear that x changes a finite number of times in $[0, t[$ for any t. Therefore, $F(x) = Const_{True}$ which is a non-Zeno signal, i.e, F maps non-Zeno signals into non-Zeno signals.

4. Given a signal x and an instant $t \in T^+$, by the definition of F, we conclude that its value at t depends only on instants before t. Therefore, F is strong retrospective by definition 2.1.11.

5. By 4 and by definition 2.1.11, F is retrospective.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$ such that x does not change a finite number of times in $[0, t[$, F will have has residual with respect to x and t the following operator:

- $\lambda z \lambda \tau. \begin{cases} True & \text{if } \tau - t \text{ is rational} \\ False & \text{otherwise.} \end{cases}$

Therefore, since the set of irrationals is uncountable and for t, t' irrationals $Res(F, x, t) \neq Res(F, x, t')$, F has not a finite number of residuals, i.e., F has not finite memory by definition 2.1.17.

7. By 6, F has not also countable memory.

Example 4.0.16 (Operator G [Rab97]) The unary operator G on signals is defined as:

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

1. Let $\rho : T^+ \longrightarrow T^+$ be an order preserving bijection such that

$$\rho(1) = \sqrt{2},$$

and let x be a signal such that $x(\sqrt{2}) = 1$ and $x(t) = 0$ for $t \neq \sqrt{2}$, therefore:

$$\begin{aligned} G(x \circ \rho)(1) &= False \\ &\neq True \\ &= (G(x) \circ \rho)(1), \end{aligned}$$

i.e.,

$$G(x) \circ \rho \neq G(x \circ \rho).$$

Thus, G is not speed independent by definition 3.0.8.

2. Let x be a signal and $t \in T^+$ such that x is constant at t, i.e., exists $t_1, t_2 \in T^+$ such that x is constant in $[t_1, t_2]$ and $t_1 < t < t_2$. If exists $t_0 < t$ such that t_0 is irrational, x is constant in $[0, t_0[$ and $x(t_0) \neq x(0)$, then $G(x)(\tau) = True$, for every $\tau \in [t_1, t_2]$. Otherwise, $G(x)(\tau) = False$, for every $\tau \in [t_1, t_2]$. Therefore, $G(x)$ is constant at t, i.e., G is stable by definition 3.0.5.

3. By 2 and by proposition 3.0.6, we know that G maps non-Zeno signals into non-Zeno signals.

4. Given a signal x and an instant $t \in T^+$, by the definition of G, we conclude that its value at t may depend from t. Therefore, G is not strong retrospective by definition 2.1.11.

5. Given a signal x and an instant $t \in T^+$, by the definition of G, we conclude that its value at t may depend only on t and on instants before t. Therefore, G is retrospective by definition 2.1.11.

6. Accordingly with definition 2.1.16, if x is a signal and $t \in T^+$, G will have as residual with respect to x and t one of the following operators:

- if $t = 0$,

$$\lambda z \lambda \tau.\ G(z)(\tau);$$

- if there exists an irrational $0 < t_0 < t$ such that x is constant in $[0, t_0[$ and $x(t_0) \neq x(0)$,

$$\lambda z \lambda \tau.\ True;$$

- if x is not constant at $0 \leq t_0 < t$ with t_0 rational,

$$\lambda z \lambda \tau.\ False;$$

- if $x(\tau) = a$ for $\tau \in [0, t[$ and $0 < t$ is irrational,

$$\lambda z \lambda \tau. \begin{cases} True & \text{if } z(0) \neq a \\ G(z)(\tau) & \text{otherwise;} \end{cases}$$

- if $x(\tau) = a$ for $\tau \in [0, t[$ and $0 < t$ is rational,

$$\lambda z \lambda \tau. \begin{cases} True & \text{if there is an irrational } t_0 \leq \tau \text{ such that} \\ & z(t') = a \text{ for } t' \in [0, t_0[\text{ and } z(t_0) \neq a \\ False & \text{otherwise.} \end{cases}$$

Therefore, as the set Σ is finite, G has a finite number of residuals, i.e., G has finite memory by definition 2.1.17.

7. By 6, G has also countable memory.

Chapter 5

Properties of Operators

5.1 Closure Properties

In this section we will prove some propositions found in [Rab97] on some closure properties of operators on signals. Namely, we provide the complete proofs of propositions 5.1.1, 5.1.2, 5.2.4 and 5.2.8, which were partially given or not provided in the preceding paper [Rab97]. We also emphasize some other results which have been proved in [Rab97] and which ones will be important for our study.

Proposition 5.1.1 ([Rab97]) *The following sets of operators on signals are closed under taking residual:*

1. *retrospective operators;*
2. *strong retrospective operators;*
3. *stable operators;*
4. *speed independent operators;*
5. *finite memory retrospective operators.*

Proof: Assume that G is the residual of an operator F, i.e., G is the residual of F with respect to some $\vec{x} \in Sig(\Sigma)^n$ and some $t \in T^+$. From definition 2.1.16 we know that:

$$G(\vec{z})(t') = F((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n))(t + t'),$$

for $\vec{z} \in Sig(\Sigma)^n$.
(1) Suppose that F is retrospective and $\vec{z}, \vec{w} \in Sig(\Sigma)^n$ such that

$$z_i(t') = w_i(t'),$$

for $t' \in [0, \tau]$ and $i = 1, ..., n$. Then

$$(x_i\rfloor^t; z_i)(\tau) = (x_i\rfloor^t; w_i)(t + t')$$

with $t' \in [-t, \tau]$ and $i = 1, ..., n$. Since F is a retrospective operator, we have

$$F((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n))(t + t') = F((x_1\rfloor^t; w_1), ..., (x_n\rfloor^t; w_n))(t + t'),$$

for $t' \in [-t, \tau]$ and $i = 1, ..., n$. Hence for $t' \in [0, \tau]$, we have

$$G(\vec{z})(t') = G(\vec{w})(t'),$$

i.e, by definition 2.1.11, G is retrospective.
(2) Similar to (1).
(3) Suppose that F is a stable operator and take $\vec{z} \in Sig(\Sigma)^n$ such that $\vec{z}$ is constant at instant τ. Then $(x_i\rfloor^t; z_i)$ is constant at instant $t + \tau$. Since F is stable, we have that $F((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n))$ is constant at instant $t + \tau$. Hence $G(\vec{z})$ is constant at instant τ, i.e, by definition 3.0.5, G is stable.
(4) Suppose that F is a speed independent operator and that ρ is an order preserving bijection. Defining ρ' as

$$\rho'(\tau) = \begin{cases} \tau & \text{if } \tau < t \\ t + \rho(\tau - t) & \text{if } \tau \geq t \end{cases}$$

we have that ρ' is an order preserving bijection too. From the definition of ρ' we observe that

$$(x_i\rfloor^t; z_i) \circ \rho' = (x_i\rfloor^t; (z_i \circ \rho)),$$

for $i = 1, ..., n$. Therefore, since F is speed independent, we have that

$$\begin{aligned} G(z_1 \circ \rho, ..., z_n \circ \rho)(t') &= F((x_1\rfloor^t; (z_1 \circ \rho)), ..., (x_n\rfloor^t; (z_n \circ \rho))(t + t') \\ &= F((x_1\rfloor^t; z_1) \circ \rho', ..., (x_n\rfloor^t; z_n) \circ \rho')(t + t') \\ &= F((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n))(\rho'(t + t')) \\ &= F((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n))(t + \rho(t')) \\ &= G(z_1, ..., z_n)(\rho(t')). \end{aligned}$$

Hence $G(\vec{z} \circ \rho) = G(\vec{z}) \circ \rho$, i.e., G is speed independent.
(5) Suppose that F is a finite memory retrospective operator and G is a residual of F as before. Let H be the residual of G with respect to $\vec{y}$ and τ, then

$$\begin{aligned} H(\vec{z})(t') &= G((y_1\rfloor^\tau; z_1), ..., (y_n\rfloor^\tau; z_n))(\tau + t') \\ &= F((x_1\rfloor^t; (y_1\rfloor^\tau; z_1)), ..., (x_n\rfloor^t; (y_n\rfloor^\tau; z_n)))(t + \tau + t') \\ &= F(((x_1\rfloor^t; y_1)\rfloor^{t+\tau}; z_1), ..., ((x_n\rfloor^t; y_n)\rfloor^{t+\tau}; z_n))((t + \tau) + t'). \end{aligned}$$

Thus, H is a residual of F with respect to $((x_1\rfloor^t; y_1), ..., (x_n\rfloor^t; y_n))$ and $t+\tau$. We know that F has a finite number of distinct residuals and that G is a residual of F. Hence, G has a finite number of distinct residuals and so is a finite memory retrospective[1] operator. $\square$

Proposition 5.1.2 ([Rab97]) *The following sets of operators from signals to signals are closed under composition:*

1. *retrospective operators;*

2. *strong retrospective operators;*

3. *stable operators;*

[1]Retrospective by 1.

4. *speed independent operators;*

5. *finite memory retrospective operators.*

Proof:
(1) Let $F_i : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$, for $i = 1, ..., m$, and $G : Sig(\Sigma)^m \longrightarrow Sig(\Sigma)$ be retrospective operators. Suppose $\vec{x}, \vec{x}' \in Sig(\Sigma)^n$ such that $x_j(\tau) = x'_j(\tau)$, for $j = 1, ..., n$ and $\tau \in [0, t]$. If F_i is retrospective, for every i, then $F_i(\vec{x})(\tau) = F_i(\vec{x}')(\tau)$, for every i and $\tau \in [0, t]$. If G is retrospective, then by the above statements and definition 2.1.11 we know that

$$G(F_1(\vec{x}), ..., F_n(\vec{x}))(\tau) \quad = \quad G(F_1(\vec{x}'), ..., F_n(\vec{x}'))(\tau)),$$

for $\tau \in [0, t]$, i.e., the composition of G with $F_1, ..., F_m$ is a retrospective operator.
(2) Similar to (1).
(3) Let $F_i : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$, for $i = 1, ..., m$, and $G : Sig(\Sigma)^m \longrightarrow Sig(\Sigma)$ be stable operators. Suppose $\vec{x} \in Sig(\Sigma)^n$ such that x_j is constant at t with $j = 1, ..., n$. If F_i is stable, for every i, then $F_i(\vec{x})$ is constant at t for every i (by definition 3.0.5). If G is stable, then by the above statements we know that

$$G(F_1(\vec{x}), ..., F_n(\vec{x}))$$

is constant at t, i.e., by definition 3.0.5, $G \circ (F_1, ..., F_m)$ is a stable operator.
(4) Let $F_i : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$, for $i = 1, ..., m$, and $G : Sig(\Sigma)^m \longrightarrow Sig(\Sigma)$ be speed independent operators. Suppose that ρ is an order preserving bijection. If F_i is speed independent, for every i, then

$$F_i(\vec{x} \circ \rho) \quad = \quad F_i(\vec{x}) \circ \rho,$$

for every i and $\vec{x} \in Sig(\Sigma)^n$. If G is speed independent, then

$$\begin{aligned} (G(F_1(\vec{x}), ..., F_m(\vec{x}))) \circ \rho &= G(F_1(\vec{x}) \circ \rho, ..., F_m(\vec{x}) \circ \rho) \\ &= G(F_1(\vec{x} \circ \rho), ..., F_m(\vec{x} \circ \rho)), \end{aligned}$$

for every $\vec{x} \in Sig(\Sigma)^n$, i.e., the composition of G with $F_1, ..., F_m$ is a speed independent operator.
(5) Let $F_i : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$, for $i = 1, ..., m$, and $G : Sig(\Sigma)^m \longrightarrow Sig(\Sigma)$ be finite memory retrospective operators. Suppose that F'_i is the residual of F_i with respect to $\vec{x}$ and t, for every i, and that G' is the residual of G with respect to $(F_1(\vec{x}), ..., F_m(\vec{x}))$ and t. Then by definition 2.1.16 the residual R of the composition of G with $F_1, ..., F_m$ with respect to $\vec{x}$ and t is

$$\begin{aligned} & R(\vec{z})(t') = \\ = \ & G(F_1((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n)), ..., F_m((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n)))(t + t') \\ = \ & G((F_1(\vec{x})\rfloor^t; F'_1(\vec{z})), ..., (F_m(\vec{x})\rfloor^t; F'_m(\vec{z})))(t + t') \\ = \ & G'(F'_1(\vec{z}), ..., F'_m(\vec{z}))(t') \end{aligned}$$

with $\vec{z} \in Sig(\Sigma)^n$. Thus, R is equal to the composition of G' with $F'_1, ..., F'_m$ and as G and F_i, for $i = 1, .., m$, have finitely many residuals by definition 2.1.17, the composition of G with $F_1, ..., F_m$ have finitely many residuals too. In fact, if G has k residuals and each F_i has n_i residuals, the composition of G with

$F_1, ..., F_m$ will have $k \times n_1 \times ... \times n_m$ or less residuals because this is the number of possible compositions of G' with $F'_1, ..., F'_m$. Therefore, by definition 2.1.17 and by (1), the composition of G with $F_1, ..., F_m$ is a finite memory retrospective operator. □

Analyzing the above proofs we can check that they hold for the set of operators over any set C of signals which is closed under concatenation, suffix and order preserving bijections. In particular the next proposition holds:

Proposition 5.1.3 ([Rab97]) *Let C be a set of signals which is closed under concatenation, suffix and the order preserving bijections. The following sets of operators over C signals are closed under composition and under residual:*

1. *retrospective operators;*
2. *strong retrospective operators;*
3. *stable operators;*
4. *speed independent operators;*
5. *finite memory retrospective operators.*

Proof: Similar to the previous proofs of this section. □

5.2 Properties of Finite Memory Retrospective Operators

To study the properties of finite memory retrospective operators we will deal with general signals and we will follow the main ideas in [Rab97].

