Logical Methods in Computer Science
Volume 21, Issue 3, 2025, pp. 8:1–8:26
https://lmcs.episciences.org/

Submitted Apr. 09, 2024
Published Jul. 23, 2025

THE CHURCH SYNTHESIS PROBLEM OVER CONTINUOUS TIME

ALEXANDER RABINOVICH AND DANIEL FATTAL

Tel-Aviv University
e-mail address: rabinoa@tauex.tau.ac.il, danielf3141@gmail.com

Abstract. Church's Problem asks for the construction of a procedure which, given a logical specification $\varphi(I, O)$ between input ω-strings I and output ω-strings O, determines whether there exists an operator F that implements the specification in the sense that $\varphi(I, F(I))$ holds for all inputs I. Büchi and Landweber provided a procedure to solve Church's problem for the specifications in Monadic Second-Order logic and operators computable by finite-state automata. We investigate a generalization of the Church synthesis problem to the continuous time domain of the non-negative reals. We show that in the continuous time domain there are phenomena which are very different from the canonical discrete time domain of the natural numbers.

1. Introduction

1.1. **Church Synthesis Problem.** Let *Spec* be a specification language and *Pr* be an implementation language. The following problem is known as the Synthesis problem.

Synthesis problem
Input: A specification $S(I, O) \in Spec$.
Question: Is there a program $P \in Pr$ which implements it, i.e., $\forall I.\ S(I, P(I))$?

We first discuss the **classical Church synthesis problem over discrete time** $\omega = (\mathbb{N}, <)$, and then explain its generalization to continuous time.

The specification language for the Church Synthesis problem is the Monadic Second-Order Logic of Order ($MSO[<]$). Monadic second-order logic is the extension of first-order logic by monadic predicates and quantification over them. We use lower case letters $x, y, z, \dots$ to denote first-order variables and upper case letters $X, Y, Z, \dots$ to denote second-order variables. The atomic formulas of $MSO[<]$ are $x < y$ and $X(y)$. The formulas are constructed from the atomic formulas by boolean connectives and the first-order and the second-order quantifiers. A formula $\varphi(x_0, \dots, x_{k-1}, X_0, \dots, X_{l-1})$ is a formula with free variables among $x_0, \dots, x_{k-1}, X_0, \dots, X_{l-1}$.

In an $MSO[<]$ formula $\varphi(X, Y)$, free variables X and Y range over monadic predicates over $\mathbb{N}$. Since each such monadic predicate can be identified with its characteristic ω-string,

Supported in part by Len Blavatnik and the Blavatnik Family foundation. D. Fattal was supported in part by the Israel Science Foundation (grant number 5166651).

DOI:10.46298/LMCS-21(3:8)2025
© A. Rabinovich and D. Fattal
Creative Commons

φ defines a binary relation on ω-strings. A function F from ω-strings to ω-strings is definable by $\varphi(X,Y)$ iff $\forall X \forall Y (Y = F(X) \leftrightarrow \varphi(X,Y))$.

The *implementation* languages defines functions from ω-strings to ω-strings. Such functions are called operators. A machine that computes an operator at every moment $t \in \mathbb{N}$ reads an input symbol $X(t) \in \{0,1\}$, updates its internal state, and produces an output symbol $Y(t) \in \{0,1\}$. Hence, the output $Y(t)$ produced at t depends only on inputs symbols $X(0), X(1), \dots, X(t)$. Such operators are called causal operators (C-operators); if the output $Y(t)$ produced at t depends only on inputs symbols $X(0), \dots, X(t-1)$, the corresponding operator is called a strongly causal (SC-operator).

Another property of interest is that the machine computing F is finite-state. In light of Büchi's proof [Büc60] of the expressive equivalence of $MSO[<]$ and finite automata, F is finite-state if and only if it is $MSO[<]$-definable.

The following problem is known as the Church Synthesis problem over the discrete time of naturals $\omega := (\mathbb{N}, <)$.

