Finite Variability Interpretation of Monadic Logic of Order

A. Rabinovich
Department of Computer Science
Raymond and Beverly Sackler Faculty of Exact Sciences
Tel Aviv University
Tel Aviv 69978, Israel
e.mail: rabino@math.tau.ac.il

Abstract

We consider an interpretation of monadic second-order logic of order in the continuous time structure of finitely variable signals. We provide a characterization of the expressive power of monadic logic. As a by-product of our characterization we show that many fundamental theorems which hold in the discrete time interpretation of monadic logic are still valid in the continuous time interpretation.

1 Introduction

In the recent years systems whose behavior change in the continuous (real) time were extensively investigated. Hybrid and control systems are prominent examples of real time systems.

A run of a real time system is represented by a function from non-negative reals into a set of values - the instantaneous states of a system. Such a function will be called a signal. Usually, there is a further restriction on behavior of continuous time systems. For example, a function that gives value q_0 for the rationals and value q_1 for the irrationals is not accepted as a 'legal' signal.

A requirement that is often imposed in the literature is that in every bounded time interval a system can change its state only finitely many times. This requirement is called a finite variability (or a non-Zeno) requirement. A function from the non-negative real into a set Σ that satisfies this requirement is called a finitely variable signal. If in addition such function x satisfies the requirement that for every t there is $\epsilon > 0$ such that x is constant on $[t,\ t+\epsilon)$, then it is called a right continuous signal. It is clear that finite variability and right continuous requirements are not metric requirements.

Recall that the language $L_2^<$ of monadic second order logic of order contains individual variables, second order variables and the binary predicate $<$. In the discrete time structure ω (this structure will be defined precisely in section 3), the individual variables are interpreted as natural numbers, the second order variables as monadic predicates (monadic functions from the natural numbers into the booleans), and $<$ is the standard order on the set of natural numbers.

A monadic formula $\phi(X)$ with one free predicate variable X defines a set of ω-strings over $\{0,\ 1\}$ that satisfies ϕ. There exists a natural one-one correspondence between the set of ω-strings over the alphabet $\{0,1\}^n$ and the set of n-tuples of monadic predicates over the set of natural numbers. With a formula $\psi(X_1,\ldots,X_n)$ the set of ω-strings which satisfies ψ through this correspondence can be associated. Such a set of ω-strings is called the ω-language definable by ψ. So, monadic logic can be considered as a formalism for the specification of the behavior (set of runs) of discrete time systems. This logic is accepted as a kind of an universal formalism among decidable formalisms for the specification of discrete time behavior [13].

In this paper we consider interpretations of monadic logic in the continuous time structures of the finitely variable signals and the right continuous signals. In these structures the individual variables range over the non-negative real numbers, the second order variables range over the finitely variable (respectively, right continuous) boolean signals, and $<$ is the standard order relation on the set of real numbers. Similar to the discrete case, monadic logic can be considered as a formalism for the specification of the behavior of continuous time systems. Note that metric properties of reals cannot be specified in this logic.

We provide (see Theorem 1) a characterization of signal languages definable in the monadic logic of order. The result is significant due to the fact that many specification formalisms for reasoning about real time which were considered in the literature can be effectively embedded in $L_2^<$. In [10] we illustrated the expressive power of $L_2^<$ by providing meaning preserving compositional translations from Restricted Duration Calculus [4], Propositional Mean Value Calculus [5] and Temporal Logic of Reals - TLR [3] into the first-order fragment of $L_2^<$.

We apply Theorem 1, for the analysis of a number of fundamental problems. First, as an immediate consequence of our main result we obtain the decidability of $L_2^<$ under finitely variable and right continuous interpretations. These decidability results were obtained in [9, 10] by the method of interpretation [7]. We reduced in [9, 10] these decidability problem to the decidability problem for $L_2^<$ under F_σ interpretation. In F_σ interpretation the monadic predicate variables range over the countable unions of closed subsets of reals. The decidability of $L_2^<$ under under F_σ interpretation was shown to be decidable in [7].

Second, we show that under finitely variable and right continuous interpretations the existential fragment of $L_2^<$ is expressive complete, i.e. for every $L_2^<$ formula ϕ there is an equivalent formula of the form $\exists X_1 \ldots \exists X_n \psi$, where ψ is a first-order monadic formula.

Then we reconsider two fundamental problems of Classical Automata Theory: (1) Automata characterization of the $L_2^<$ definable languages and (2) The uniformization problem.

The classical result of Büchi provides an automata theoretical characterization of the $L_2^<$ definable languages It says that an ω-language is definable by a monadic formula (under the discrete interpretation) iff it is accepted by a finite state automaton.

Let $\phi(X, Y)$ be a formula such that $\forall X \exists Y \phi$ holds. The uniformization problem for ϕ is to find a finite state input-output automaton (transducer) such that the function F computable by the automata satisfies $\forall X.\phi(X, F(X))$. The uniformization problem for the monadic second order theory of order of the structure ω was solved positively by Büchi and Landweber [2].

We check whether these classical results can be extended to continuous time. In [11] automata that accept finitely variable languages were defined. It was announced there that a finitely variability language is definable in monadic logic iff it is accepted by a finite state automata. Here we show that this theorem is a consequence of Theorem 1. In [11] it was announced that the uniformization problem has a positive solution in the continuous time. We found a bug in our proof. In the last section we show that the failure of the uniformization is a consequence of Theorem 1.

The rest of this paper is organized as follows. In Section 2 terminology and notations are fixed. In Section 3 the syntax and semantics of monadic second order logic of order is recalled. Theorem 1 provides reductions between definable ω-languages and definable signal languages; it is stated in Section 4. In Section 5 we collect some simple lemmas and in Section 6 we prove the Theorem 1. Section 7 gives some important corollaries.