5.2.1 Finite Memory Signals

The following proposition is proved[2] in [Rab97] and is a key technical proposition needed in our study.

Proposition 5.2.1 ([Rab97]) *A general signal x is finite memory if and only if x is constant on the positive reals.*

Such signals are of the form:

Definition 5.2.2 ($Jump_{a\to b}$) For some $a, b \in \Sigma$ we define $Jump_{a\to b} \in Sig(\Sigma)$ as follows:

$$Jump_{a\to b}(\tau) = \begin{cases} a & \text{if } \tau = 0 \\ b & \text{otherwise.} \end{cases}$$

As was stated in proposition 5.2.1, the finite memory signals are very simple. We have seen before that a signal is a 0-ary operator on signals, may we conclude the same about the simplicity of general operators on signals? Looking at the examples given in chapter 4 we say no. In fact is enough to consider the pointwise

[2] This is a long proof.

extension of a function $g : \Sigma^n \longrightarrow \Sigma$, which has finite memory as we have seen in example 4.0.8.

We will state now some consequences of proposition 5.2.1. The following two propositions were proved in [Rab97], however we reproduce their proofs in order to emphasize the importance of proposition 5.2.1 and because some minor corrections are needed.

Proposition 5.2.3 ([Rab97]) *If $F : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$ is a finite memory retrospective operator, then*

$$F(Jump_{a_1 \to b_1}, ..., Jump_{a_n \to b_n}) \quad = \quad Jump_{c \to d},$$

for some c and d in Σ.

Proof: F is a finite memory retrospective operator. For every i, $Jump_{a_i \to b_i}$ is a finite memory signal by proposition 5.2.1. Since signals are 0-ary operators on signals, $Jump_{a_i \to b_i}$ is a finite memory 0-ary operator. Therefore, $F(Jump_{a_1 \to b_1}, ..., Jump_{a_n \to b_n})$ is a finite memory 0-ary operator by proposition 5.1.2, i.e., $F(Jump_{a_1 \to b_1}, ..., Jump_{a_n \to b_n})$ is a finite memory signal and we conclude by proposition 5.2.1 that

$$F(Jump_{a_1 \to b_1}, ..., Jump_{a_n \to b_n}) \quad = \quad Jump_{c \to d},$$

for some c and d in Σ. $\Box$

Proposition 5.2.4 ([Rab97]) *Every finite memory retrospective operator is stable.*

Proof: Assume that $F : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$ is a finite memory retrospective operator. We pretend to show that if $\vec{x} \in Sig(\Sigma)^n$ is constant at $t > 0$ then $F(\vec{x})$ is constant at t. If $\vec{x}$ is constant at $t > 0$, then there exists $\epsilon > 0$ such that

$$x_i(\tau) = x_i(t) = b_i, \text{ for } \tau \in [t - \epsilon, t + \epsilon].$$

Let $G = Res(F, \vec{x}, t - \epsilon)$, from the previous and because F is retrospective it follows that

$$G(Const_{b_1}, ..., Const_{b_n})(\tau) = F(\vec{x})(t - \epsilon + \tau),$$

where $\tau \in [0, 2\epsilon]$ and $Const_{b_i}$ is the signal which is constant (and equal to b_i) everywhere, for every i. So, $F(\vec{x})$ is constant at t iff $G(Const_{b_1}, ..., Const_{b_n})$ is constant at ϵ. By proposition 5.1.1, since G is a residual of F, G is a finite memory retrospective operator. From proposition 5.2.1 we may infer that $G(Const_{b_1}, ..., Const_{b_n})$ is constant on the positive reals and therefore it is constant at ϵ, i.e., $F(\vec{x})$ is constant at t. $\Box$

From this last proposition and from proposition 3.0.6, we get the following corollary.

Corollary 5.2.5 *A finite memory retrospective operator maps non-Zeno signals into non-Zeno signals.*

Let F be an operator over general signals, we denote the restriction of F to non-Zeno signals by $Rest(F)$ which is defined as

$$Rest(F) = \lambda \vec{x} \in nZSig(\Sigma)^n.\ F\vec{x}.$$

As result of corollary 5.2.5, the following proposition holds and is proved in [Rab97].

Proposition 5.2.6 ([Rab97]) *If F is a finite memory retrospective operator on signals then $Rest(F)$ is an operator from non-Zeno signals to non-Zeno signals. Moreover, $Rest(F)$ is a finite memory retrospective operator over the non-Zeno signals.*

5.2.2 State Function

We pretend in this subsection generalize to n-ary case the definitions and results presented in [Rab97] about the state function.

Definition 5.2.7 (State Operator [Rab97]) Let G_0 be a finite memory retrospective operator from $Sig(\Sigma)^n$ to $Sig(\Sigma')$ and let $\vec{G} =< G_0, G_1, ..., G_k >$ be a sequence of all its residuals. It is clear that any residual of G_i is a residual of G_0. Define functions

$$out_{\vec{G}} : \Sigma^n \times \{0, .., k\} \longrightarrow \Sigma'$$
$$state_{\vec{G}} : Sig(\Sigma)^n \longrightarrow Sig(\{0, ..., k\} \to \{0, ..., k\})$$

as follows:

$$\begin{aligned} out_{\vec{G}}(a_1, ..., a_n, i) &= G_i(Const_{a_1}, ..., Const_{a_n})(0), \\ state_{\vec{G}}(\vec{x})(t)i &= j, \text{ if } G_j = Res(G_i, \vec{x}, t) \end{aligned}$$

The following result holds and, however stated, its proof is not provided in [Rab97].

Proposition 5.2.8 (Properties of the State Operator [Rab97]) .

1. *$state_{\vec{G}}(\vec{x})(0) = id$, the identity permutation.*
2. *$state_{\vec{G}}$ is a strong retrospective operator.*
3. *$state_{\vec{G}}((x_1\rfloor^{t_1}; z_1), ..., (x_n\rfloor^{t_1}; z_n))(t_1 + t_2) =$ $= (state_{\vec{G}}(\vec{z})(t_2)) \circ (state_{\vec{G}}(\vec{x})(t_1))$*
4. *$G(\vec{x})(t) = out_{\vec{G}}(\vec{x}(t), state_{\vec{G}}(\vec{x})(t)0)$*

Proof:
1. By definition 2.1.16, for every G_i and $\vec{x} \in Sig(\Sigma)^n$, we have that:

$$Res(G_i, \vec{x}, 0) = \lambda z_1...z_n.\lambda t'.G_i((x_1\rfloor^0; z_1), ..., (x_n\rfloor^0; z_n))(0 + t') = G_i.$$

Therefore $state_{\vec{G}}(\vec{x})(0)i = i$, for every i, i.e., $state_{\vec{G}}(\vec{x})(0)$ is the identity permutation.
2. Again, by definition 2.1.16, we know that if $\vec{x}, \vec{y} \in Sig(\Sigma)^n$ coincide in the interval $[0, t[$, then $Res(G_i, \vec{x}, \tau) = Res(G_i, \vec{y}, \tau)$, for every $\tau \in [0, t]$ and i, i.e., $state_{\vec{G}}(\vec{x})(\tau) = state_{\vec{G}}(\vec{y})(\tau)$ for every $\tau \in [0, t]$. Therefore, by definition 2.1.11, $state_{\vec{G}}$ is a strong retrospective operator.

3. Assume that $state_{\vec{G}}((x_1\rfloor^{t_1}; z_1), ..., (x_n\rfloor^{t_1}; z_n))(t_1+t_2)i = j$ for some i and j, i.e., $G_j = Res(G_i, ((x_1\rfloor^{t_1}; z_1), ..., (x_n\rfloor^{t_1}; z_n)), t_1+t_2)$. By definition 2.1.16, we have the following:

$$\begin{aligned} G_j &= Res(G_i, ((x_1\rfloor^{t_1}; z_1), ..., (x_n\rfloor^{t_1}; z_n)), t_1+t_2) \\ &= \lambda\vec{w}\lambda t'.\ G_i(((x_1\rfloor^{t_1}; z_1)\rfloor^{t_1+t_2}; w_1), ..., ((x_n\rfloor^{t_1}; z_n)\rfloor^{t_1+t_2}; w_n))(t_1+t_2+t') \\ &= \lambda\vec{w}\lambda t'.\ G_i((x_1\rfloor^{t_1}; (z_1\rfloor^{t_2}; w_1)), ..., (x_n\rfloor^{t_1}; (z_n\rfloor^{t_2}; w_n)))(t_1+t_2+t') \\ &= \lambda\vec{w}\lambda t'.\ G_k((z_1\rfloor^{t_2}; w_1), ..., (z_n\rfloor^{t_2}; w_n))(t_1+t_2+t'), \end{aligned}$$

where $G_k = Res(G_i, \vec{x}, t_1)$ and more, $G_j = Res(G_k, \vec{z}, t_2)$. Therefore there exists k such that

$$\begin{aligned} state_{\vec{G}}(\vec{x})(t_1)i &= k \\ state_{\vec{G}}(\vec{z})(t_2)k &= j, \end{aligned}$$

i.e.,

$$state_{\vec{G}}((x_1\rfloor^{t_1}; z_1), ..., (x_n\rfloor^{t_1}; z_n))(t_1+t_2) = (state_{\vec{G}}(\vec{z})(t_2)) \circ (state_{\vec{G}}(\vec{x})(t_1)).$$

4. By definition 5.2.7 we verify that

$$out_{\vec{G}}(\vec{x}(t), state_{\vec{G}}(\vec{x})(t)0) = out_{\vec{G}}(\vec{x}(t), i),$$

where i is such that $G_i = Res(G_0, \vec{x}, t)$. Therefore

$$\begin{aligned} out_{\vec{G}}(\vec{x}(t), i) &= G_i(Const_{x_1(t)}, ..., Const_{x_n(t)}) \\ &= G_0((x_1\rfloor^t; Const_{x_1(t)}), ..., (x_n\rfloor^t; Const_{x_n(t)})). \end{aligned}$$

Since G_0 is retrospective, we have

$$out_{\vec{G}}(\vec{x}(t), state_{\vec{G}}(\vec{x})(t)0) = G_0(\vec{x})(t). \quad \Box$$

The above proposition implies the following one which proof in [Rab97] can be generalized straightforward with respect to n-ary operators on signals.

Proposition 5.2.9 ([Rab97]) *$state_{\vec{G}}$ is a finite memory strong retrospective operator on signals. Moreover, there exists*

$$\delta_{\vec{G}} : \Sigma^{2n} \longrightarrow (\{0, ..., k\} \to \{0, ..., k\})$$

such that:

1. $\delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n) = state_{\vec{G}}(Jump_{a_1\to b_1}, ..., Jump_{a_n\to b_n})t$, *for every* $t > 0$*;*

2. $\delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n) = \delta_{\vec{G}}(b_1, ..., b_n, b_1, ..., b_n) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n)$, *for any* $a_1, ..., a_n, b_1, ..., b_n \in \Sigma$.

We will now study the same concepts over non-Zeno signals. Let G_0 be a finite memory retrospective operator from non-Zeno signals over Σ to non-Zeno signals over Σ'. As we did before, let $\vec{G} = < G_0, ..., G_k >$ be the sequence of all its residuals. Thus, defining $state_{\vec{G}}$ like in definition 5.2.7, we state the following theorem which proof may be found in [Rab97] and generalized straightforward too.

Theorem 5.2.10 ([Rab97]) *The operator* $state_{\vec{G}}$ *maps non-Zeno signals to non-Zeno signals. Moreover, propositions 5.2.8 and 5.2.9 hold whenever all notions are relativized to non-Zeno signals. In particular, there exist*

$$\delta_{\vec{G}} : \Sigma^{2n} \longrightarrow (\{0, ..., k\} \to \{0, ..., k\})$$

and

$$out_{\vec{G}} : \Sigma^n \times \{0, .., k\} \longrightarrow \Sigma'$$

such that:

1. $\delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n) = \delta_{\vec{G}}(b_1, ..., b_n, b_1, ..., b_n) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n)$;
2. $G(\vec{x})(t) = out_{\vec{G}}(\vec{x}(t), state_{\vec{G}}(\vec{x})(t)0)$;
3. $state_{\vec{G}}$ *is a strong retrospective operator from non-Zeno signals over* Σ *to non-Zeno signals over* $(\{0, ..., k\} \to \{0, ..., k\})$.
4. $state_{\vec{G}}(\vec{x})(0) = id$*, the identity permutation.;*
5. $state_{\vec{G}}(Jump_{a_1 \to b_1}, ..., Jump_{a_n \to b_n})t = \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n)$;
6. $state_{\vec{G}}((Jump_{a_1 \to b_1}\rfloor^{t'}; x_1), ..., (Jump_{a_n \to b_n}\rfloor^{t'}; x_1))(t + t') =$
   $= (state_{\vec{G}}(\vec{x})t) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n)$.

Motivated by the above theorem we get the following definition.

Definition 5.2.11 (Definability [Rab97]) Let Σ and Q be finite sets and let $\delta : \Sigma^{2n} \longrightarrow (Q \to Q)$. An operator 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 strong retrospective operator;
2. $F(\vec{x})(0) = id_Q$;
3. $F(Jump_{a_1 \to b_1}, ..., Jump_{a_n \to b_n})(t) = \delta(a_1, ..., a_n, b_1, ..., b_n)$, for every $t > 0$ and $a_1, ..., a_n, b_1, ..., b_n \in \Sigma$;
4. $F((Jump_{a_1 \to b_1}\rfloor^{t'}; x_1), ..., (Jump_{a_n \to b_n}\rfloor^{t'}; x_n))(t + t') =$
   $= (F(\vec{x})t) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n)$.