Church Synthesis problem
Input: an $MSO[<]$ formula $\Psi(X,Y)$.
Question: Is there a C-operator F such that $\forall X \Psi(X, F(X))$ holds over $\omega := (\mathbb{N}, <)$?

Büchi and Landweber [BL69] proved that the Church synthesis problem is computable. Their main theorem can be stated as:

Theorem 1.1. *Let $\Psi(X,Y)$ be an $MSO[<]$ formula.*

(1) *Determinacy: exactly one of the following holds for Ψ:*
 (a) *There is a C-operator F such that $\omega \models \forall X.\ \Psi(X, F(X))$.*
 (b) *There is a SC-operator G such that $\omega \models \forall Y.\ \neg\Psi(G(Y), Y)$.*
(2) *Decidability: it is decidable whether 1 (a) or 1 (b) holds.*
(3) *Definability:*
 (a) *If 1 (a) holds, then there is an $MSO[<]$ formula U that defines a C-operator which implements Ψ.*
 (b) *If 1 (b) holds, then there is an $MSO[<]$ formula U that defines a SC-operator G such that $\omega \models \forall Y.\ \neg\Psi(G(Y), Y)$.*
(4) *Computability: There is an algorithm such that for each $MSO[<]$ formula $\Psi(X,Y)$:*
 (a) *If 1 (a) holds, constructs an $MSO[<]$ formula $\varphi(X,Y)$ that defines F.*
 (b) *If 1 (b) holds, constructs an $MSO[<]$ formula $\varphi(X,Y)$ that defines G.*

Recall that a formula $U(X,Y)$ defines an operator F if $\forall X \forall Y (Y = F(X) \leftrightarrow U(X,Y))$. In the above theorem we use "C-operator definable by an $MSO[<]$ formula" instead of an equivalent "operator computable by a finite state transducer (automaton)." This allows us to lift this theorem to continuous time where finite state transducers were not defined.

The proof of Theorem 1.1 is based on the fundamental connections between Logic, Games and Automata.

Classical Automata theory results show an equivalence between three fundamental formalisms: logical nets, automata and $MSO[<]$ logic. Trakhtenbrot [Tra95] suggested a program to extend the classical results from discrete time to other time domains.

In this work we investigate the Church synthesis problem over continuous time.

$MSO[<]$ has a natural interpretation over the non-negative reals $(\mathcal{R}^{\geq 0}, <)$.

Shelah [She75] proved that $MSO[<]$ is undecidable over the reals if we allow quantification over arbitrary predicates. In Computer Science, however, it is natural to restrict predicates to finitely variable (non-Zeno) predicates.

A signal is a function f from $\mathcal{R}^{\geq 0}$ into a finite set Σ such that f satisfies the finite variability (non-Zeno) property: there exists an unbounded increasing sequence $\tau_0 = 0 < \tau_1 < \cdots$ such that f is constant on every open interval (τ_i, τ_{i+1}). In other words, f is finitely variable if it has finitely many discontinuities in any bounded sub-interval of $\mathcal{R}^{\geq 0}$.

We denote the structure for $MSO[<]$ over $(\mathcal{R}^{\geq 0}, <)$ with the finite variability predicates as the interpretation for the monadic variables, by $FVsig$; $MSO[<]$ over $FVsig$ is decidable. The C-operators and SC-operators from signals to signals are defined naturally: F is a C (respectively, SC) operator if the output $F(X)$ at $t \in \mathcal{R}^{\geq 0}$ depends only on X in the interval $[0, t]$ (respectively, $[0, t)$). We investigate the following synthesis problem:

Church Synthesis problem for continuous time
Input: an $MSO[<]$ formula $\Psi(X, Y)$.
Question: Is there a C-operator F such that $\forall X.\ \Psi(X, F(X))$ holds in $FVsig$?

1.2. **Results.** It turns out that the discrete case and the case for continuous time are very different. Here are our main results.