2 Terminology and Notations

$\mathbf{N}$ is the set of natural numbers; $\mathbf{R}$ is the set of real numbers, $\mathbf{R}^{\geq \mathbf{0}}$ is the set of non negative reals; $BOOL$ is the set of booleans and Σ is a finite non-empty set. We use $f \circ g$ for the composition of f and g.

A function from $\mathbf{N}$ to Σ is called an ω-string over Σ. A function h from the non-negative reals into a finite set Σ is called a *finitely variable signal* over Σ if there exists an unbounded increasing sequence $\tau_0 = 0 < \tau_1 < \tau_2 \ldots < \tau_n < \ldots$ such that h is constant on every interval (τ_i, τ_{i+1}). Below we will use 'signal' for 'finitely variable signal'. We say that a signal x is *right continuous* at t iff there is $t_1 > t$ such that $x(t) = x(t')$ for all t' which satisfies $t < t' < t_1$. We say that a signal is *right continuous* if it is right continuous at every t.

A set of ω-strings over Σ is called an ω-language over Σ. Similarly, a set of finitely variable (respectively, right continuous) signals over Σ is called a finitely variable (respectively, right continuous) Σ-signal language.

3 Monadic Second Order Theory of Order

3.1 Syntax

The language $L_2^<$ of monadic second order theory of order has individual variables, monadic second order variables, a binary predicate $<$, the usual propositional connectives and first and second order quantifiers $\exists^1$ and $\exists^2$. We use $t,\ v$ for individual variables and $X,\ Y$ for second order variables. Often it will be clear from the context whether a quantifier is the first or the second order; in such cases we will drop the superscript. We use the standard abbreviations, in particular, "$\exists!$" means "there is a unique".

The atomic formulas of $L_2^<$ are formulas of the form: $t < v$ and $X(t)$. The formulas are constructed from atomic formulas by using logical connectives and first and second order quantifiers.

We write $\psi(X, Y, t, v)$ to indicate that the free variables of a formula ψ are among $X,\ Y,\ t,\ v$.

3.2 Semantics

A structure $K = \langle A,\ B, <_K\ \rangle$ for $L_2^<$ consists of a set A partially ordered by $<_K$ and a set B of monadic functions from A into $BOOL$. The letters τ and $x,\ y$ will range over the elements of A and B respectively. We will not distinguish between a subset of A and its characteristic function. The satisfiability relation $K, \tau_1, \dots \tau_m, x_1 \dots x_n \models \psi(t_1, \dots t_m,\ X_1, \dots X_n)$ is defined in a standard way. We sometimes use $K \models \psi(\tau_1, \dots \tau_m,\ x_1, \dots x_n)$ for $K, \tau_1, \dots \tau_m, x_1 \dots x_n \models \psi(t_1, \dots t_m,\ X_1, \dots X_n)$.

We will be interested in the following structures:

1. Structure $\omega = \langle \mathbf{N}, 2^{\mathbf{N}}, <_N\ \rangle$, where $2^{\mathbf{N}}$ is the set of all monadic functions from $\mathbf{N}$ into $BOOL$.

2. The signal structure Sig is defined as $Sig = \langle \mathbf{R}^{\geq \mathbf{0}},\ SIG, <_R\ \rangle$, where SIG is the set of finitely variable boolean signals.

3. The right continuous signal structure $Rsig$ is defined as $Rsig = \langle \mathbf{R}^{\geq \mathbf{0}},\ RSIG, <_R\ \rangle$, where $RSIG$ is the set of right continuous boolean signals.

3.3 Definability

Let $\phi(X)$ be an $L_2^<$ formula and $K = \langle A,\ B, <_K\ \rangle$ be a structure. We say that a set $C \subseteq B$ is definable by $\phi(X)$ if $x \in C$ if and only if $K,\ x \models \phi(X)$.

Example (Interpretations of Formulas).

1. The formula $\forall t_1 \forall t_2.\ t_1 < t_2 \wedge (\neg \exists t_3.\ t_1 < t_3 < t_2) \rightarrow (X(t_1) \leftrightarrow \neg X(t_2))$ defines the ω-language $\{(01)^\omega,\ (10)^\omega\}$ in the structure ω and defines the set of all signals in the signal and right continuous signal structures.

2. The formula $\exists Y.\ \exists t'.Y(t') \wedge (\forall t.\ X(t) \rightarrow Y(t)) \wedge (\forall t_1 \forall t_2.t_1 < t_2 \wedge Y(t_1) \wedge Y(t_2) \rightarrow \exists t_3.t_1 < t_3 < t_2 \wedge \neg Y(t_3))$ defines in the structure ω the set of strings in which between any two occurrences of 1 there is an occurrence of 0. In the signal structure the above formula defines the set of signals that receive value 1 only at isolated points. The formula defines the empty language under the right continuous signal interpretation.

In the above examples, all formulas have one free second order variable and they define languages over the alphabet $\{0,\ 1\}$. A formula $\psi(X_1, \ldots X_n)$ with n free second order variables defines a language over the alphabet $\{0,\ 1\}^n$.

We say that an ω-languages (finitely variable or right continuous signal language) is definable if it is definable by a monadic formula in the structure ω (respectively in the structure Sig or $Rsig$).