We conclude this chapter with two propositions which will be useful on next chapter. Both them have been proved in [Rab97] and their proofs can be straightforward generalized.

Proposition 5.2.12 ([Rab97]) *Let* δ *be a function in* $\Sigma^{2n} \longrightarrow (Q \to Q)$*. Then there exists at most one operator definable by* δ*.*

Proposition 5.2.13 ([Rab97]) *If for every* $a_1, ..., a_n, b_1, ..., b_n \in \Sigma$*:*

$$\delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n) =$$
$$= \delta_{\vec{G}}(b_1, ..., b_n, b_1, ..., b_n) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n),$$

then there exists a finite memory speed independent operator definable by δ*.*

The above propositions imply the following corollary.

Corollary 5.2.14 ([Rab97]) *If for every* $a_1, ..., a_n, b_1, ..., b_n \in \Sigma$*:*

$$\delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n) =$$
$$= \delta_{\vec{G}}(b_1, ..., b_n, b_1, ..., b_n) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n),$$

then there exists a unique operator definable by δ*. Moreover, the operator definable by* δ *is finite memory strong retrospective and speed independent.*

Results of this chapter

5.1.1 The following sets of operators on signals are closed under taking residual:

1. retrospective operators;
2. strong retrospective operators;
3. stable operators;
4. speed independent operators;
5. finite memory retrospective operators.

5.1.2 The following sets of operators from signals to signals are closed under composition:

1. retrospective operators;
2. strong retrospective operators;
3. stable operators;
4. speed independent operators;
5. finite memory retrospective operators.

5.1.3 Let C be a set of signals which is closed under concatenation, suffix and the order preserving bijections. The following sets of operators over C signals are closed under composition and under residual:

1. retrospective operators;
2. strong retrospective operators;
3. stable operators;
4. speed independent operators;
5. finite memory retrospective operators.

5.2.1 A general signal x is finite memory if and only if x is constant on the positive reals.

5.2.3 If $F : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$ is a finite memory retrospective operator then $F(Jump_{a_1 \to b_1}, ..., Jump_{a_n \to b_n}) = Jump_{c \to d}$ for some c and d in Σ.

5.2.4 Every finite memory retrospective operator is stable.

5.2.5 A finite memory retrospective operator maps non-Zeno signals to non-Zeno signals.

5.2.6 If F is a finite memory retrospective operator on signals then $Rest(F)$ is an operator from non-Zeno signals to non-Zeno signals. Moreover, $Rest(F)$ is a finite memory retrospective operator over the non-Zeno signals.

5.2.8
1. $state_{\vec{G}}(\vec{x})(0) = id$, the identity permutation.
2. $state_{\vec{G}}$ is a strong retrospective operator.
3. $state_{\vec{G}}((x_1\rfloor^{t_1}; z_1), ..., (x_n\rfloor^{t_1}; z_n))(t_1 + t_2) =$
   $= (state_{\vec{G}}(\vec{z})(t_2)) \circ (state_{\vec{G}}(\vec{x})(t_1))$
4. $G(\vec{x})(t) = out_{\vec{G}}(\vec{x}(t), state_{\vec{G}}(\vec{x})(t)0)$

5.2.9 $state_{\vec{G}}$ is a finite memory strong retrospective operator on signals. Moreover, there exists

$$\delta_{\vec{G}} : \Sigma^{2n} \longrightarrow (\{0, ..., k\} \to \{0, ..., k\})$$

such that:

1. $\delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n) = state_{\vec{G}}(Jump_{a_1 \to b_1}, ..., Jump_{a_n \to b_n})t$, for every $t > 0$;
2. $\delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n) =$ $= \delta_{\vec{G}}(b_1, ..., b_n, b_1, ..., b_n) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n)$, for any $a_1, ..., a_n, b_1, ..., b_n \in \Sigma$.

5.2.10 The operator $state_{\vec{G}}$ maps non-Zeno signals to non-Zeno signals. Moreover, propositions 5.2.8 and 5.2.9 hold whenever all notions are relativized to non-Zeno signals. In particular, there exist

$$\delta_{\vec{G}} : \Sigma^{2n} \longrightarrow (\{0, ..., k\} \to \{0, ..., k\})$$

and

$$out_{\vec{G}} : \Sigma^n \times \{0, .., k\} \longrightarrow \Sigma'$$

such that:

1. $\delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n) =$ $= \delta_{\vec{G}}(b_1, ..., b_n, b_1, ..., b_n) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n)$;
2. $G(\vec{x})(t) = out_{\vec{G}}(\vec{x}(t), state_{\vec{G}}(\vec{x})(t)0)$;
3. $state_{\vec{G}}$ is a strong retrospective operator from non-Zeno signals over Σ to non-Zeno signals over $(\{0, ..., k\} \to \{0, ..., k\})$.
4. $state_{\vec{G}}(\vec{x})(0) = id$, the identity permutation.;
5. $state_{\vec{G}}(Jump_{a_1 \to b_1}, ..., Jump_{a_n \to b_n})t = \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n)$;
6. $state_{\vec{G}}((Jump_{a_1 \to b_1} \rfloor^{t'}; x_1), ..., (Jump_{a_n \to b_n} \rfloor^{t'}; x_n))(t + t') =$ $= (state_{\vec{G}}(\vec{x})t) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n)$.

5.2.13 If for every $a_1, ..., a_n, b_1, ..., b_n \in \Sigma$:

$$\delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n) = \\ = \delta_{\vec{G}}(b_1, ..., b_n, b_1, ..., b_n) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n),$$

then there exists a finite memory speed independent operator definable by δ.

5.2.14 If for every $a_1, ..., a_n, b_1, ..., b_n \in \Sigma$:

$$\delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n) = \\ = \delta_{\vec{G}}(b_1, ..., b_n, b_1, ..., b_n) \circ \delta_{\vec{G}}(a_1, ..., a_n, b_1, ..., b_n),$$

then there exists a unique operator definable by δ. Moreover, the operator definable by δ is finite memory strong retrospective and speed independent.

Chapter 6

Continuous-time Automata

The finite state transducer will be defined in this chapter and for that the results studied in the last chapter will be very helpful. Our main objective is to show that every finite state transducer computes a finite memory retrospective operator over non-Zeno signals and that the inverse also holds.

In the first section we give also a result found in [Rab97] that implies the impossibility of have a finite description of finite memory retrospective operators over general signals. We conclude this chapter by given some examples of transducers.

6.1 Finite State Transducers over Non-Zeno Signals

As we have done in the last chapter with respect to the operators over signals, we will consider transducers with multiple input channels. We define now a finite state transducer.

Definition 6.1.1 (Finite State Transducer [Rab97]) A finite state transducer $\mathcal{A}$ over non-Zeno signals is a tuple $< Q, q_0, \Sigma_{in}, \Sigma_{out}, n, out, \delta >$ such that

- Q is a finite non empty set of states,
- $q_0 \in Q$ is the initial state,
- Σ_{in}[1] is the input alphabet,
- Σ_{out} is the output alphabet,
- n is the arity of $\mathcal{A}$,
- $out : Q \times \Sigma_{in}^n \longrightarrow \Sigma_{out}$ is the output function and
- $\delta : \Sigma_{in}^{2n} \longrightarrow (Q \to Q)$ is the transition function which verifies

$$\delta(b_1, ..., b_n, b_1, ..., b_n) \circ \delta(a_1, ..., a_n, b_1, ..., b_n) = \delta(a_1, ..., a_n, b_1, ..., b_n).$$

[1]We assume that all input signals are defined over the same alphabet, however multiple input alphabets may be considered without lost of generality.

Note that this definition translates the ideas introduced in the last chapter. Each one of the states corresponds to a residual and each transition corresponds to a jump from $a_1, ..., a_n$ to $b_1, ..., b_n$. The condition for δ comes straightforwardly from the results obtained before.

We will use a graphical representation for transducers. As we may see in figure 6.1, the states will be represented by nodes and the transitions we will be represented by labeled arcs. The output function can be represented by labeling the nodes as follows:

$$q \text{ has label } < (c_{11}, ..., c_{n1})/d_1, ..., (c_{1k}, ..., c_{nk})/d_k >$$
$$\text{if and only if}$$
$$out(q, c_{1i}, ..., c_{ni}) = d_i \text{ for every } i.$$

The initial state will be indicated with an arrow as is done in figure 6.1.

A transducer receives a signal $\vec{x} \in nZSig(\Sigma_{in})$ as input and produces as output a signal $y \in nZSig(\Sigma_{out})$. If $\vec{x}(t) = (a_1, ..., a_n)$, for $t \in T^+$, then $y(t) = b$ where b is given accordingly with the label of the current node, i.e., accordingly with the *out* function in each state. A transition between states occurs accordingly with the jump of $\vec{x}$ at t, if $\vec{x}$ produces a jump from $(a_1, ..., a_n)$ to $(b_1, ..., b_n)$ at t, then occurs a transition from the current state to other one, where the transition arc is labeled by $< (a_1, ..., a_n), (b_1, ..., b_n) >$.

The operator computable by a transducer is obtained as follows.

Definition 6.1.2 (Operator Computable by a Transducer [Rab97]) . Let $\mathcal{A} =< Q, q_0, \Sigma_{in}, \Sigma_{out}, n, out, \delta >$ be a transducer. Note that by proposition 5.2.13 there exists a unique operator F^δ definable by δ. The operator $F_\mathcal{A}$ computable by $\mathcal{A}$ is defined as $out(F^\delta \vec{x} t q_0, \vec{x} t)$.

Making use of the results achieved in the last chapter, we will give now a characterization of finite memory retrospective operators and we will state that they are speed independent. The following two theorems have been stated and partially proved in [Rab97], complete proofs will be provided by us.

Theorem 6.1.3 *An operator over non-Zeno signals is a finite memory retrospective operator if and only if it is computable by a transducer.*

Proof: Let $\mathcal{A} =< Q, q_0, \Sigma_{in}, \Sigma_{out}, n, out, \delta >$ be a transducer. As we have seen in previous chapter, namely in corollary 5.2.14 , F^δ is a finite memory retrospective operator. Therefore, the operator $F_\mathcal{A}$ is a finite memory because it is defined as the composition of the pointwise operator out[2] and F^δ. □

Theorem 6.1.4 *Every finite memory retrospective operator over non-Zeno signals is speed independent.*

Proof: By corollary 5.2.14, the operator F^δ is speed independent and then the operator computable by a transducer is also speed independent. Therefore the theorem 6.1.3 implies that every finite memory retrospective operator over non-Zeno signals is speed independent. □

In [Rab97] is provided a description of the operator computable by a transducer in terms of ω-languages and proved that the set of finite memory speed

[2]Clearly a pointwise operator is retrospective because it only depends on current instant and it is finite memory because it only have one residual, itself (chapter 4).

independent retrospective operators can not be representable by finite means. We will not prove this result, we only state a further result that implies it, proved in [Rab97].

Theorem 6.1.5 ([Rab97]) *The set of finite memory speed independent retrospective operator (over general signals) is at least uncountable.*

6.2 Examples

In this section we will provide three examples of transducers for operators that we already give as example or that we will use in next chapter. In the construction of such transducers, we will make use of the properties shown in chapter 4 for each case.

Example 6.2.1 The unary operator

$$LLim_0 : nZSig(\{0,1\}) \longrightarrow nZSig(\{0,1\})$$

is defined as follows:

$$LLim_0(x)(t) = \begin{cases} 0 & \text{if } t = 0 \\ b & \text{if } t > 0 \text{ and } b \in \{0,1\} \text{ is the left limit of } x \text{ at } t. \end{cases}$$

As we will see in chapter 7, this operator have two residuals:

$$LLim_0(z)(\tau) = \begin{cases} 0 & \text{if } \tau = 0 \\ b & \text{if } \tau > 0 \text{ and } b \in \{0,1\} \text{ is the left limit of } z \text{ at } \tau, \end{cases}$$

$$LLim_1(z)(\tau) = \begin{cases} 1 & \text{if } \tau = 0 \\ b & \text{if } \tau > 0 \text{ and } b \in \{0,1\} \text{ is the left limit of } z \text{ at } \tau. \end{cases}$$

Thus, we construct the transducer $\mathcal{A}_{LLim_0} =< Q, q_0, \Sigma_{in}, \Sigma_{out}, n, out, \delta >$.