Theorem 1.2 (Indeterminacy)**.** *The synthesis problem for continuous time is indeterminate. There exists an $MSO[<]$ formula $\Psi(X, Y)$ such that:*
(1) *There is no C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*
(2) *There is no SC-operator G such that $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.*

Theorem 1.3 (Dichotomy Fails)**.** *There exists an $MSO[<]$ formula $\Psi(X, Y)$ such that:*
(1) *There is a C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*
(2) *There is a SC-operator G such that $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.*

Theorem 1.4 (Undefinability)**.** *There exists an $MSO[<]$ formula $\Psi(X, Y)$ such that:*
(1) *There is no $MSO[<]$-definable C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*
(2) *There is a C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*

Theorem 1.4 leads us to two natural questions: (1) Is it decidable whether a formula is implementable by an $MSO[<]$-definable C-operator, and if so, is a formula defining an implementation computable? (2) Is it decidable whether a formula is implementable by some C-operator?
The following theorems answer these questions:

Theorem 1.5 (Computability of Definable Synthesis)**.** *Given an $MSO[<]$ formula $\Psi(X, Y)$, it is decidable whether there exists an $MSO[<]$-definable C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$ and if so, there is an algorithm that constructs an $MSO[<]$ formula that defines F.*

Theorem 1.6 (Decidability of Synthesis)**.** *Given an $MSO[<]$ formula $\Psi(X, Y)$, it is decidable whether there exists a C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*

The proofs of Theorems 1.2-1.5 are not difficult. The last theorem is our main result.

1.3. **Related Works.** As far as we know, the Church synthesis problem over continuous time was considered only in [JORW11]. This paper considered only the *right continuous* signals. A signal f is right continuous if there is an unbounded increasing sequence $\tau_0 = 0 < \tau_1 < \cdots$ such that f is constant on every interval $[\tau_i, \tau_{i+1})$. Note that there is no $MSO[<]$ -definable non-constant SC-operator from the right continuous signals to the right continuous signals. Computability of Definable Synthesis (like Theorem 1.5) was proved there. In [JORW11] we also considered $MSO[<, +1]$ - the extension of $MSO[<]$ by a metrical plus one function. This is a very strong logic and it is undecidable over the reals; yet, it is decidable over any bounded subinterval $[0, n]$ of the reals for every $n \in \mathbb{N}$. We proved Computability of Definable Synthesis for $MSO[<, +1]$ for every bounded interval.

Timed automata [AD94] are popular models of real-time systems. Timed automata accept timed words - ω-sequences annotated with real numbers. These timed words are different from signals. Two-player games on timed automata have been considered (see e.g., [CHP11, FLTM02]). These games are concurrent games. Both players suggest a move simultaneously. Due to concurrency, timed automata games are indeterminate. Patricia Bouyer [Bou05] writes that there is a "prolific literature on timed games." Unfortunately, "one paper = one definition of timed games which are parameterized by $\cdots$."

We were unable to deduce our results from the results about timed automata games.

1.4. **Structure of the Paper.** The paper is organized as follows. Section 2 contains preliminaries. Section 3 proves Theorems 1.2 - 1.4. In Section 4 computability of definable synthesis problem is proved. Decidability of the Church synthesis problem is proved in Sections 5-6. Section 5 provides a proof for right continuous signals and contains the main ideas. Section 6 extends the proof to FV-signals. This proof is longer and more technical, but does not require new insights. Section 7 contains the Conclusion. The appendix contains a proof of a lemma.

2. Preliminaries

2.1. Monadic Second-Order Logic of Order.

2.1.1. *Syntax.* The language of Monadic Second-Order Logic of Order ($MSO[<]$) 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. It will be clear from the context whether a quantifier is the first or the second-order, and we will drop the superscript. We use standard abbreviations, in particular, "$\exists!$" means "there is a unique."

The atomic formulas of $MSO[<]$ are formulas of the form: $t < v$ and $X(t)$. 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.