4 Characterization of Definable Signal Languages

Recall that a function x from the non-negative reals into a finite set Σ is called a *finitely variable signal* over Σ if there exists an unbounded increasing sequence $\tau_0 = 0 < \tau_1 < \tau_2 \ldots < \tau_n < \ldots$ such that x is constant on every interval (τ_i, τ_{i+1}). In this case there exists an ω-string $\langle a_0, b_0\rangle\langle a_1, b_1\rangle \ldots \langle a_n, b_n\rangle \ldots$ over alphabet $\Sigma \times \Sigma$ such that $x(\tau_i) = a_i$ $x(\tau) = b_i$ for $\tau \in (\tau_i,\ \tau_{i+1})$; this ω-string is said to *represent* a finite variable signal x. We denote by $FV(s)$ the set of finitely variable signals represented by an ω-string s. For an ω-language L we use $FV(L)$ for $\cup_{s\in L} FV(s)$. Similarly, an ω-string $a_0a_1 \ldots a_n \ldots$ over the alphabet Σ represents a right continuous signal x if there is an unbounded increasing sequence $0 = \tau_0 < \tau_1 < \ldots < \tau_i < \ldots$ of reals such that $x(\tau) = a_i$ for $\tau \in [\tau_i,\ \tau_{i+1})$. It is clear that every right continuous signal over Σ is represented by an ω-string over Σ. We denote by $RC(s)$ the set of right continuous signals represented by an ω-string s. For an ω-language L we use $RC(L)$ for $\cup_{s\in L} RC(s)$.

Theorem 1

1. *A finitely variable signal language S is definable if and only if there is a definable ω-language L such that $S = FV(L)$.*

2. *A right continuous signal language S is definable if and only if there is a definable ω-language L such that $S = RC(L)$.*

Our proof of Theorem 1 is constructive. In the proof of the if direction of Theorem 1(2) we will provide a compositional mapping $Tr :\ L_2^< \rightarrow L_2^<$ such that $Tr(\phi)$ defines the right continuous signal language $RC(L)$, where L is an ω-language defined by ϕ. From the proof of the only-if direction of Theorem 1(2), one can also extract an effective mapping $Tr' :\ L_2^< \rightarrow L_2^<$ such that the right continuous signal language defined by ϕ is equal to $RC(L)$, where L is . the ω-language defined by $Tr'(\phi)$. However, Tr' is not compositional.

It is not difficult to show that there is no n such that the length of $Tr'(\phi)$ is bounded by $exp_n(|\phi|)$, where $exp_m(k)$ is the m-times iterated exponential function (e.g. $exp_2(k) = 2^{2^k}$). We do not know whether there exists a more efficient translation. Similar remarks hold for the proof of Theorem 1(1).

A natural question is whether Theorem 1 holds if we replace "definable" by "definable in first-order monadic logic of order". The proof method used in this paper does not allow to establish directly this result. However, the question has a positive answer. The proof is based on the Shelah's compositional method and will be given somewhere else.

In the next section some preliminary lemmas are collected. The proof of Theorem 1 is given in Section 6.

5 Representation of signals by ω-strings

In this section some basic notions and lemmas about the representation of signals by ω-languages are collected. The proofs of some lemmas are straightforward and we omit them.

Lemma 2 *Suppose that an ω-string $\langle a_0, b_0\rangle\langle a_1, b_1\rangle \dots \langle a_n, b_n\rangle \dots$ represents a finitely variable signal x. Then x is right continuous if and only if $a_i = b_i$ for all i.*

Lemma 3 *Let ρ be an order preserving bijection from $\mathbf{R}^{\geq 0}$ to $\mathbf{R}^{\geq 0}$. An ω-string s represents x if and only if s represents $x \circ \rho$.*

5.1 Stuttering and Speed-Independence

ω-strings that represent the same right continuous signal are said to be *stuttering* equivalent. We will use $\simeq_1$ for stuttering equivalence. It is easy to see that the stuttering equivalence on ω-strings is the smallest equivalence such that $a_0a_1 \dots a_na_{n+1} \dots$ is equivalent to $a_0a_1 \dots a_na_na_{n+1} \dots$.

ω-strings over an alphabet $\Sigma \times \Sigma$ that represent the same finitely variable language are said to be $\simeq_2$-equivalent. It is easy to see that the $\simeq_2$-equivalence on ω-strings over alphabet $\Sigma \times \Sigma$ is the smallest equivalence such that $\langle a_0, b_0\rangle\langle a_1, b_1\rangle \dots \langle a_n, b_n\rangle\langle a_{n+1}, b_{n+1}\rangle \dots$ is equivalent to $\langle a_0, b_0\rangle\langle a_1, b_1\rangle \dots \langle a_n, b_n\rangle\langle b_n, b_n\rangle\langle a_{n+1}, b_{n+1}\rangle \dots$.

An ω-language L is *stuttering closed* if whenever $s \in L$ and $s' \simeq_1 s$ then $s' \in L$. We use $Stut_1(L)$ for the stuttering closure of L, i.e. for the ω-language $\{s : s \simeq_1 s' \wedge s' \in L\}$. $Stut_2$-closure of the languages over $\Sigma \times \Sigma$ is defined similarly.

A finitely variable (right continuous) signal language L is speed-independent if for every order preserving bijection ρ, $x \in L$ iff $x \circ \rho \in L$. We use $SI(L)$ for speed-independent closure of L, i.e. for the language $\{x' \circ \rho : x \in L \text{ and } \rho \text{ is an order preserving bijection}\}$. Recall that $FV(L)$ (respectively $RC(L)$)

denotes the set of finitely variable (respectively right continuous) signals represented by the ω-strings of L. It is clear that $FV(L)$ and $RC(L)$ are speed-independent.

We say that an ω-language L *represents* a finitely variable (right continuous) signal language S if for every $x \in S$ there is $s \in L$ that represents x and for every $s \in L$ there is $x \in S$ represented by s.

The following is straightforward.

Lemma 4 *1. If $RC(s_1) \cap RC(s_2) \neq \emptyset$ then $s_1 \simeq_1 s_2$.*

2. if $s_1 \simeq_1 s_2$ then $RC(s_1) = RC(s_2)$.