- Since there are two residuals, we need two states, i.e, $Q = \{q_0, q_1\}$. Each state is associated to a residual as follows:

$$q_0 \dashrightarrow LLim_0 \text{ and } q_1 \dashrightarrow LLim_1.$$

- The initial state is the state associated to the operator $LLim_0$, i.e., q_0.
- $\Sigma_{in} = \{0,1\}$.
- $\Sigma_{out} = \{0,1\}$.
- Since $LLim_0$ is an unary operator, we have $n = 1$.
- We obtain the function *out* as follows:

$$\begin{array}{c} out(q_0,0) = LLim_0(Const_0)(0) = 0 \\ out(q_0,1) = LLim_0(Const_1)(0) = 0 \\ out(q_1,0) = LLim_1(Const_0)(0) = 1 \\ out(q_1,1) = LLim_1(Const_1)(0) = 1 \end{array}$$

- Since Σ_{in} has two symbols, we know that there exist four different jumps:

State	Transition	δ	*out*
q_0	$0,0$	q_0	$0 \mapsto 0$
q_0	$0,1$	q_1	$0 \mapsto 0$
q_0	$1,0$	q_0	$1 \mapsto 0$
q_0	$1,1$	q_1	$1 \mapsto 0$
q_1	$0,0$	q_0	$0 \mapsto 1$
q_1	$0,1$	q_1	$0 \mapsto 1$
q_1	$1,0$	q_0	$1 \mapsto 1$
q_1	$1,1$	q_1	$1 \mapsto 1$

Table 6.1: δ and *out* functions in $\mathcal{A}_{LLim_0}$.

$$Jump_{0\to 0},\ Jump_{0\to 1},\ Jump_{1\to 0} \text{ and } Jump_{1\to 1},$$

i.e., there exist four possible transitions at each state. Therefore, the function δ is defined as follows:

$$\begin{aligned}
&\delta(0,0)q_0 = q_0 \text{ because } Res(LLim_0, Jump_{0\to 0}, t) = LLim_0,\\
&\delta(0,0)q_1 = q_0 \text{ because } Res(LLim_1, Jump_{0\to 0}, t) = LLim_0,\\
&\delta(0,1)q_0 = q_1 \text{ because } Res(LLim_0, Jump_{0\to 1}, t) = LLim_1,\\
&\delta(0,1)q_1 = q_1 \text{ because } Res(LLim_1, Jump_{0\to 1}, t) = LLim_1,\\
&\delta(1,0)q_0 = q_0 \text{ because } Res(LLim_0, Jump_{1\to 0}, t) = LLim_0,\\
&\delta(1,0)q_1 = q_0 \text{ because } Res(LLim_1, Jump_{1\to 0}, t) = LLim_0,\\
&\delta(1,1)q_0 = q_1 \text{ because } Res(LLim_0, Jump_{1\to 1}, t) = LLim_1,\\
&\delta(1,1)q_1 = q_1 \text{ because } Res(LLim_1, Jump_{1\to 1}, t) = LLim_1,
\end{aligned}$$

for $t > 0$.

In figure 6.1 we give the transducer $\mathcal{A}_{LLim_0}$.

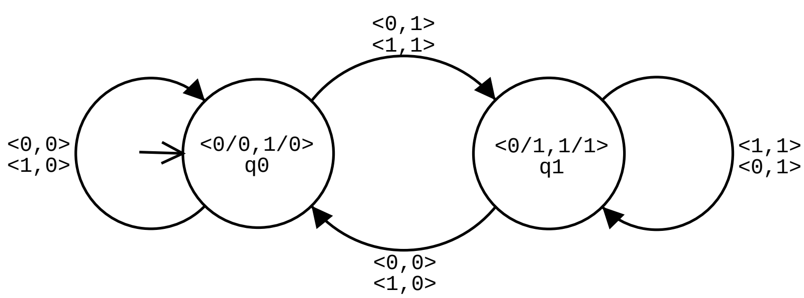

Figure 6.1: Transducer $\mathcal{A}_{LLim_0}$.

Example 6.2.2 The unary operator

$$LeftCont : nZSig(\{0,1\}) \longrightarrow nZSig(\{True, False\})$$

is defined as follows:

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

As we have seen in chapter 4 and since we are dealing with non-Zeno signals, *LeftCont* has three residuals:

$$\begin{aligned} LeftCont(z)(\tau) &= \begin{cases} True & \text{if } z \text{ is continuous at } \tau \\ False & \text{otherwise.} \end{cases} \\ G_0(z)(\tau) &= \begin{cases} True & \text{if } \tau = 0 \text{ and } z(0) = 0 \\ False & \text{if } \tau = 0 \text{ and } z(0) \neq 0 \\ LeftCont(z)(\tau) & \text{if } \tau > 0 \end{cases} \\ G_1(z)(\tau) &= \begin{cases} True & \text{if } \tau = 0 \text{ and } z(0) = 1 \\ False & \text{if } \tau = 0 \text{ and } z(0) \neq 1 \\ LeftCont(z)(\tau) & \text{if } \tau > 0 \end{cases} \end{aligned}$$

Thus, we construct the transducer $\mathcal{A}_{LeftCont} =< Q, q_0, \Sigma_{in}, \Sigma_{out}, n, out, \delta >$.

- Since there are three residuals, we need three states, i.e, $Q = \{q_0, q_1, q_2\}$. Each state is associated to a residual as follows:

$$q_0 \dashrightarrow LeftCont,\ q_1 \dashrightarrow G_0 \text{ and } q_2 \dashrightarrow G_1.$$

- The initial state is the state associated to the operator *LeftCont*, i.e., q_0.
- $\Sigma_{in} = \{0, 1\}$.
- $\Sigma_{out} = \{True, False\}$.
- Since *LeftCont* is an unary operator, we have $n = 1$.

State	Transition	δ	*out*
q_0	$0,0$	q_1	$0 \mapsto False$
q_0	$0,1$	q_2	$0 \mapsto False$
q_0	$1,0$	q_1	$1 \mapsto False$
q_0	$1,1$	q_2	$1 \mapsto False$
q_1	$0,0$	q_1	$0 \mapsto True$
q_1	$0,1$	q_2	$0 \mapsto True$
q_1	$1,0$	q_1	$1 \mapsto False$
q_1	$1,1$	q_2	$1 \mapsto False$
q_2	$0,0$	q_1	$0 \mapsto False$
q_2	$0,1$	q_2	$0 \mapsto False$
q_2	$1,0$	q_1	$1 \mapsto True$
q_2	$1,1$	q_2	$1 \mapsto True$

Table 6.2: δ and *out* functions in $\mathcal{A}_{LeftCont}$.

- We obtain the function *out* as follows:

$$out(q_0, 0) = LeftCont(Const_0)(0) = False$$
$$out(q_0, 1) = LeftCont(Const_1)(0) = False$$
$$out(q_1, 0) = G_0(Const_0)(0) = True$$
$$out(q_1, 1) = G_0(Const_1)(0) = False$$
$$out(q_2, 0) = G_1(Const_0)(0) = False$$
$$out(q_2, 1) = G_1(Const_1)(0) = True$$

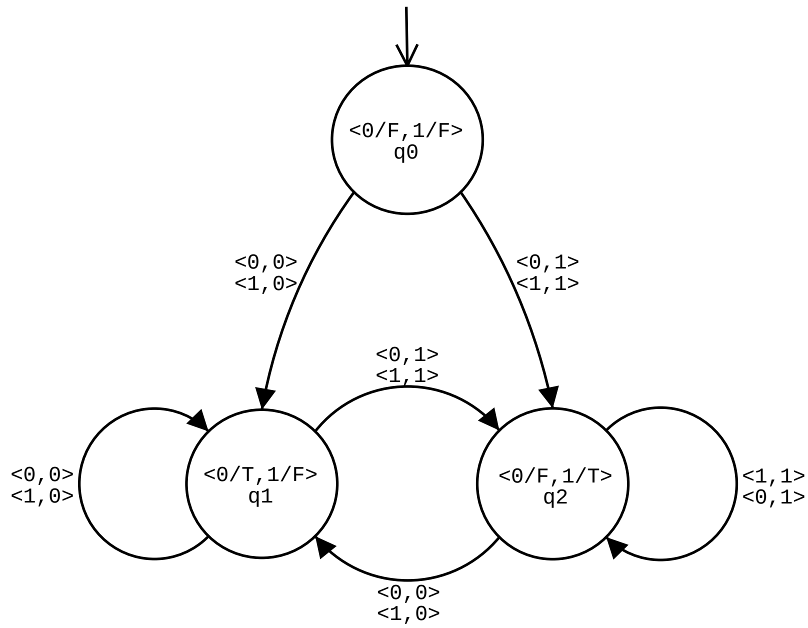

Figure 6.2: Transducer $\mathcal{A}_{LeftCont}$.

- Since Σ_{in} has two symbols, we know that there exist four different jumps:

  $Jump_{0\to0}$, $Jump_{0\to1}$, $Jump_{1\to0}$ and $Jump_{1\to1}$,

  i.e., there exist four possible transitions at each state. Therefore, the function δ is defined as follows:

$$\delta(0,0)q_0 = q_1 \text{ because } Res(LeftCont, Jump_{0\to0}, t) = G_0,$$
$$\delta(0,0)q_1 = q_1 \text{ because } Res(G_0, Jump_{0\to0}, t) = G_0,$$
$$\delta(0,0)q_2 = q_1 \text{ because } Res(G_1, Jump_{0\to0}, t) = G_0,$$
$$\delta(0,1)q_0 = q_2 \text{ because } Res(LeftCont, Jump_{0\to1}, t) = G_1,$$
$$\delta(0,1)q_1 = q_2 \text{ because } Res(G_0, Jump_{0\to1}, t) = G_1,$$
$$\delta(0,1)q_2 = q_2 \text{ because } Res(G_1, Jump_{0\to1}, t) = G_1,$$
$$\delta(1,0)q_0 = q_1 \text{ because } Res(LeftCont, Jump_{1\to0}, t) = G_0,$$
$$\delta(1,0)q_1 = q_1 \text{ because } Res(G_0, Jump_{1\to0}, t) = G_0,$$
$$\delta(1,0)q_2 = q_1 \text{ because } Res(G_1, Jump_{1\to0}, t) = G_0,$$
$$\delta(1,1)q_0 = q_2 \text{ because } Res(LeftCont, Jump_{1\to1}, t) = G_1,$$
$$\delta(1,1)q_1 = q_2 \text{ because } Res(G_0, Jump_{1\to1}, t) = G_1,$$
$$\delta(1,1)q_2 = q_2 \text{ because } Res(G_1, Jump_{1\to1}, t) = G_1,$$

for $t > 0$.

In figure 6.2 we give the transducer $\mathcal{A}_{LeftCont}$.

Example 6.2.3 The unary operator

$$LJV^1_{1,0} : nZSig(\{0,1\})^n \longrightarrow nZSig(\{0,1\})$$

is defined as follows:

$$LJV^1_{1,0}(x)(t) \quad = \quad \begin{cases} 0 & \text{if } t = 0 \\ x(\tau) & \text{if } t > 0 \; and \; \exists \; \tau, x \; is \; constant \; in \;]\tau, t[\\ & and \; x \; is \; not \; continuous \; at \; \tau. \end{cases}$$

As we will see in chapter 7, this operator have four residuals:

$$\begin{aligned} F^1_{1,0,0}(z)(\tau) &= LJV^1_{1,0}((Jump_{0\to 0}\rfloor^1; z))(\tau+1), \\ F^1_{1,0,1}(z)(\tau) &= LJV^1_{1,0}((Jump_{0\to 1}\rfloor^1; z))(\tau+1), \\ F^1_{1,1,0}(z)(\tau) &= LJV^1_{1,1}((Jump_{1\to 0}\rfloor^1; z))(\tau+1), \\ F^1_{1,1,1}(z)(\tau) &= LJV^1_{1,1}((Jump_{1\to 1}\rfloor^1; z))(\tau+1). \end{aligned}$$

Thus, we construct the transducer $\mathcal{A}_{LJV^1_{1,0}} =< Q, q_0, \Sigma_{in}, \Sigma_{out}, n, out, \delta >$.

- Since there are four residuals, we need four states, i.e, $Q = \{q_0, q_1, q_2, q_3\}$. Each state is associated to a residual as follows:

  $q_0 \dashrightarrow F^1_{1,0,0}$, $q_1 \dashrightarrow F^1_{1,0,1}$, $q_2 \dashrightarrow F^1_{1,1,0}$ and $q_3 \dashrightarrow F^1_{1,1,1}$.

- The initial state is the state associated to the operator $LJV^1_{1,0}$ $(= F^1_{1,0,0})$, i.e., q_0.

State	Transition	δ	*out*
q_0	$0, 0$	q_0	$0 \mapsto 0$
q_0	$0, 1$	q_1	$0 \mapsto 0$
q_0	$1, 0$	q_2	$1 \mapsto 0$
q_0	$1, 1$	q_3	$1 \mapsto 0$
q_1	$0, 0$	q_0	$0 \mapsto 0$
q_1	$0, 1$	q_1	$0 \mapsto 0$
q_1	$1, 0$	q_2	$1 \mapsto 0$
q_1	$1, 1$	q_1	$1 \mapsto 0$
q_2	$0, 0$	q_2	$1 \mapsto 1$
q_2	$0, 1$	q_1	$1 \mapsto 1$
q_2	$1, 0$	q_2	$0 \mapsto 1$
q_2	$1, 1$	q_3	$0 \mapsto 1$
q_3	$0, 0$	q_0	$1 \mapsto 1$
q_3	$0, 1$	q_1	$1 \mapsto 1$
q_3	$1, 0$	q_2	$0 \mapsto 1$
q_3	$1, 1$	q_3	$0 \mapsto 1$

Table 6.3: δ and *out* functions in $\mathcal{A}_{LJV^1_{1,0}}$.