2.1.2. *Semantics.* A structure $K = \langle A, B, <_K \rangle$ for $MSO[<]$ consists of a set A partially ordered by $<_K$ and a set B of monadic functions from A into $\{0, 1\}$. The letters τ, t, s, x, y will range over the elements of A and capital bold letter $\mathbf{X}$,$\mathbf{Y}$,$\mathbf{Z}, \dots$ will range over the elements of B. We will not distinguish between a subset of A and its characteristic function. The satisfiability relation $K, \tau_1, \dots, \tau_m, \mathbf{X}_1 \dots \mathbf{X}_n \models \psi(t_1, \dots, t_m,\ X_1, \dots, X_n)$ is defined in a standard way. We sometimes use $K \models \psi(\tau_1, \dots \tau_m,\ \mathbf{X}_1, \dots, \mathbf{X}_n)$ for $K, \tau_1, \dots, \tau_m, \mathbf{X}_1, \dots, \mathbf{X}_n \models \psi(t_1, \dots, t_m,\ X_1, \dots, X_n)$.

We will be interested in the following structures:

(1) The structure $\omega = \langle \mathbb{N}, 2^{\mathbb{N}}, <_{\mathbb{N}} \rangle$ where $2^{\mathbb{N}}$ is the set of all monadic functions from $\mathbb{N}$ into $\{0, 1\}$.
(2) The signal structure $FVsig$ is defined as $FVsig = \langle \mathcal{R}^{\geq 0}, SIG, <_R \rangle$, where SIG is the set of finitely variable boolean signals.
(3) A signal is right continuous if there exists an unbounded increasing sequence $\tau_0 = 0 < \tau_1 < \cdots$ such that f is constant on every half-open interval $[\tau_i, \tau_{i+1})$. The right continuous signal structure $Rsig$ is defined as $Rsig = \langle \mathcal{R}^{\geq 0}, RSIG, <_R \rangle$ where $RSIG$ is the set of right continuous boolean signals.

Terminology: For a FV (or RC) signal f we say that f is *constant* or *continuous* at τ if there are $\tau_1 < \tau < \tau_2$ such that f is constant in the interval $[\tau_1, \tau_2]$. If f is not constant at τ, we say that it *jumps* at τ or it is *discontinuous* at τ. In particular, every signal jumps at $\tau = 0$.

2.2. **Coding.** Let Δ be a finite set. We can code a function from a set D to Δ by a tuple of unary predicates on D. This type of coding is standard, and we shall use explicit variables which range over such mappings and expressions of the form "$F(u) = d$" (for $d \in \Delta$) in MSO-formulas, rather than their codings.

Formally, for each finite set Δ we have second-order variables $X_1^{\Delta}, X_2^{\Delta}, \dots$ which range over the functions from D to Δ, and atomic formulas $X_i^{\Delta}(u) = d$ for $d \in \Delta$ and a first-order variable u [TB73]. Often, the type of the second-order variables will be clear from the context (or unimportant) and we drop the superscript Δ.

2.3. **Definability.** Let $\phi(X)$ be an $MSO[<]$ formula. We say that a language (set of predicates) L is definable by $\phi(X)$ in a structure K if L is the set of monadic predicates which satisfies $\phi(X)$ in K.

Example (Interpretations of Formulas).

(1) 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 it defines the set of all signals in the signal structures, since every set $X \subseteq \mathbb{R}^{\geq 0}$ satisfies this formula.
(2) 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 FV-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 $\psi(X_1, \dots, X_n)$ with n free second order variables defines a language over the alphabet $\{0,\ 1\}^n$. We say that a ω-language is definable if it is definable by a monadic formula in the structure ω.

Let $\psi(X^{\Sigma_{in}}, Y^{\Sigma_{out}})$ be an $MSO[<]$ formula. We say that ψ defines an operator F if for all X and Y: $\psi(X, Y)$ if and only if $Y = F(X)$.