3. If $FV(s_1) \cap FV(s_2) \neq \emptyset$ then $s_1 \simeq_2 s_2$.

4. if $s_1 \simeq_2 s_2$ then $FV(s_1) = FV(s_2)$.

5. A finite variable (right continuous) signal language S is speed-independent iff $S = FV(L)$ (respectively $S = RC(L)$) for some ω-language L.

6. An ω-language L represents a finitely variable (respectively, right continuous) signal language S iff $Stut_2(L)$ (respectively, $Stut_1(L)$) represents S.

7. An ω-language L represents S iff L represents $SI(S)$.

8. FV induces bijection between the set of $Stut_2$-closed ω-languages and the set of speed-independent finitely variable languages.

9. RC induces bijection between the set of $Stut_1$-closed ω-languages and the set of speed-independent right continuous languages.

Lemma 5 *Every definable finitely variable (respectively right continuous) language is speed-independent.*

Proof: Let K be the finitely variable signal structure or the right continuous signal structure. Let $\rho : \mathbf{R}^{\geq \mathbf{0}} \rightarrow \mathbf{R}^{\geq \mathbf{0}}$ be an order preserving bijection. By the structural induction on $L_2^<$ formulas it is easy to show that $K,\ \tau_1, \ldots \tau_n,\ x_1, \ldots x_m \models \phi$ if and only if $K,\ \rho(\tau_1), \ldots \rho(\tau_n),\ x_1 \circ \rho, \ldots x_m \circ \rho \models \phi$. This implies the lemma. $\square$

Lemma 6 *(1) If L is a definable ω-language, then $Stut_1(L)$ is definable. (2) If L is a definable ω-language over an alphabet $\Sigma \times \Sigma$, then $Stut_2(L)$ is definable. Moreover, there exists an algorithm that for every ϕ constructs ϕ' such that the ω-language definable by ϕ' is the stuttering closure of the ω-language definable by ϕ.*

Proof: Lemma 6(1) was proved in [9]. The proof of Lemma 6(2) is almost the same and is sketched below.

Recall [1], that a set L of ω-strings is $L_2^<$ definable iff L is a regular ω-language (see [13, 12] for a survey of automata on infinite objects). Moreover, there exist algorithms for translations between ω-regular expressions and $L_2^<$ formulas (see [13]). Let h be a regular language substitution defined as $h(\langle a,b\rangle) \triangleq \{\langle a,b\rangle\langle b,b\rangle^n : n \geq 0\}$. Observe that $Stut_2(L) = h(h^{-1}(L))$. Hence, the lemma follows from the the fact that regular ω-languages are closed under the regular morphisms and the inverse images of the regular morphisms.

Actually, the proof of this fact gives an algorithm for constructing an ω-regular expression for the image (pre-image) of an ω-language L from an ω-regular expression that defines L and regular expressions that define a morphism.

Hence, there is an algorithm that for every ϕ constructs ϕ' such that the ω-language definable by ϕ' is the stuttering closure of the ω-language definable by ϕ. $\square$

Remark 7 *The complexity of the algorithm extracted from the proof is non-elementary, i.e. there is no n such that for every formula ϕ, the run time of the algorithm on ϕ is bounded by $exp_n(|\phi|)$, where $exp_m(k)$ is m-time iterated exponential function.*

5.2 Set theoretical operations on languages

Let f be a function from a set A into $\Sigma_1 \times \Sigma_2 \times \ldots \Sigma_n$. We use the notation $Proj_{\Sigma_i}(f)$ for the projection of f onto Σ_i; when Σ_i is clear from the context we sometimes will drop the subscript. Similarly for an ω-string s over $(\Sigma_1 \times \Sigma_2) \times (\Sigma_1 \times \Sigma_2)$ we will use $Proj_{\Sigma_1 \times \Sigma_1}(s)$ for the corresponding projection onto $\Sigma_1 \times \Sigma_1$. Projections are extended pointwise to sets, i.e. for a set F of functions we use $Proj_\Sigma(F)$ for the set $\{Proj_\Sigma(f) : f \in F\}$. Below we use Σ^ω, FV_Σ and RC_Σ for the sets of all ω-strings over Σ, finitely variable signals over Σ, and right continuous signals over Σ, respectively.

Lemma 8 *(operations on finitely variable signal languages and stuttering)*

1. *(Union) Let L_1, L_2 be ω-languages over $\Sigma \times \Sigma$ and let S_1, S_2 be finitely variable signal languages over Σ. If L_i represents S_i $(i = 1, 2)$ then $L_1 \cup L_2$ represents $S_1 \cup S_2$.*
2. *(Complementation) Let S be a speed-independent language. If L represents S then the complementation of $Stut_2(L)$ represents the complementation of S.*
3. *(Projection) Let L be an ω-language (over $\Sigma_1 \times \Sigma_2 \times \Sigma_1 \times \Sigma_2$) that represents a finitely variable signal language S (over the alphabet $\Sigma_1 \times \Sigma_2$). Then $Proj_{\Sigma_i \times \Sigma_i}(L)$ represents $Proj_{\Sigma_i}(S)$.*

4. *(Cylindrification) Let L be an ω-language that represents a finitely variable signal language S (over an alphabet Σ_1). Then the language $\{x \in FV_{\Sigma_1 \times \Sigma_2} : Proj_{\Sigma_1}(x) \in S\}$ is represented by $\{s \in (\Sigma_1 \times \Sigma_2 \times \Sigma_1 \times \Sigma_2)^\omega : Proj_{\Sigma_1 \times \Sigma_1}(s) \in Stut_2(L)\}$.*