- $\Sigma_{in} = \{0, 1\}$.
- $\Sigma_{out} = \{0, 1\}$.
- Since $LJV^1_{1,0}$ is an unary operator, we have $n = 1$.
- We obtain the function *out* as follows:

$$\begin{aligned}
out(q_0, 0) &= F^1_{1,0,0}(Const_0)(0) = 0\\
out(q_0, 1) &= F^1_{1,0,0}(Const_1)(0) = 0\\
out(q_1, 0) &= F^1_{1,0,1}(Const_0)(0) = 0\\
out(q_1, 1) &= F^1_{1,0,1}(Const_1)(0) = 0\\
out(q_2, 0) &= F^1_{1,1,0}(Const_0)(0) = 1\\
out(q_2, 1) &= F^1_{1,1,0}(Const_1)(0) = 1\\
out(q_3, 0) &= F^1_{1,1,1}(Const_0)(0) = 1\\
out(q_3, 1) &= F^1_{1,1,1}(Const_1)(0) = 1
\end{aligned}$$

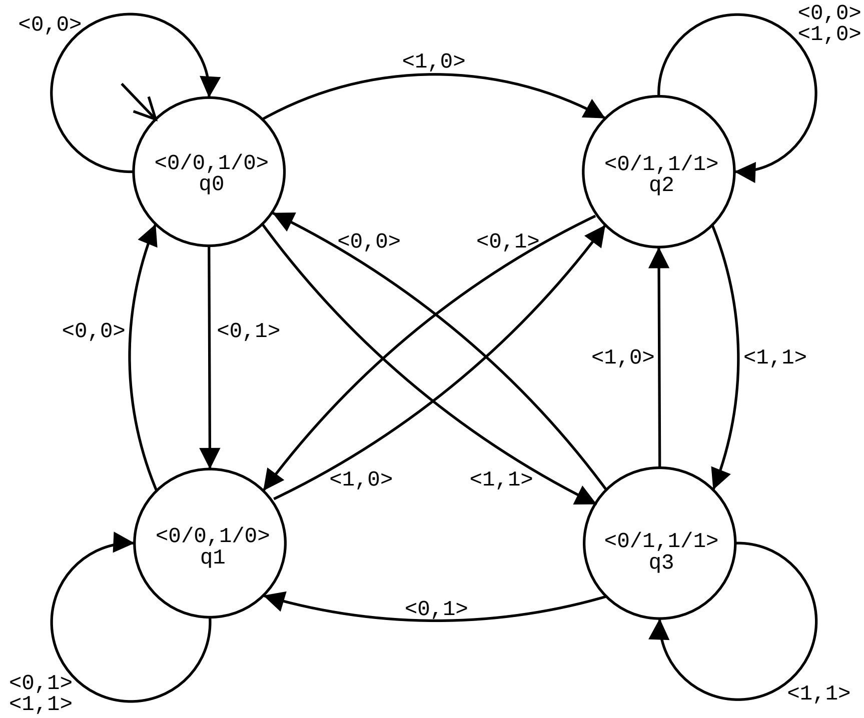

Figure 6.3: Transducer $\mathcal{A}_{LJV^1_{1,0}}$.

- Since Σ_{in} has two symbols, we know that there exist four different jumps:

$$Jump_{0\to0},\ Jump_{0\to1},\ Jump_{1\to0} \text{ and } Jump_{1\to1},$$

i.e., there exist four possible transitions at each state. Therefore, the function δ is defined as follows:

$$
\begin{aligned}
&\delta(0,0)q_0 = q_0 \text{ because } Res(F^1_{1,0,0}, Jump_{0\to 0}, t) = F^1_{1,0,0},\\
&\delta(0,0)q_1 = q_0 \text{ because } Res(F^1_{1,0,1}, Jump_{0\to 0}, t) = F^1_{1,0,0},\\
&\delta(0,0)q_2 = q_2 \text{ because } Res(F^1_{1,1,0}, Jump_{0\to 0}, t) = F^1_{1,1,0},\\
&\delta(0,0)q_3 = q_0 \text{ because } Res(F^1_{1,1,1}, Jump_{0\to 0}, t) = F^1_{1,0,0},\\
&\delta(0,1)q_0 = q_1 \text{ because } Res(F^1_{1,0,0}, Jump_{0\to 1}, t) = F^1_{1,0,1},\\
&\delta(0,1)q_1 = q_1 \text{ because } Res(F^1_{1,0,1}, Jump_{0\to 1}, t) = F^1_{1,0,1},\\
&\delta(0,1)q_2 = q_1 \text{ because } Res(F^1_{1,1,0}, Jump_{0\to 1}, t) = F^1_{1,0,1},\\
&\delta(0,1)q_3 = q_1 \text{ because } Res(F^1_{1,1,1}, Jump_{0\to 1}, t) = F^1_{1,0,1},\\
&\delta(1,0)q_0 = q_2 \text{ because } Res(F^1_{1,0,0}, Jump_{1\to 0}, t) = F^1_{1,1,0},\\
&\delta(1,0)q_1 = q_2 \text{ because } Res(F^1_{1,0,1}, Jump_{1\to 0}, t) = F^1_{1,1,0},\\
&\delta(1,0)q_2 = q_2 \text{ because } Res(F^1_{1,1,0}, Jump_{1\to 0}, t) = F^1_{1,1,0},\\
&\delta(1,0)q_3 = q_2 \text{ because } Res(F^1_{1,1,1}, Jump_{1\to 0}, t) = F^1_{1,1,0},\\
&\delta(1,1)q_0 = q_3 \text{ because } Res(F^1_{1,0,0}, Jump_{1\to 1}, t) = F^1_{1,1,1},\\
&\delta(1,1)q_1 = q_1 \text{ because } Res(F^1_{1,0,1}, Jump_{1\to 1}, t) = F^1_{1,0,1},\\
&\delta(1,1)q_2 = q_3 \text{ because } Res(F^1_{1,1,0}, Jump_{1\to 1}, t) = F^1_{1,1,1},\\
&\delta(1,1)q_3 = q_3 \text{ because } Res(F^1_{1,1,1}, Jump_{1\to 1}, t) = F^1_{1,1,1},
\end{aligned}
$$

for $t > 0$.

In figure 6.3 we give the transducer $\mathcal{A}_{LJV^1_{1,0}}$.

Results of this chapter

6.1.3 An operator over non-Zeno signals is a finite memory retrospective operator if and only if it is computable by a transducer.

6.1.4 Every finite memory retrospective operator over non-Zeno signals is speed independent.

6.1.5 The set of finite memory speed independent retrospective operator (over general signals) is at least uncountable.

Chapter 7

Circuits of operators

In this chapter we will study the possibility of characterizing the class of finite memory retrospective operators over non-Zeno signals by a function algebra [Clo99]. Circuits of finite memory retrospective operators will be considered and our main results will concern the construction of such circuits from a finite number of primitives.

The concept of function algebra is introduced[1] in [Clo99] and is clarified in the following definition.

Definition 7.0.1 (Function Algebra [Clo99]) An *operation* maps functions into functions. If $\mathcal{X}$ is a set of functions and OP is a set of operations, then $[\mathcal{X}; OP]$ denotes the smallest set of functions containing $\mathcal{X}$ and is closed under the operations of OP. The set $[\mathcal{X}; OP]$ is called a function algebra.

The set of functions that we pretend to characterize is the set of finite memory retrospective operators over non-Zeno signals, $\mathcal{O}_{FMR}$. In the next section we will specify the set $\mathcal{X}$ of primitive operators and verify that in fact $\mathcal{X}$ is a set of finite memory retrospective operators. The set of operations will be given in the second section and some properties involving them will be stated. The main result will arise in the third section, namely will be proved that the function algebra specified coincide precisely with the class $\mathcal{O}_{FMR}$.

7.1 Primitives

Intuitively we may consider some operators that seem to be most relevant or basic. Given a finite memory retrospective operator F over non-Zeno signals, its value at an instant t depends on the past values of the signal $\vec{x}$ given as argument. Thus, in order to compute the value of F at t, it will be useful to know the behavior of $\vec{x}$ before t, namely the last jump of $\vec{x}$ before t and the left limit of $\vec{x}$ at t.

Following these intuitions, we define now two primitive operators which are needed to get the main characterization results.

[1]However introduced before, the concept of function algebra was used for the first time to characterize a complexity classes in [Clo99].

Definition 7.1.1 (Left Limit $LLim_a$) Let Σ be a finite set and $a \in \Sigma$. We define the unary operator

$$LLim_a : nZSig(\Sigma) \longrightarrow nZSig(\Sigma)$$

as follows:

$$LLim_a(x)(t) = \begin{cases} a & \text{if } t = 0 \\ b & \text{if } t > 0 \text{ and } b \text{ is the left limit of } x \text{ at } t. \end{cases}$$

Definition 7.1.2 (Last Jump Value $LJV^n_{k,a}$) Let Σ be a finite set, $a \in \Sigma$ and $n, k \in \mathbb{N}$ such that $k \leq n$. We define the n-ary operator

$$LJV^n_{k,a} : nZSig(\Sigma)^n \longrightarrow nZSig(\Sigma)$$

as follows:

$$LJV^n_{k,a}(\vec{x})(t) = \begin{cases} a & \text{if } t = 0 \\ x_k(\tau) & \text{if } t > 0 \text{ and } \tau < t \text{ such that } \vec{x} \text{ is constant in }]\tau, t[\\ & \text{and } \vec{x} \text{ is not continuous at } \tau. \end{cases}$$

Since the left limit is undefined at the instant 0 and there is no jumps before that, we have introduced default values at $t = 0$ in the above definitions. Therefore, we grant that $LLim_a$ and $LJV^n_{k,a}$ are well defined at $t = 0$.

Example 7.1.3 Some applications of the above operators:

$$\begin{aligned} LLim_a(Const_b\rfloor^1; Jump_{c\to d})(0) &= a \\ LLim_a(Const_b\rfloor^1; Jump_{c\to d})(1) &= b \\ LLim_a(Const_b\rfloor^1; Jump_{c\to d})(2) &= d \end{aligned}$$

$$\begin{aligned} LJV^2_{1,a}((Const_b\rfloor^1; Jump_{c\to d}), (Jump_{b\to e}\rfloor^2; Jump_{c\to d}))(0) &= a \\ LJV^2_{2,a}((Const_b\rfloor^1; Jump_{c\to d}), (Jump_{b\to e}\rfloor^2; Jump_{c\to d}))(0) &= a \\ LJV^2_{1,a}((Const_b\rfloor^1; Jump_{c\to d}), (Jump_{b\to e}\rfloor^2; Jump_{c\to d}))(2) &= c \\ LJV^2_{1,a}((Const_b\rfloor^1; Jump_{c\to d}), (Jump_{b\to e}\rfloor^2; Jump_{c\to d}))(3) &= c \\ LJV^2_{2,a}((Const_b\rfloor^1; Jump_{c\to d}), (Jump_{b\to e}\rfloor^2; Jump_{c\to d}))(3) &= b \end{aligned}$$

These operators are evidently strong retrospective operators. We shall see that they are finite memory operators.

Proposition 7.1.4 *$LLim_a$ is a finite memory operator.*

Proof: Let $x \in Sig(\Sigma)$ and $t \in T^+$, by definition 2.1.16 we know that the residual G of $LLim_a$ with respect to x and t is defined as follows:

$$\begin{aligned} G(z)(t') &= LLim_a(x\rfloor^t; z)(t + t') \\ &= LLim_b(z)(t') \end{aligned}$$

where b is the left limit of x at t whenever $t > 0$, otherwise $b = a$. Thus, the set of residuals of $LLim_a$ is given by

$$\{LLim_a : a \in \Sigma\}.$$

Since Σ is a finite set, the set of residuals is finite and $LLim_a$ is a finite memory operator. $\square$

Proposition 7.1.5 *$LJV^n_{k,a}$ is a finite memory operator.*

Proof: Let $\vec{x} \in Sig(\Sigma)^n$ and $t \in T^+$, by definition 2.1.16 we know that the residual G of $LJV^n_{k,a}$ with respect to $\vec{x}$ and t is given by

$$\begin{aligned} G(\vec{z})(t') &= LJV^n_{k,a}((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n))(t+t') \\ &= F^n_{k,b,\xi}(\vec{z})(t') \end{aligned}$$

where $b = LJV^n_{k,a}(\vec{x})(t)$, $\xi =< c_1, ..., c_n >=< LLim_a(x_i)(t) : i = 1, ..., n >$ and $F^n_{k,b,\xi}$ is defined as follows:

$$F^n_{k,b,\xi}(\vec{z})(t') \quad = \quad LJV^n_{k,b}((Jump_{b\to c_1}\rfloor^1; z_1), ..., (Jump_{b\to c_n}\rfloor^1; z_n))(t'+1)$$

Therefore the set of residuals of $LJV^n_{k,a}$ is given by

$$\{F^n_{k,b,\xi} : b \in \Sigma \textit{ and } \xi \in \Sigma^n\}.$$

Since Σ is a finite set, the set of residuals is finite and $LJV^n_{k,a}$ is a finite memory operator. $\square$

Definition 7.1.6 (Pointwise Extension) If $f : \Sigma^n \longrightarrow \Sigma'$ is a function where Σ and Σ' are both finite non-empty sets, then we define the operator $P_f : nZSig(\Sigma)^n \longrightarrow nZSig(\Sigma')$ as the pointwise extension of f as follows:

$$P_f(\vec{x})(t) \quad = \quad f(\vec{x}(t))$$

Since the value of these operators only depends on the current instant, we see that these operators are retrospective and that have finite memory.

Definition 7.1.7 (Set $\mathcal{X}[\Sigma_1, ..., \Sigma_m]$ of Primitives) Given Σ_1, ..., Σ_m finite non-empty sets, the set $\mathcal{X}[\Sigma_1, ..., \Sigma_m]$ of primitives is defined as follows:

$$\begin{aligned} \mathcal{X}[\Sigma_1, ..., \Sigma_m] \quad = \quad & \{LLim_a : \ a \in \Sigma_i, \ i = 1, ..., m\} \\ & \cup \{LJV^n_{k,a} : \ a \in \Sigma_i, \ n, k \in \mathbb{N} \text{ s.t. } k \le n, \ i = 1, ..., m\} \\ & \cup \{P_f : \ n \in \mathbb{N}, \ f : \Sigma_i^n \longrightarrow \Sigma'_j, \ i, j = 1, ..., m\} \end{aligned}$$

7.2 Operations

We will discuss now which operators are needed in the set OP referred in definition 7.0.1.

In definition 5.1.2, we stated that the set of finite memory retrospective operators is closed under composition and naturally we will consider the composition $\circ$ as one of the available operations.