Definition 2.1 (Speed independent signal language)**.** Let L be a signal language. L is speed independent if for each order preserving bijection $\rho : \mathcal{R}^{\geq 0} \to \mathcal{R}^{\geq 0}$:

$$X \in L \text{ if and only if } \rho(X) \in L.$$

It is clear:

Lemma 2.2. *The $MSO[<]$ definable languages are speed independent.*

Moreover,

Lemma 2.3 (The output signal jumps only if the input signal jumps)**.** *If F is an $MSO[<]$ definable operator and $F(\boldsymbol{X})$ is discontinuous at $t > 0$, then $\boldsymbol{X}$ is discontinuous at t.*

Proof. Assume that F is definable by $\psi(X, Y)$ and $\mathbf{Y} = F(\mathbf{X})$ is discontinuous at $t > 0$. If $\mathbf{X}$ is continuous at t, then there is an interval (a, b) such that t is the only point of discontinuity of $(\mathbf{X},\mathbf{Y})$ in (a, b). Let $\rho : \mathcal{R}^{\geq 0} \to \mathcal{R}^{\geq 0}$ be an order preserving bijection that is the identity map outside of (a, b) and maps t to $t' \neq t$. Now $(\rho(\mathbf{X}), \rho(\mathbf{Y}))$ satisfies ψ. Note that $\rho(\mathbf{X}) = \mathbf{X}$. Hence, by the functionality of F, $\rho(\mathbf{Y}) = \mathbf{Y}$, and this is a contradiction, because the only point of discontinuity of $\mathbf{Y}$ in (a, b) is t, and the only point of discontinuity of $\rho(\mathbf{Y})$ in (a, b) is $t' \neq t$. □

2.4. **Parity Automata.** A deterministic parity automaton is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_{init}, pr)$ that consists of the following components:

- Q is a finite set. The elements of Q are called the states of $\mathcal{A}$.
- Σ is a finite set called the alphabet of $\mathcal{A}$.
- $\delta :\ Q \times \Sigma \to Q$ is the transition function of $\mathcal{A}$.
- $q_{init} \in Q$ is the initial state.
- $pr :\ Q \to \mathbb{N}$ is the priority function.

An input for $\mathcal{A}$ is an ω-sequence $\alpha = \sigma_0\sigma_1 \cdots$ over Σ. The run of $\mathcal{A}$ over α is an infinite sequence $\rho = r_0, r_1, r_2, \cdots$ of states, defined as follows:

- $r_0 := q_{init}$.
- $r_{i+1} := \delta(r_i, \sigma_i)$ for $i \geq 0$.

Acceptance condition: Let $Inf(\rho)$ be the set of states that occur infinitely often in ρ. $\mathcal{A}$ accepts exactly those runs ρ for which $\min(pr(q) \mid q \in Inf(\rho))$ is even. An ω-string α is accepted if the run over α is accepted. An ω-language of $\mathcal{A}$ is the set of ω-strings accepted by $\mathcal{A}$.

The following fundamental result is due to Büchi:

Theorem 2.4. *An ω-language is definable by an $MSO[<]$ formula if and only if it is accepted by a deterministic parity automaton.*

3. Proofs of Theorems 1.2 - 1.4

3.1. **Indeterminacy.**

Theorem 1.2 (Indeterminacy)**.** *The synthesis problem for continuous time is indeterminate. There exists an MSO*$[<]$ *formula* $\Psi(X,Y)$ *such that:*

(1) *There is no C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
(2) *There is no SC-operator* G *such that* $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.

To prove that the synthesis problem is indeterminate, for each $x \in \mathcal{R}^{\geq 0}$ consider the following signal

$$\delta_x(t) := \begin{cases} 1, & \text{if } t = x \\ 0 & \text{else} \end{cases}$$

Recall that a signal X jumps at t if it is not continuous at t. Let

$$\Psi(X,Y) := \exists t > 0 \text{ such that } X \text{ is constant on } (0,t] \text{ and } Y \text{ jumps at } t$$

Proof of Theorem 1.2.