Lemma 9 *(operations on right continuous signal languages and stuttering)*

1. *(Union) Let L_1, L_2 be ω-languages over Σ and let S_1, S_2 be right continuous signal languages over Σ. If L_i represents S_i ($i = 1, 2$) then $L_1 \cup L_2$ represents $S_1 \cup S_2$.*

2. *(Complementation) Let S be a speed-independent language. If L represents S then the complementation of $Stut_1(L)$ represents the complementation of S.*

3. *(Projection) Let L be an ω-language that represents a right continuous signal language S (over an alphabet $\Sigma_1 \times \Sigma_2$). Then $Proj_{\Sigma_i}(L)$ represents $Proj_{\Sigma_i}(S)$.*

4. *(Cylindrification) Let L be an ω-language that represents a right continuous signal language S (over an alphabet Σ_1). Then the language $\{x \in RC_{\Sigma_1 \times \Sigma_2} : Proj_{\Sigma_1}(x) \in S\}$ is represented by $\{s \in (\Sigma_1 \times \Sigma_2)^\omega : Proj_{\Sigma_1}(s) \in Stut_1(L)\}$.*

6 Proof of Theorem 1

6.1 The if direction

Let $Cont(X, t)$ be the formula $\exists t_1 t_2.\ t_1 < t < t_2 \wedge \forall t'. t_1 < t' < t_2 \rightarrow (X(t) \leftrightarrow X(t'))$. Let $Jump(X, t)$ be defined as $\neg Cont(X, t)$. If x is a finitely variable signal and $\tau \in \mathbf{R}^{\geq \mathbf{0}}$, then $Jump(x, \tau)$ holds under the finitely variable interpretation iff $\tau = 0$ or x is not continuous at τ. Such τ are called jump points of x. Similarly, if x is a right continuous and $\tau \in \mathbf{R}^{\geq \mathbf{0}}$, then $Jump(x, \tau)$ holds under the right continuous interpretation iff $\tau = 0$ or x is not continuous at τ.

Let $\phi(X_1, \ldots X_n)$ be a monadic formula. Let $\psi(X_1, \ldots X_n,\ Scale)$ be the formula obtained from ϕ when the first order quantifiers are relativized to the jump points of $Scale$, i.e., when "$\exists t.\ \ldots$" and "$\forall t.\ \ldots$" are replaced by "$\exists t.\ Jump(Scale, t) \wedge \ldots$" and by "$\forall t.\ Jump(Scale, t) \rightarrow \ldots$" respectively.

The following lemma is immediate

Lemma 10 *Assume that ψ is obtained from ϕ as described above. Let $s^i = a_0^i a_1^i \ldots$ be ω-strings for $i = 1, \ldots n$. Assume that the set $0 = \tau_0 < \tau_1 < \ldots$ of jump points of finitely variable (respectively right continuous) signal scale is infinite. Let x_i be finitely variable (respectively, right continuous) signals such that $x_i(\tau_j) = a_j^i$ for $j \in \mathbf{N}$ and $i = 1, \ldots n$. Then, (1) $\omega \models \phi(s_1, \ldots s_n)$ iff $Sig \models \psi(x_1, \ldots x_n, scale)$; (2) $\omega \models \phi(s_1, \ldots s_n)$ iff $Rsig \models \psi(x_1, \ldots x_n, scale)$.*

Let $Infjump(X)$ be $\forall t \exists t'.t' > t \wedge (X(t) \leftrightarrow \neg X(t'))$. It is cleat that under both the finitely variable and the right continuous interpretations, $Infjump(x)$ holds iff the set of jump points of x is infinite.

Let $\phi(X_1, \dots X_n)$ be a monadic formula and let $\psi(X_1, \dots X_n, Scale)$ be obtained from ϕ as above by relativizing the first order quantifiers to the jump points of $Scale$. Lemma 10 implies that the right continuous signal language definable by

$$\begin{array}{ll} \exists Scale. & Infjump(Scale) \wedge \psi(X_1, \dots, X_n,\ Scale) \wedge \\ \wedge \bigwedge_{i=1}^{i=n} \forall t. & Cont(Scale, t) \rightarrow Cont(X_i, t) \end{array}$$

is equal to $RC(L)$, where L is the ω-language definable by ϕ. This completes the proof of the if direction of Theorem 1(2).

Now, let $Pred(X, t_1, t_2)$ be the formula $t_1 \leq t_2 \wedge Jump(X, t_1) \wedge \forall t.t_1 < t \leq t_2 \rightarrow \neg Jump(X, t)$. If $Pred(x, \tau_1, \tau_2)$, then x is constant in $(\tau_1, \tau_2]$ and τ_1 is a jump point of x.

Let $\phi(X_1^j, \dots X_m^j, X_1^s, \dots X_m^s)$ be a monadic formula. Let the formula ψ be obtained from ϕ by relativizing the first order quantifiers to the jump points of $Scale$. Lemma 10 implies that the finitely variable language definable by

$$\begin{array}{ll} \exists Scale. & Infjump(Scale) \wedge \exists X_1^j \dots X_m^j X_1^s \dots X_m^s . \psi \wedge \\ \wedge \bigwedge_{i=1}^{i=m} \forall t. & Cont(Scale, t) \rightarrow (Cont(X_i^j, t) \wedge Cont(X_i^s, t)) \\ & \wedge\ (Jump(Scale, t) \rightarrow (X_i(t) \leftrightarrow X_i^j(t)) \\ & \wedge (\neg Jump(Scale, t) \rightarrow \exists t'.Pred(Scale, t', t) \wedge (X_i(t) \leftrightarrow X_i^s(t')) \end{array}$$

is equal to $FV(L)$, where L is the ω-language definable by ϕ.