Definition 7.2.1 (Composition) Let $G : nZSig(\Sigma')^m \longrightarrow nZSig(\Sigma'')$ be a m-ary finite memory retrospective operator and let $F_i : nZSig(\Sigma)^n \longrightarrow nZSig(\Sigma')$ be also a n-ary finite memory operator, for $i = 1, ..., m$. We define the composition of G with $F_1, ..., F_m$,

$$\circ(G, F_1, .., F_m) : nZSig(\Sigma)^n \longrightarrow nZSig(\Sigma''),$$

as follows:

$$\circ(G, F_1, ..., F_m)(\vec{x}) \quad = \quad G(F_1(\vec{x}), ..., F_m(\vec{x})).$$

The state function studied in section 5.2.2 give us some ideas about another operation. As we have seen, sometimes we need to know previous values of operator to get its current value, so we define next the operation *Rec*.

Definition 7.2.2 (*Rec*) Let $G : Sig(\Sigma)^{n+1} \longrightarrow Sig(\Sigma)$ be a finite memory retrospective operator that verify the following conditions:

- is strong retrospective with respect to the last argument;
- given $x_1, ..., x_n, y, z \in Sig(\Sigma)$, $t > 0$ and $t' \in [0, t[$ such that x_i is constant in $]t', t[$ for all i and x_j is not continuous at t' for some j, if $y(\tau) = z(\tau)$ for $\tau \in [0, t'[$ then, $G(x_1, ..., x_n, y)(t) = G(x_1, ..., x_n, z)(t)$.

For an arbitrary $a \in \Sigma$, the operation *Rec* is defined as follows:

$$Rec(G)\vec{x}t = \begin{cases} G(x_1, ..., x_n, const_a)t & \text{if } t = 0 \\ G(x_1, ..., x_n, (Rec(G)(\vec{x})\rfloor^{t'}; const_a))t & \text{if } t > 0 \end{cases}$$

and clearly $Rec(G) : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$.

We will prove that the operators obtained with *Rec*, are well defined and are finite memory retrospective operators.

Given the operator $H_{\vec{x}} : Sig(\Sigma) \longrightarrow Sig(\Sigma)$ such that $H_{\vec{x}}(y) = G(\vec{x}, y)$ for $\vec{x} \in Sig(\Sigma)$ and making use of the ideas in chapter 3, we will study the fixpoints of $H_{\vec{x}}$ as fixpoints of a function over ω-strings. We observe that $Rec(G)(\vec{x})$ is a fixpoint of $H_{\vec{x}}$.

Proposition 7.2.3 *Let G be in the conditions of definition 7.2.2 and let $\vec{x} \in Sig(\Sigma)^n$. The operator $H_{\vec{x}}$, such that $H_{\vec{x}}(y) = G(\vec{x}, y)$ for $y \in Sig(\Sigma)$, is characterized by a function $K_{\vec{x}} : \Sigma^\omega \times \Sigma^\omega \longrightarrow \Sigma^\omega \times \Sigma^\omega$ with the property:*

if $\alpha_i = \beta_i$ and $\alpha'_i = \beta'_i$, for $i < n$, then $K(\alpha, \alpha')_i = K(\beta, \beta')_i$, for $i \leq n$.

Proof: If G is a finite memory operator, then G is speed independent as we have seen in proposition 6.1.4. Thus $H_{\vec{x}}$ is a finite memory operator and by proposition 3.2.3 there exists a function $K_{\vec{x}} : \Sigma^\omega \times \Sigma^\omega \longrightarrow \Sigma^\omega \times \Sigma^\omega$ that characterizes $H_{\vec{x}}$ and verifies the generalized SI condition in definition 3.2.1.

In order to prove the above property let $\alpha, \alpha', \beta, \beta' \in \Sigma^\omega$ such that $\alpha_i = \beta_i$ and $\alpha'_i = \beta'_i$ for $i < n$. Suppose also that y is the non-Zeno signal characterized by α, α', τ and z is the non-Zeno signal characterized by β, β', τ where the time scale $\tau = < t_i : i \in \mathbb{N}_0 >$ satisfies:

$$\text{if } t \leq t', \text{ then } t \in \tau \text{ iff } \exists\ i,\ x_i \text{ is not constant at } t,$$

for $t' \in T^+$ such that $\vec{x}$ is not constant at t' and such that $\vec{x}$ is constant at τ for $\tau > t'$ [2]. Thus we verify that $y(t) = z(t)$ for $t \in [0, t_{n-1}[$ and then we conclude by the given properties of G in definition 7.2.2 that

[2] Clearly, such t' may not exist. This condition express the minimality of the time scale τ needed to characterize the non-Zeno signal $\vec{x}$.

$$H_{\vec{x}}(y)(t) = H_{\vec{x}}(z)(t)$$

with $t \in [0, t_n[$. Therefore $K_{\vec{x}}(\alpha, \alpha')_i = K_{\vec{x}}(\beta, \beta')_i$ for $i \leq n$ because $K_{\vec{x}}$ is a characterization of $H_{\vec{x}}(y)$. $\square$

In appendix A we provide a few notes about partial orders that will be useful to prove the following results. Before that and since we will work with prefixes, we recall the concept of prefix.

Definition 7.2.4 Let $\alpha \in \Sigma^\omega$ be a ω-string, we define the set of its prefixes as:

$$P_\alpha = \{u \in \Sigma^* : \exists\ \beta \in \Sigma^\omega,\ \alpha = u\beta\}$$

Definition 7.2.5 (Set Closed for Prefixes) A set $X \subseteq \Sigma^*$ is closed under prefixes whenever it satisfies the following condition:

$$\forall\ u \in X,\ \forall\ v \in \Sigma^*, \text{ if } u = sv, \text{ then } s \in X.$$

Let $X \subseteq \Sigma^*$ be a set closed under prefixes such that there exists $u \in X$ with length n, for all $n \in \mathbb{N}$. Thus, it is clear that there exists at least one $\alpha \in \Sigma^\omega$ such that $P_\alpha \subseteq X$. Moreover, if for all $u \in X$ there exists only one $a \in \Sigma$ such that $ua \in Xv$, then $X = P_\alpha$ for some $\alpha \in \Sigma^\omega$ and we say that X characterizes univocally[3] α.

Proposition 7.2.6 *Let $\mathcal{P} = \{X \subseteq \Sigma^* : X \text{ is closed for prefixes}\}$, then $< \mathcal{P} \times \mathcal{P}, \leq>$ is a complete lattice with $\leq$ defined as:*

$$(X, X') \leq (Y, Y') \textit{ iff } X \subseteq Y \textit{ and } X' \subseteq Y'.$$

Proof: Let $(X, X'), (Y, Y'), (Z, Z') \in \mathcal{P} \times \mathcal{P}$. Clearly:

- $(X, X') \leq (X, X')$;
- if $(X, X') \leq (Y, Y')$ and $(Y, Y') \leq (X, X')$, then $(X, X') = (Y, Y')$;
- if $(X, X') \leq (Y, Y')$ and $(Y, Y') \leq (Z, Z')$, then $(X, X') \leq (Z, Z')$.

Therefore $< \mathcal{P} \times \mathcal{P}, \leq>$ is a partial order.

Let $\mathcal{S} \subseteq \mathcal{P} \times \mathcal{P}$ and $(Y, Y') \in \mathcal{P} \times \mathcal{P}$ such that, for all $(X, X') \in \mathcal{S}$, $(X, X') \leq (Y, Y')$. Then $\vee\mathcal{S} = (\bigcup_{(X,X')\in\ \mathcal{S}} X, \bigcup_{(X,X')\in\ \mathcal{S}} X') \leq (Y, Y')$, i.e, $\vee\mathcal{S}$ is the least upper bound of $\mathcal{S}$.

Suppose now that $(Z, Z') \in \mathcal{P} \times \mathcal{P}$ is such that, for all $(X, X') \in \mathcal{S}$, $(Z, Z') \leq (X, X')$. Then $(Z, Z') \leq \wedge\mathcal{S} = (\bigcap_{(X,X')\in\ \mathcal{S}} X, \bigcap_{(X,X')\in\ \mathcal{S}} X')$, i.e, $\wedge\mathcal{S}$ is the greatest lower bound of $\mathcal{S}$.

Clearly $\vee\mathcal{S} \in \mathcal{P} \times \mathcal{P}$ and $\wedge\mathcal{S} \in \mathcal{P} \times \mathcal{P}$, therefore $< \mathcal{P} \times \mathcal{P}, \leq>$ is a complete lattice. $\square$

Through this section we will work with the complete lattice $< \mathcal{P} \times \mathcal{P}, \leq>$. Given a subset $S \subseteq \mathcal{P} \times \mathcal{P}$, the least upper bound is easily given by

$$\vee\mathcal{S} = (\bigcup_{(X,X')\in\ \mathcal{S}} X, \bigcup_{(X,X')\in\ \mathcal{S}} X')$$

as we already see in proof of proposition 7.2.6 and the greatest lower bound is obtained simply by

[3] When we do not consider sets closed under prefixes, a ω-string α may be characterized by more sets than P_α, namely by any infinite subset of P_α.

$$\wedge\mathcal{S} = (\bigcap_{(X,X')\in\ \mathcal{S}} X, \bigcap_{(X,X')\in\ \mathcal{S}} X').$$

Thus, in $<\mathcal{P}\times\mathcal{P}, \leq>$, we have $\bot = (\{\epsilon\},\{\epsilon\})$.

Definition 7.2.7 Let $K_{\vec{x}} : \Sigma^\omega \times \Sigma^\omega \longrightarrow \Sigma^\omega \times \Sigma^\omega$ be the function obtained in proposition 7.2.3. Given $a_1a_2...a_n, a'_1a'_2...a'_n \in \Sigma^*$, for $n \in \mathbb{N}$, we define the function $K'_{\vec{x}}$ as follows:

$$K'_{\vec{x}}(a_1a_2...a_n, a'_1a'_2...a'_n) = (b_1b_2...b_{n+1}, b'_1b'_2...b'_{n+1})$$
$$\text{iff}$$
$$\forall\ \alpha,\alpha' \in \Sigma^\omega,\ \exists\ \beta,\beta' \in \Sigma^\omega,$$
$$K_{\vec{x}}(a_1a_2...a_n\alpha, a'_1a'_2...a'_n\alpha') = (b_1b_2...b_{n+1}\beta, b'_1b'_2...b'_{n+1}\beta').$$

Let $(X,X') \in \mathcal{P}\times\mathcal{P}$, we extend $K'_{\vec{x}}$ to $\mathcal{P}\times\mathcal{P}$ as follows:

$$K'_{\vec{x}}(X,X') = \{K'_{\vec{x}}(a_1...a_n, a'_1...a'_n) : a_1...a_n \in X,\ a'_1...a'_n \in X'\}\ \cup\ \{(\epsilon,\epsilon)\}.$$

Proposition 7.2.8 *$K'_{\vec{x}}$ is well defined.*

Proof: Let $a_1a_2...a_n, a'_1a'_2...a'_n \in \Sigma^*$, for $n \in \mathbb{N}$. Suppose that

$$K'_{\vec{x}}(a_1a_2...a_n, a'_1a'_2...a'_n) = (b_1b_2...b_{n+1}, b'_1b'_2...b'_{n+1}),$$

by definition 7.2.7, we have

$$\forall\ \alpha,\alpha' \in \Sigma^\omega,\ \exists\ \beta,\beta' \in \Sigma^\omega,$$
$$K_{\vec{x}}(a_1a_2...a_n\alpha, a'_1a'_2...a'_n\alpha') = (b_1b_2...b_{n+1}\beta, b'_1b'_2...b'_{n+1}\beta')$$

and by proposition 7.2.3 we conclude that $(b_1b_2...b_{n+1}, b'_1b'_2...b'_{n+1})$ is univocally determined. Thus, the extension of $K'_{\vec{x}}$ to $\mathcal{P}\times\mathcal{P}$ is well defined. $\square$

Proposition 7.2.9 *Let $K'_{\vec{x}}$ be the extension obtained in definition 7.2.7. This function is defined from $\mathcal{P}\times\mathcal{P}$ to $\mathcal{P}\times\mathcal{P}$ and is continuous.*

Proof: Let $(X,X') \in \mathcal{P}\times\mathcal{P}$. By the property of $K'_{\vec{x}}$ in proposition 7.2.3, we know that if $a_1a_2...a_n \in X$, $a'_1a'_2...a'_n \in X'$ and

$$K'_{\vec{x}}(a_1a_2...a_n, a'_1a'_2...a'_n) \quad = \quad (b_1b_2...b_nb_{n+1}, b'_1b'_2...b'_nb'_{n+1}),$$

then, for every $k \leq n$,

$$K'_{\vec{x}}(a_1a_2...a_k, a'_1a'_2...a'_k) \quad = \quad (b_1b_2...b_kb_{k+1}, b'_1b'_2...b'_kb'_{k+1}).$$

Thus, since $\{(\epsilon,\epsilon)\} \in K'_{\vec{x}}(X,X')$ by definition 7.2.7, $K'_{\vec{x}}(X,X')$ is closed under prefixes, i.e., $K'_{\vec{x}}(X,X') \in \mathcal{P}\times\mathcal{P}$.