Remark 11 *Note that for every formula ϕ we constructed a formula ϕ' such that the ω-language definable by ϕ represents the right continuous signal language definable by ϕ'. In our construction (1) the length of ϕ' is linear in the length of ϕ; (2) if ϕ has the form $Q_1 X_1 \dots Q_n X_n \psi$, where ψ does not have the second order quantifiers and Q_i are the second order quantifiers, then ϕ' has the form $\exists Scale.Q_1 X_1 \dots Q_n X_n \psi'$, where ψ' does not have the second order quantifiers. Hence, we added one existential second order quantifier; (3) An alternative proof of the if-direction: first, construct an automaton A_ϕ that accepts the ω-language L definable by ϕ and then from A_ϕ construct a monadic formula χ that defines the signal language $RC(L)$. However, the size of χ extracted from this proof is proportional to the size of A_ϕ and is non-elementary in the size of ϕ.*

Similar remarks hold for the translation to formulas interpreted over finitely variable signal structure, however, in this case several existential second order quantifiers are added.

6.2 Proof of the only-if direction of Theorem 1(1)

Let L be an ω-language. Lemma 4(5) implies that $FV(L)$ (respectively $RC(L)$) is the unique speed-independent finitely variable (respectively, right continuous) signal language represented by L. Therefore, by Lemma 5, in order to show the only-if direction of Theorem 1 it is sufficient to prove that if a finitely variable (right continuous) language is definable, then it is representable by a definable ω-language.

It is convenient instead of the finitely variable signal structure to consider the first order structure $M = \langle SIG;\ Sing,\ <,\ \subseteq\rangle$, where SIG is the set of finitely variable signals,

$$M \models Sing(x) \qquad \text{iff } x = \{t\} \text{ for some } t \in \mathbf{R}^{\geq \mathbf{0}},$$

$$M \models x_1 < x_2 \quad \text{iff } x_1 = \{t_1\}, x_2 = \{t_2\} \text{ for some } t_1 < t_2 \in \mathbf{R}^{\geq \mathbf{0}},$$

and $\subseteq$ is interpreted as the usual inclusion relation.

Let $\phi^*(X_1, \ldots, X_n)$ be the formula in the first order language appropriate for M, which is obtained from a monadic formula $\phi(X_1, \ldots X_n)$ by relativizing the first order quantifiers to $Sing$ (i.e., through the replacement of "$\exists^1 t.$" by "$\exists t.\ Sing(t)\wedge$"), and by the replacement of "$X(t)$" by "$t \subseteq X$". It is easy to see that the signal language definable in Sig by ϕ is the same as the signal language definable in M by ϕ^*.

Therefore, to establish the *only-if direction* of Theorem 1 it is sufficient to show

Proposition 12 *A language definable in M is representable by a definable ω-language.*

Proof: Let L_M be the the first order language appropriate for M. The proof proceeds by the structural induction on the L_M formulas. For every L_M formula $\phi(X_1, \ldots X_n)$ we will construct monadic formula $\psi(X_1^j, X_1^s, \ldots X_n^j, X_n^s)$ such that the ω-language definable by ψ represents the signal language definable by ϕ.

Basis. The formula $sing(X^j, X^s) \triangleq \forall t.\ \neg X^s(t) \wedge \exists! t.\ X^j(t)$ corresponds to the atomic formula $Sing(X)$. The formula $sing(X_1^j, X_1^s) \wedge sing(X_1^j, X_1^s) \wedge \exists t_1 t_2.\ t_1 < t_2 \wedge X_1^j(t_1) \wedge X_2^j(t_2)$ corresponds to $X_1 < X_2$. The formula $\forall t.\ X_1^j(t) \rightarrow X_2^j(t) \wedge X_1^s(t) \rightarrow X_2^s(t)$ corresponds to $X_1 \subseteq X_2$.

Inductive Step. The inductive step is immediately obtained from Lemma 8 and Lemma 6. Indeed, negation corresponds to the complementation, the existential quantifier corresponds to the projection and disjunction is easily expressible from union and cylindrification. $\square$

6.3 Proof of the only-if direction of Theorem 1(2)

The proof is obtained by the method of interpretation [7] as follows. Let $\phi(X_1, \ldots X_n)$ be a monadic formula. First, we construct a monadic formula

$\phi^*(X_1, \ldots X_n)$ such that the language definable by ϕ under the finitely variable interpretation coincides with the language definable by ϕ^* under the right continuous interpretation.

Let $rsignal(X)$ be the formula $\forall t \exists t'.\ t' > t \wedge \forall t''.t < t'' < t' \rightarrow X(t) = X(t'')$. It is clear that a finitely variable signal satisfies $rsignal(X)$ iff it is right continuous.

Let ϕ' be obtained from ϕ by relativizing all the second order quantifiers to right continuous signals, i.e. by replacing "$\forall X.\ \ldots$" (respectively "$\exists X.\ \ldots$") by "$\forall X.\ rsignal(X) \rightarrow \ldots$" (respectively, "$\exists X.\ rsignal(X) \wedge \ldots$"). It is easy to see that a right continuous signal satisfies ϕ under right continuous interpretation iff it satisfies ϕ' under finitely variable interpretation. Hence, the required formula $\phi^*(X_1, \ldots X_n)$ can be defined as $rsignal(X_1) \wedge rsignal(X_2) \wedge \ldots \wedge rsignal(X_n) \wedge \phi'$.

Now, by Theorem 1(1), there exists a $\psi(X_1^j, \ldots X_n^j, X_1^s \ldots X_n^s)$ such that the ω-language definable by ψ represents the language definable by ϕ^* under the finitely variable interpretation. Lemma 2 and the fact that the language definable by ϕ under the right continuous interpretation is the same as the right continuous language definable by ϕ^* under the finitely variable interpretation, imply that the ω-language definable by $\exists X_1^j \ldots X_n^j.\ \psi$ represents the language definable by ϕ under the right continuous interpretation.

7 Fundamental Corollaries

In this section we re-examine four fundamental theorems that hold in the structure ω. Three of these theorems still hold in the finitely variable and the right continuous structures. Their proofs are easily derivable from Theorem 1. In [11] we announced that the uniformization problem has a positive solution in the structures *Rsig* and *Sig*. We found a bug in our proof, and here we will show that in contrast to the discrete case of ω-structure, the uniformization fails in the *Rsig* and *Sig* structures.

7.1 Decidability

The satisfiability problem of the monadic second order theory of the structure ω is decidable (Büchi [1]). As a consequence of the Büchi theorem and the effectiveness of the proof of Theorem 1 we obtain a new proof of

Theorem 13 *The monadic second order theory of the right continuous structure is decidable. The monadic second order theory of the finitely variable structure is decidable.*

Note that the proofs of Theorem 13 given in [9, 10] is obtained by interpreting the monadic theories of the right continuous and finitely variable signal structures in the monadic theory of two successors. No characterizations of definable signal languages can be extracted from these proofs..

7.2 Completeness of the Existential Fragment of Monadic Logic

An existential monadic formula is a formula of the form $\exists X_1 \dots \exists X_n.\psi$, where ψ does not contain the second order quantifiers. It is well known that every monadic formula ϕ is equivalent (in the structure ω) to an existential formula χ. Moreover, χ can be constructed effectively from ϕ. This together with Theorem 1 and Remark 11 imply

Theorem 14 *For every monadic formula ϕ there exists an existential monadic formula ψ such that ϕ is equivalent to ψ in $Rsig$. Moreover, ψ can be constructed effectively from ϕ.*

Theorem 15 *For every monadic formula ϕ there exists an existential monadic formula ψ such that ϕ is equivalent to ψ in Sig. Moreover, ψ can be constructed effectively from ϕ.*

In the F_σ interpretation of monadic logic, the monadic predicates range over F_σ subsets of the real line. The decidability of the monadic logic under F_σ interpretation was established by Rabin [7].

As far as we know the following is an open problem

Open Question: Is the existential fragment of monadic logic complete for F_σ interpretation?

Note that the existential fragment is in the first level of the alternation hierarchy.

Open Question: Does the alternation hierarchy collapse for F_σ interpretation?

7.3 Failure of Uniformization

The uniformization problem for a theory Th in a language $\mathcal{L}$ can be formulated as follows [6]: Suppose $Th \vdash \forall \vec{X} \exists . \vec{Y} . \phi(\vec{X}, \vec{Y})$, where ϕ is an $\mathcal{L}$-formula and $\vec{X}$, $\vec{Y}$ are tuples of variables. Is there another formula ϕ^* such that

$$Th \vdash \forall \vec{X} \forall \vec{Y} . \phi^*(\vec{X}, \vec{Y}) \rightarrow \phi(\vec{X}, \vec{Y}) \text{ and } Th \vdash \forall \vec{X} \exists ! \vec{Y} . \phi^*(\vec{X}, \vec{Y}).$$

Here $\exists !$ means "there is a unique". Hence, ϕ^* defines the graph of a function which lies inside the set definable by ϕ.

The uniformization problem for the monadic second order theory of order of the structure ω was solved positively by Büchi and Landweber [2]. Below we show that the uniformization fails in both the finitely variable and the right continuous signal structures.

First observe that if $x = x \circ \rho$ for every order preserving bijection ρ on $\mathbf{R}^{\geq 0}$, then x is constant on the positive reals. Recall that the languages definable in $Rsig$ and in Sig are speed-independent (see Lemma 5). Therefore,

Lemma 16 *If a singleton language $\{x\}$ is definable in Rsig or in Sig, then x is constant on the positive reals.*

Remark 17 *(Contrast with a discrete case) Note that a singleton ω-language $\{x\}$ is definable in the structure ω iff x is quasiperiodic, i.e., $x = uv^{\omega}$.*

As a consequence we have

Theorem 18 *The uniformization fails for the monadic second order theory of order of the finitely variable structure. The uniformization fails for the monadic second order theory of order of the right continuous structure.*

Proof: Let $\phi(Y)$ be the formula $\forall t \exists t'.\ t' > t \wedge (Y(t) \leftrightarrow \neg Y(t'))$.

It is clear that $Sig \models \exists Y.\phi(Y)$ and that if $Sig \models \phi(y)$ then y changes infinitely often.

Assume that $Sig \models \exists! Y.\ \phi^*(Y)$. Then Lemma 16 implies that the unique y that satisfies $\phi^*(Y)$ is constant on the positive reals. Therefore, it cannot satisfy $\phi(Y)$. To sum up, there is no $\phi^*(Y)$ such that

$$Sig \models \forall Y.\phi^*(Y) \rightarrow \phi(Y) \text{ and } Sig \models \exists! Y.\phi^*(Y).$$

Hence, the uniformization fails for the monadic second order theory of order of the finitely variable structure. The proof for the right continuous structure signals is the same. $\square$

7.4 Characterization of definable Languages by Automata

In this section we provide an automata theoretical characterization of the finitely variable and the right continuous signal languages definable in monadic logic. The characterization is a simple consequence of Theorem 1 and the automata theoretical characterization of ω-languages definable in monadic logic.