Let $\mathcal{S} \subseteq \mathcal{P}\times\mathcal{P}$ be a directed subset. If $(X,X') \in \mathcal{S}$, then $K'_{\vec{x}}(X,X') \leq K(\vee\mathcal{S})$ where $\vee\mathcal{S} = (\bigcup_{(X,X')\in\ \mathcal{S}} X, \bigcup_{(X,X')\in\ \mathcal{S}} X')$. Clearly

$$\vee K'_{\vec{x}}(\mathcal{S}) \leq K(\vee\mathcal{S}).$$

Suppose that $(b_1b_2...b_nb_{n+1}, b'_1b'_2...b'_nb'_{n+1}) \in \vee K'_{\vec{x}}(\mathcal{S})$, for some $n \in \mathbb{N}_0$, then there exists $(a_1a_2...a_n, a'_1a'_2...a'_n) \in \vee\mathcal{S}$ such that

$$K'_{\vec{x}}(a_1a_2...a_n, a'_1a'_2...a'_n) \quad = \quad (b_1b_2...b_nb_{n+1}, b'_1b'_2...b'_nb'_{n+1}).$$

Thus, there exist $(X_1,X'_1),(X_2,X'_2) \in \mathcal{S}$ such that

$$a_1a_2...a_n \in X_1 \text{ and } a'_1a'_2...a'_n \in X_2.$$

Because $\mathcal{S}$ is directed, we know that there exists $(Y, Y') \in \mathcal{S}$ such that

$$(X_1, X'_1) \leq (Y, Y') \text{ and } (X_2, X'_2) \leq (Y, Y')$$

and clearly

$$(b_1b_2...b_nb_{n+1}, b'_1b'_2...b'_nb'_{n+1}) \in K'_{\vec{x}}(Y, Y').$$

As $(Y, Y') \leq \vee\mathcal{S}$, we have that $(b_1b_2...b_nb_{n+1}, b'_1b'_2...b'_nb'_{n+1}) \in K'_{\vec{x}}(\vee\mathcal{S})$, i.e.

$$K'_{\vec{x}}(\vee\mathcal{S}) \leq \vee K(\mathcal{S}).$$

Therefore $K'_{\vec{x}}(\vee\mathcal{S}) = \vee K'_{\vec{x}}(\mathcal{S})$, $K'_{\vec{x}}$ is continuous. □

Since we have a continuous function, it is now possible find its fixpoint. In the following three propositions we will find that and verify that *Rec*(G) is well defined as promised.

Proposition 7.2.10 *If $K'_{\vec{x}}$ is the extension obtained in definition 7.2.7, then $K'_{\vec{x}}$ has a least fixpoint (P, P') and*

$$(P, P') = \bigvee\{K'^n_{\vec{x}}(\{\epsilon\}, \{\epsilon\}) : n \in \mathbb{N}_0\}.$$

Proof: Directly from propositions 7.2.9 and A.0.8. □

Since for all $u \in P$ there exists only one $a \in \Sigma$ such that $ua \in P$, we conclude that P characterizes an ω-string ξ. Similarly, we conclude that P' characterizes an ω-string ξ' and the following result holds.

Proposition 7.2.11 *Let ξ, ξ' be the ω-strings characterized by the sets P, P' of proposition 7.2.10, $\vec{x} = (x_1, ..., x_n) \in Sig(\Sigma)^n$ and τ be a time scale such that*

$$\text{if } t \leq t', \text{ then } t \in \tau \text{ iff } \exists\ i,\ x_i \text{ is not constant at } t,$$

for $t' \in T^+$ such that $\vec{x}$ is not constant at t' and such that $\vec{x}$ is constant at τ for $\tau > t'$. Then the non-Zeno signal characterized by ξ, ξ', τ is a fixpoint of $H_{\vec{x}}$.

Proof: Since ξ, ξ' are characterized by P, P' respectively and P, P' is a fixpoint of $K'_{\vec{x}}$, we have that $K_{\vec{x}}(\xi, \xi') = (\xi, \xi')$. This function $K'_{\vec{x}}$ is a characterization of $H_{\vec{x}}$ as we have seen in proposition 7.2.3 and then, if ξ, ξ', τ characterizes z, $K'_{\vec{x}}(\xi, \xi'), \tau$ characterizes $H_{\vec{x}}(z)$. But $K'_{\vec{x}}(\xi, \xi') = (\xi, \xi')$, therefore $H_{\vec{x}}(z) = z$, i.e., z is a fixpoint of $H_{\vec{x}}$. □

Proposition 7.2.12 *Rec*(G) *is well defined.*

Proof: Given $\vec{x} \in Sig(\Sigma)^n$ and if we look at the definition 7.2.2, we verify that the operator *Rec* constructs the least fix point found in proposition 7.2.11. Therefore it is well defined since that there exists only one least fixpoint when we fix the scale τ as we did in proposition 7.2.11. □

About retrospectivity and memory finiteness of *Rec*(G) we leave the following result.

Proposition 7.2.13 *Let G be in the conditions of definition 7.2.2, then* *Rec*(G) *is a finite memory retrospective operator.*

Proof: Looking at definition 7.2.2 we can see that G is retrospective and therefore $Rec(G)$ is obviously retrospective. If $F = Rec(G)$ and $t = 0$, then we get from definition 2.1.16:

$$\begin{aligned} & Res(F, \vec{x}, t)(\vec{z})(\tau) \\ = \ & F(\vec{x})(t+\tau) \\ = \ & G(x_1, ..., x_n, Const_a)(\tau). \end{aligned}$$

If $t > 0$, then we have:

$$\begin{aligned} & Res(F, \vec{x}, t)(\vec{z})(\tau) \\ = \ & F((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n))(t+\tau) \\ = \ & G((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n), (F((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n))\rfloor^{t'}; Const_a))(t+\tau) \\ = \ & G((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n), (((F\vec{x})\rfloor^t; Res(F, \vec{x}, t)\vec{z})\rfloor^{t'}; Const_a))(t+\tau) \\ = \ & G((x_1\rfloor^t; z_1), ..., (x_n\rfloor^t; z_n), ((F\vec{x})\rfloor^t; ((Res(F, \vec{x}, t)\vec{z})\rfloor^{t'-t}; Const_a)))(t+\tau) \\ = \ & Res(G, (x_1, ..., x_n, F\vec{x}), t)(z_1, ..., z_n, (Res(F, \vec{x}, t)\vec{z})\rfloor^{t'-t}; Const_a))(\tau), \end{aligned}$$

where $t' < t + \tau$, i.e, $t' - t < \tau$.

As $Res(G, (x_1, ..., x_n, F\vec{x}), t)$ is a finite memory retrospective operator, by proposition 7.2.12, the number of residuals of F and G is the same, i.e., F is a finite memory operator. $\square$

We may now give the complete definition of $[\mathcal{X}[\Sigma_1, ..., \Sigma_m], OP]$.

Definition 7.2.14 Given Σ_1, ..., Σ_m finite non-empty sets, the function algebra $[\mathcal{X}[\Sigma_1, ..., \Sigma_m]; OP]$ is defined as follows:

$$\begin{aligned} OP &= \{\circ, Rec\}, \\ \mathcal{X}[\Sigma_1, ..., \Sigma_m] &= \{LLim_a : \ a \in \Sigma_i, \ i = 1, ..., m\} \\ & \cup \{LJV^n_{k,a} : \ a \in \Sigma_i, \ n, k \in \mathbb{N} \text{ s.t. } k \leq n, \ i = 1, ..., m\} \\ & \cup \{P_f : \ n \in \mathbb{N}, \ f : \Sigma_i^n \longrightarrow \Sigma'_j, \ i, j = 1, ..., m\} \end{aligned}$$

The following important result holds.

Theorem 7.2.15 *If F is in $[\mathcal{X}[\Sigma_1, ..., \Sigma_m]; OP]$, then F is a finite memory retrospective operator.*

Proof: Directly from the above propositions. $\square$

7.3 Circuits

If we consider the set of circuits obtained from the function algebra defined in the previous section, we know that all representable operators are finite memory retrospective operators by theorem 7.2.15. Now we will prove that all of them can be represented by these circuits.

Before that we note that we will use the operation Rec with $\vec{x}$ such that each x_i may be defined over different alphabets. If we look at the above section, we easily verify that the results continue to be true, in fact they are independent of the alphabets.

Theorem 7.3.1 *If $G : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$ is a finite memory retrospective operator, then there is a circuit obtained from the function algebra $[\mathcal{X}[\Sigma_1, ..., \Sigma_m]; OP]$ that represents G.*

Proof: Given a finite memory retrospective operator $G : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$, let

$$\{G_0, ..., G_k\}$$

with $G_0 = G$ be the set of its residuals. We define now the function

$$g : \Sigma^{2n} \times \{0, ..., k\} \longrightarrow \{0, ..., k\}$$

as follows:

$$g(b_1, ..., b_n, a_1, ..., a_n, p) = q$$
$$\text{iff}$$
$$Res(G_p, Jump_{a_1 \to b_1}, ..., Jump_{a_n \to b_n}, t) = G_q$$

where $t > 0$, $\vec{a} = (a_1, ..., a_n)$ and $\vec{b} = (b_1, ..., b_n)$.

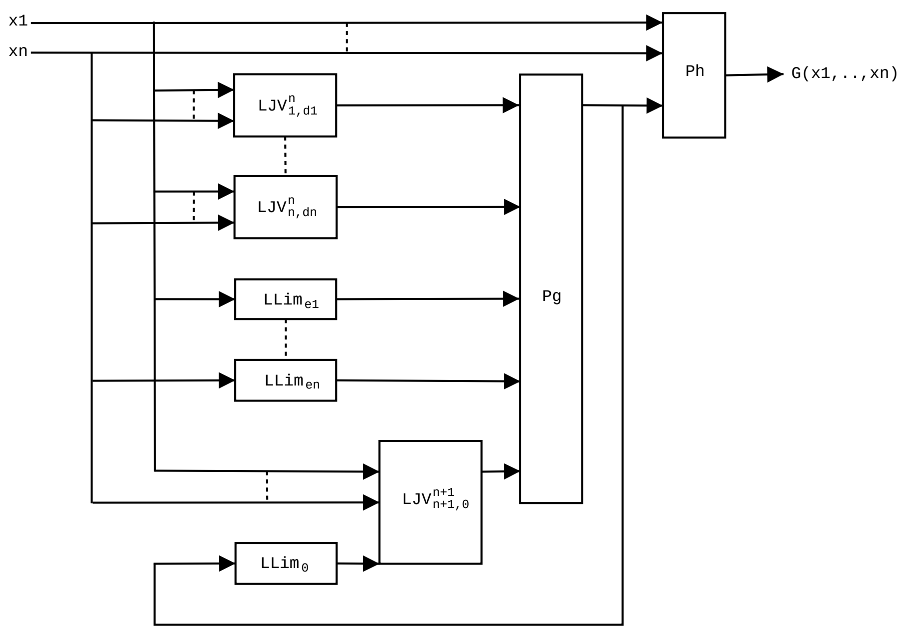

Figure 7.1: Construction schema for theorem 7.3.1.

Making use of the function algebra $[\mathcal{X}[\Sigma_1, ..., \Sigma_m]; OP]$, we can define

$$\begin{aligned} J_g(\vec{x}, r) \quad = \quad & P_g(LLim_{e_1}(x_1), ..., LLim_{e_n}(x_n), \\ & LJV^n_{1,d_1}(\vec{x}), ..., LJV^n_{n,d_n}(\vec{x}), \\ & LJV^{n+1}_{n+1,0}(\vec{x}, LLim_0(r))) \end{aligned}$$

with $\vec{e}$ and $\vec{d}$ such that $g(d_1, ..., d_n, e_1, ..., e_n, 0) = 0$ [4]. Looking at the definitions 7.1.1 and 7.1.2, we verify that J_g is in the conditions for the application of operator *Rec*.

[4] To guarantee this requisite we introduce if needed new symbols in the domain of g, that will not affect the result.

Given $\vec{x} \in Sig(\Sigma)^n$, we have

$$\begin{aligned} Rec(J_g)(\vec{x})(t) &= P_g(LLim_{e_1}(x_1), ..., LLim_{e_n}(x_n), \\ &\quad LJV^n_{1,d_1}(\vec{x}), ..., LJV^n_{n,d_n}(\vec{x}), \\ &\quad LJV^{n+1}_{n+1,0}(\vec{x}, LLim_0(Rec(J_g)\vec{x})))(t) \\ &= g(LLim_{e_1}(x_1)(t), ..., LLim_{e_n}(x_n)(t), \\ &\quad LJV^n_{1,d_1}(\vec{x})(t), ..., LJV^n_{n,d_n}(\vec{x})(t), \\ &\quad LJV^{n+1}_{n+1,0}(\vec{x}, LLim_0(Rec(J_g)(\vec{x})))(t)) \end{aligned}$$

If $t' \in [0, t[$ is such that x_i is constant in $]t', t[$ for all i and x_j is not continuous at t' for some j, for each i

$$x_i(t) = x_i\rfloor^{t'}; Jump_{LJV^n_{i,d_i}(\vec{x}) \rightarrow LLim_{e_i}(x_i)}(t)$$

and $G_{LJV^{n+1}_{n+1,0}(\vec{x}, LLim_0(Rec(J)(\vec{x})))(t)}$ is the residual of G with respect to $\vec{x}$ and t'. By definition of g:

$$G_{Rec(J)(\vec{x})(t)} = Res(G, \vec{x}, t)$$

Defining $h : \Sigma^n \times \{0, ..., k\} \longrightarrow \Sigma$ as

$$h(\vec{a}, j) = b \text{ iff } G_j(Const_{a_1}, ..., Const_{a_n})(0)$$

we get $G(\vec{x})(t) = P_h(x_1, ..., x_n, Rec(J_g)(\vec{x}))(t)$. This is the circuit of G which is exemplified in figure 7.1. $\square$

Theorems 7.2.15 and 7.3.1 taken together state that the function algebra $[\mathcal{X}[\Sigma_1, ..., \Sigma_m]; OP]$ coincide precisely with the set of finite memory retrospective operators, i.e.,

$$[\mathcal{X}[\Sigma_1, ..., \Sigma_m]; OP] = \mathcal{O}_{FMR}.$$

7.4 Examples

In this section we will construct circuits for three finite memory retrospective operators, some of them already studied in chapter 4. We will use the function algebra $[\mathcal{X}[\Sigma_1, ..., \Sigma_n], OP]$, however the general construction schema provided in theorem 7.3.1 will not be followed[5].