7.4.1 Syntax

A Labeled Transition System T is a triple $\langle Q,\ \Sigma,\ \rightarrow\rangle$ that consists of a set Q of states, a finite alphabet Σ of actions and a transition relation $\rightarrow$ which is a subset of $Q \times \Sigma \times Q$; we write $q \xrightarrow{a} q'$ if $\langle q,\ a,\ q'\rangle \in\rightarrow$; If Q is finite we say that the LTS is finite;

Sometimes the alphabet Σ of T will be the Cartesian product $\Sigma_1 \times \Sigma_2$ of other alphabets; in such a case we will write $q \xrightarrow{a,b} q'$ for the transition from q to q' labeled by the pair $(a,\ b)$.

An **automaton** $\mathcal{A}$ over Σ is a triple $\langle T,\ INIT(\mathcal{A}),\ FAIR(\mathcal{A})\rangle$, where $T = \langle Q,\ \Sigma,\ \rightarrow\rangle$ is an LTS over the alphabet Σ; $INIT(\mathcal{A}) \subseteq Q$ - the initial states of $\mathcal{A}$. and $FAIR(\mathcal{A})$ - a collection of fairness conditions (subsets of Q).

7.4.2 Semantics

A run of an automaton $\mathcal{A}$ is an ω-sequence $q_0 a_0 q_1 a_1 \ldots$ such that $q_i \stackrel{a_i}{\to} q_{i+1}$ for all i. Such a run meets the initial conditions if $q_0 \in INIT(\mathcal{A})$. A run meets the fairness conditions if the set of states that occur in the run infinitely many times is a member of $FAIR(\mathcal{A})$.

An ω-string $a_0, a_1 \ldots$ over Σ is accepted by $\mathcal{A}$ if there is a run $q_0 a_0 q_1 a_1 \ldots$ that meets the initial and fairness conditions of $\mathcal{A}$. The ω-language *accepted* by $\mathcal{A}$ is the set of all ω-strings acceptable by $\mathcal{A}$.

Theorem 19 *(Büchi [1]) An ω-language is acceptable by a finite state automaton iff it is definable by a monadic formula.*

7.4.3 Automata as acceptors of signal languages

A right continuous signal x over Σ is accepted by an automaton $\mathcal{A}$ if there are an ω-string $a_0 a_1 \ldots a_n \ldots$ over alphabet Σ acceptable by $\mathcal{A}$ and an unbounded increasing sequence $0 = \tau_0 < \tau_1 < \ldots < \tau_i < \ldots$ of reals such that $x(\tau) = a_i$ for $\tau \in [\tau_i,\ \tau_{i+1})$.

A finitely variable signal x over Σ is accepted by an automaton $\mathcal{A}$ if $\mathcal{A}$ is an automaton over the alphabet $\Sigma \times \Sigma$ and there are an ω-string $\langle a_0, b_0\rangle\langle a_1, b_1\rangle \ldots \langle a_n, b_n\rangle \ldots$ acceptable by $\mathcal{A}$ and an unbounded increasing sequence $0 = \tau_0 < \tau_1 < \ldots < \tau_i < \ldots$ of reals such that $x(\tau_i) = a_i$ and $x(\tau) = b_i$ for $\tau \in (\tau_i,\ \tau_{i+1})$.

A version of the next theorem was announced in [11] and it is immediately derived from Theorem 1 and Theorem 19.

Theorem 20 *A finitely variable (respectively, right continuous) signal language is acceptable by a finite state automaton if and only if it is definable by a monadic formula under the finitely variable (respectively, right continuous) interpretation.*

Acknowledgements

I would like to thank the anonymous referee for his very helpful suggestions.

References

[1] J. R. Büchi. On a decision method in restricted second order arithmetic In *Proc. International Congress on Logic, Methodology and Philosophy of Science*, E. Nagel at al. eds, Stanford University Press, pp 1-11, 1960.

[2] J. R. Büchi and L. H. Landweber. Solving sequential conditions by finite-state strategies. Transactions of the American Mathematical Society, vol 138 (1969), pp. 295-311.

[3] H. Baringer, R. Kuiper and A. Pnueli. A really abstract concurrent model and its fully abstract semantics. In *Proc. 13th ACM Symp. on Princ. of Prog. Lang*, pp. 173-183, 1986.

[4] Zhou Chaochen, M. Hansen and P. Sestoft. Decidability and undecidablity results for Duration Calculus. In *STACS'93,* vol 665 of LNCS, pp. 58-68, 1993.

[5] Zhou Chaochen and L. Xiaoshan. A mean value calculus of duration. In *A Classical Mind: Essays in Honor of C. A. R. Hoare,* pp. 431-451, Prentice Hall, 1994.

[6] Y. Gurevich and S. Shelah. Rabin's uniformization problem. The Journal of Symbolic Logic, vol 48 (1983), pp. 1105-1119.

[7] M. O. Rabin. Decidability of second order theories and automata on infinite trees. In *Trans. Amer. Math. Soc.*,**141**,pp 1-35, 1969.

[8] M. O. Rabin. Decidable theories. In J. Barwise editor *Handbook of Mathematical Logic*, North-Holland, 1977.

[9] A. Rabinovich. On translation of temporal logic of actions into monadic second order logic. Theoretical Computer Science 193 (1998), pp. 197-214.

[10] A. Rabinovich. On the Decidability of Continuous Time Specification Formalisms. Journal of Logic and Computation vol. 8, No5, pp669-678, 1998.

[11] A. Rabinovich and B. A. Trakhtenbrot. From Finite Automata toward Hybrid Systems. In Proceedings of Fundamentals of Computation Theory 1997, vol. 1279 of LNCS, pp. 411-422.

[12] W. Thomas. Automata on Infinite Objects. In J. van Leeuwen editor *Handbook of Theoretical Computer Science*, The MIT Press, 1990.

[13] B. A. Trakhtenbrot and Y. M. Barzdin. Finite Automata, North Holland Amsterdam, 1973.