Example 7.4.1 $Const_b \in nZSig(\Sigma)$

$$Const_b(t) = b$$

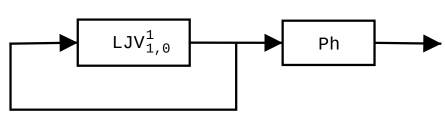

Figure 7.2: $Const_b$ circuit.

$Const_b$ is obviously a 0-ary finite memory retrospective operator with only one residual, $Const_b$. Therefore it is possible to represent this operator by the circuit of figure 7.2, where $h : \{0\} \longrightarrow \Sigma$ is such that $h(0) = b$.

[5]The circuits obtained by the general construction schema become more complex.

Example 7.4.2 $Jump_{a \to b} \in nZSig(\Sigma)$

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

As we have seen in chapter 4, this signal is a 0-ary finite memory retrospective operator with two residuals:

$$Jump_{a \to b}, \text{ for } t = 0;$$
$$Const_b, \text{ for } t > 0.$$

In order to get a circuit to $Jump_{a \to b}$ is enough compose the circuit of figure 7.2 with $LLim_a$ as we did in figure 7.3.

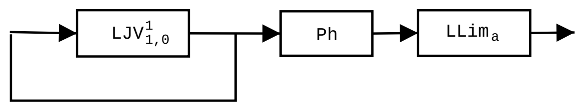

Figure 7.3: $Jump_{a \to b}$ circuit.

Example 7.4.3 $LeftCont : nZSig(\Sigma) \longrightarrow nZSig(\{True, False\})$

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

Clearly we can get this operator as follows:

$$LeftCont(x) = P_{=}(x, LLim_w(x))$$

where w is a symbol such that $w \notin \Sigma$ and $P_{=}$ is the pointwise extension of

$$=: (\Sigma \cup \{w\})^2 \longrightarrow \{True, False\}$$

defined as

$$= (a, b) = \begin{cases} True & \text{if } a = b \\ False & \text{otherwise.} \end{cases}$$

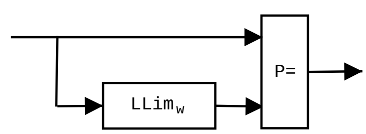

Figure 7.4: $LeftCont$ circuit.

Therefore, in figure 7.4, we give a circuit to $LeftCont$.

Results of this chapter

7.1.4 $LLim_a$ is a finite memory operator.

7.1.5 $LJV^n_{k,a}$ is a finite memory operator.

7.2.15 If F is in $[\mathcal{X}[\Sigma_1, ..., \Sigma_m]; OP]$, then F is a finite memory retrospective operator.

7.3.1 If $F : Sig(\Sigma)^n \longrightarrow Sig(\Sigma)$ is a finite memory retrospective operator, then there is a circuit obtained from the function algebra $[\mathcal{X}[\Sigma_1, ..., \Sigma_m]; OP]$ that represents F.

Chapter 8

Final Remarks

The behavior of finite devices with multiple input channels and operating in continuous time has been formalized in chapter 2, in fact we have studied the behavior of such devices operating in a time set T, where T is a subgroup of $\mathbb{R}$. This formalization was performed by looking at the classical theory of automata, i.e., looking at discrete time devices. Many postulates of automata theory were analyzed, the concept of time set was formalized in detail and the proofs of propositions 2.2.5 and 2.2.7 were provided, these are our main contributions in this chapter 2 with respect to [Rab97, RT98, PRT01].

A characterization of speed independent operators over non-Zeno signals and over right open signals was given in chapter 3, together with also some related concepts. With the lift to continuous time comes up some properties of signals invisible at discrete time, for example majority of signals will be sensible to expansion and compression of time. In this chapter our contributions with respect to [Rab97] are the complete proofs of the propositions 3.1.3 and 3.2.3.

In chapter 4, many examples were provided and studied in detail and many properties have been clarified.

In chapter 5, we studied closure properties of operators over signals and some properties of finite memory retrospective operators, which have permitted in chapter 6 to introduce the concept of a transducer and to define a representation of finite memory retrospective operators. With respect to [Rab97], the complete proofs of propositions 5.1.1, 5.1.2 and 5.2.8 were provided.

The representation of finite memory operators found in [Rab97] was discussed in chapter 6. Our contribution concerns transducers for n-ary operators and illustrative examples of automata, which construction was discussed in detail.

We note that the contents of chapters 5 and 6 relay in the characterization given in chapter 3.

Circuits of operators were introduced in chapter 7. The concept of function algebra in [Clo99] was used in order to obtain an algebra of finite memory retrospective operators. Our main contribution is the proof of the equivalence between this algebra of operators and the set of finite memory retrospective operators.

It is important to note that we did not use time delay operators, which are commonly used in the classical theory of circuits. In order to know the values of signals at previous intants we used the $LLim$ operator and the LJV operator

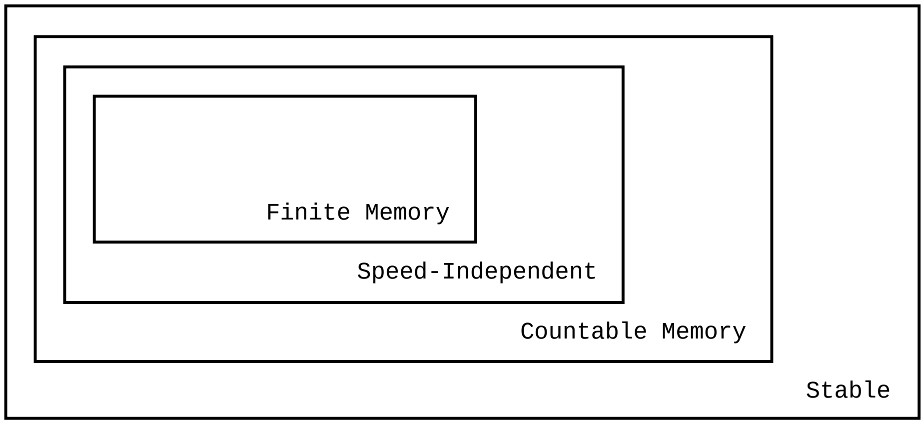

Figure 8.1: Properties of the retrospective operators that map non-Zeno signals on non-Zeno signals.

and with these operators we have proved that it is possible construct circuits for any finite memory retrospective operator.

The chapter 7 includes also examples of circuits for some operators. However not used for these examples, the general construction schema provided in the proof of theorem 7.3.1 permits the construction of circuits for any finite memory retrospective operator using the function algebra $[\mathcal{X}[\Sigma_1, ..., \Sigma_n], OP]$.

We recall two main properties showed within the dissertation.

In first place it follows straightforward that finite memory strong retrospective operators map non-Zeno signals into left open signals. Since they have finite memory, they are stable and speed independent and, if a non-Zeno signal is not constant at t, the output at t depends only on previous instants and therefore it must be equal to previous outputs.

A second property that follows is the fact that countable memory operators may not be representable by a transducer with a countable number of states because the set $\{0, ..., n, ...\} \to \{0, ..., n, ...\}$ in definition 5.2.7 have an uncountable number of elements[1].

In figure 8.1, we summarize the inclusions stated among the properties of retrospective operators on non-Zeno signals. We have stated in theorem 6.1.4 the inclusion $Finite\ Memory \subset$ *Speed-Independent*, which is proper given the example 4.0.9. The inclusion *Speed-Independent* $\subset$ *Countable* $Memory$ was proved in corollary 3.2.4. In proposition 3.0.7 we proved the inclusion *Countable* $Memory \subset$ *Stable* for operators that map non-Zeno signals into non-Zeno signals, therefore this inclusion is true for retrospective operators with this property. We also proved in proposition 6.1.4 that the finite memory retrospective operators are stable and in proposition 3.0.9 that the speed independent retrospective operators are stable, however it is now clear that these results follow immediately from the above inclusions.

The research around automata over continuous time, started in [Tra98] and [Rab97], was expanded in chapter 7, where we succeed to develop a theory of circuits of retrospective operators to fully characterize the class of finite memory

[1] As we know, the set of functions from $\mathbb{N}$ to $\mathbb{N}$ is uncountable.

retrospective operators.

However many other open problems are to be solved and the field to be enlarged towards a theory on foundations of hybrid systems. Boaz Trakhtenbrot gave us a copy of his personal notes, where many interesting problems were designed, e.g., the problem of oracles (the relativization problem), a theory of reducibilities between operators has been considered and sketched, a theory of reliable feedback has been developed in some depth, etc. We thus hope that such a fundational seminal work prosper in the near future.

Bibliography

[AD94] R. Alur and D. L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126:183–235, 1994.

[Clo99] P. Clote. Computational models and function algebras. In E. R. Griffor, editor, *Handbook of Computability Theory*, pages 589–681. Elsevier, 1999.

[HMU01] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison Wesley, 2 edition, 2001.

[KT65] N. E. Kobrinskii and B. A. Trakthtenbrot. *Introduction to the Theory of Finite Automata.* North-Holland Publishing Company Amsterdam, 1965.

[Llo87] J. W. Lloyd. *Fundations of Logic Programming.* Symbolic Computation - Artificial Intelligence. Springer-Verlag, second, extended edition, April 1987.

[LSFV96] N. Lynch, R. Seagala, and H. B. Weiberg F. Vaandreger. Hybrid I/O automata. In R. Alur, T. A. Henzinger, and E. D. Sontag, editors, *Lecture Notes in Computer Science - Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 496–510. Springer-Verlag, 1996.

[PRT01] D. Pardo, A. Rabinovich, and B. A. Trakhtenbrot. Synchronous Circuits over Continuous Time: Feedback Reliability and Completeness. Technical report, School of Computer Science, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel, 2001.

[Rab97] A. Rabinovich. Automata over Continuous Time. To appear in Theoretical Computer Science in print, 1997.

[Rab98] A. Rabinovich. On Translations of Temporal Logic of Actions into Monadic Second Order Logic. *Theoretical Computer Science*, 193:197–214, 1998.

[RT98] A. Rabinovich and B. A. Trakhtenbrot. From Finite Automata toward Hybrid Systems (Extended Abstract). In *FCT: Fundamentals (or Foundations) of Computation Theory*, 1998.

[Son90] E. D. Sontag. *Mathematical Control Theory — Deterministic Finite Dimensional Systems*, volume 6 of *Texts in Applied Mathematics*. Springer-Verlag, June 1990.

[Tra98] B. Trakhtenbrot. Automata and Hybrid Systems. Technical Report 153, Uppsala University, February 1998.

[Tra99] B. A. Trakhtenbrot. Automata and Their Interaction: Definitional Suggestions. In *Fundamentals of Computation Theory*, pages 54–89, 1999.

Index

Appendix A

Notes about Partial Order Theory

In this appendix we provide some definitions and results about partial orders that will be important for our goal in chapter 7.

Definition A.0.1 (Partial Order) A relation R on a set S is a *partial order* if the following conditions are satisfied:

(a) xRx, for all $x \in S$.

(b) xRy and yRx imply $x = y$, for all $x, y \in S$.

(c) xRy and yRz imply xRz, for all $x, y, z \in S$.

Definition A.0.2 (Upper and Lower Bounds) Let S be a set equiped with a partial order $\leq$. Then $a \in S$ is an *upper bound* of a subset X of S if $x \leq a$, for all $x \in X$. Similarly, $b \in S$ is a *lower bound* of X if $b \leq x$, for all $x \in X$.

Definition A.0.3 (Least Upper and Greatest Lower Bounds) .
Let S be a set equiped with a partial order $\leq$. Then $a \in S$ is the *least upper bound* of a subset X of S if a is an upper bound of X and, for all upper bounds a' of X, we have $a \leq a'$. Similarly, $b \in S$ is the *greatest lower bound* of a subset X of S if b is a lower bound of X and, for all lower bounds b' of X, we have $b' \leq b$.

Let $L = (S, \leq)$ be a partial order. As usual, for $X \subset L$, we will write $\vee_L X$ for the least upper bound and $\wedge_L X$ for the greatest lower bound. $\perp_L$ will denote the least upper bound of L when it exists.

Definition A.0.4 (Complete Lattice) A partially ordered set L is a *complete lattice* if $\vee_L X$ and $\wedge_L X$ exist for every subset X of L.

Definition A.0.5 (Directed Set) Let L be a complete lattice and $X \subseteq L$. We say X is *directed* if every finite subset of X has an upper bound in X.

Definition A.0.6 (Continuous Mapping) Let $L = (S, \leq)$ be a complete lattice and $T : S \longrightarrow S$ be a map. We say that T is *continuous* if $T(\vee_L X) = \vee_L T(X)$, for every directed subset X of L.

Definition A.0.7 (Least (Greatest) Fixpoint) Let L be a complete lattice and $T : L \longrightarrow L$ be a mapping. We say $a \in L$ is the *least fixpoint* of T if a is a fixpoint (that is, T(a)=a) and for all fixpoints b of T, we have $a \leq b$. Similarly, we define *greatest fixpoint*.

The following result have an important role in our study and is due to Kleene, which is supported by an earlier result due to Knaster and Tarski.

Proposition A.0.8 *Let $L = (S, \leq)$ be a complete lattice and $T : S \longrightarrow S$ be a continuous map. Then $lfp(T) = \vee_L \{T^n(\bot_L) : n \in \mathbb{N}_0\}$.*

If L is a complete lattice and $T : S \longrightarrow S$ is a continuous map, then we know that it is monotonic, i.e, $T(X) \leq T(Y)$ whenever $X \leq Y$. Therefore $\{T^n(\bot) : n \in \mathbb{N}_0\}$ is a directed set. In practice when we want to obtain a least fix point is enough to compute the $\vee_L \{T^n(\bot_L) : n \in \mathbb{N}_0\}$.