(1) Assume that there exists a C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.
Notice that for each $x > 0$, since δ_x jumps at x and is constant on $(0,x)$, we have the following equivalence

$$\Psi(\delta_x, F(\delta_x)) \text{ iff } \exists t \in (0,x) \text{ such that } F(\delta_x) \text{ jumps at } t.$$

Therefore, $FVsig \models \Psi(\delta_1, F(\delta_1))$ implies that

$$\textbf{(*)} \quad \exists t \in (0,1) \text{ such that } F(\delta_1) \text{ jumps at } t.$$

By non-Zenoness, there exists a **minimal** $t \in (0,1)$ that satisfies **(*)**. Moreover,
  (a) By causality, since $\delta_1 = \delta_t$ on $[0,t)$ then $F(\delta_1) = F(\delta_t)$ on $[0,t)$.
  (b) By minimality, $F(\delta_1)$ is constant on $(0,t)$. So by (a), both $F(\delta_1), F(\delta_t)$ are constant on $(0,t)$.
  (c) By $\Psi(\delta_t, F(\delta_t))$, $F(\delta_t)$ jumps at some $x \in (0,t)$. That contradicts (b).

(2) Assume that there exists a SC-operator G such that $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$. By applying logical equivalences we obtain:

$$\neg\Psi(X,Y) \text{ iff } \forall t > 0.\ X \text{ jumps on } (0,t] \text{ or } Y \text{ is continuous at } t.$$

Therefore, for each $x \in \mathcal{R}^{\geq 0}$:
$\neg\Psi(G(\delta_x), \delta_x)$ iff $\forall t > 0$. $G(\delta_x)$ jumps on $(0,t]$ or δ_x is continuous at t, but since δ_x is not continuous at x, we obtain $G(\delta_1)$ jumps on $(0,1]$.
Let a be the first positive jump point of $G(\delta_1)$. Since $\delta_1 = \delta_{a/2}$ on $[0, a/2)$, by strong causality of G, we obtain that $G(\delta_1) = G(\delta_{a/2})$ on $[0, a/2]$. Thus, $G(\delta_{a/2})$ is constant on $(0, a/2]$ and $\delta_{a/2}$ jumps at $a/2$. We obtained a contradiction to $\neg\Psi(G(\delta_{a/2}), \delta_{a/2})$. ☐

3.2. **Failure of dichotomy.**

Theorem 1.3 (Dichotomy Fails)**.** *There exists an MSO*$[<]$ *formula* $\Psi(X, Y)$ *such that:*

(1) *There is a C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
(2) *There is a SC-operator* G *such that* $FVsig \models \forall Y.\ \neg\Psi(G(Y), Y)$.

For the proof of Theorem 1.3, consider the formula Ψ, the C-operator F, and the SC-operator G given below:

$$\Psi(X, Y) := \forall x(X(x) \leftrightarrow Y(x))$$

$$F(X) := X$$

$G(X)(0) := 0$, and for $t > 0$:

$$G(X)(t) := \begin{cases} 1 - a & \text{if } X \text{ has a constant value } a \text{ over } (0, t) \\ 1 & \text{else} \end{cases}$$

It is clear that F is a C-operator that implements Ψ. Below we prove (2).

a **:** G is a strongly-causal operator: suppose $\mathbf{X}_1 = \mathbf{X}_2$ on $(0, t)$. Either both $\mathbf{X}_1, \mathbf{X}_2$ have a jump point in $(0, t)$ or both are constant on $(0, t)$.
In the first case, by Non-Zenoness, there exists $t_0 \in (0, t)$ such that t_0 is the **first** jump point of both $\mathbf{X}_1, \mathbf{X}_2$ in $(0, t)$. Therefore, both $\mathbf{X}_1, \mathbf{X}_2$ have a constant value a over $(0, t_0)$. Thus, by definition

$$\forall t \geq 0.\ G(\mathbf{X}_1)(t) = G(\mathbf{X}_2)(t) = \begin{cases} 1 - a & t \in (0, t_0) \\ 1 & else \end{cases}$$

The proof for the second case is similar.

b **:** G implements $\neg\Psi$: for each signal $\mathbf{Y}$ there exists a non-empty interval such that $\mathbf{Y} \neq G(\mathbf{Y})$ on it. Thus, $\mathbf{Y} \neq G(\mathbf{Y})$.

3.3. **Undefinability.**

Theorem 1.4 (Undefinability)**.** *There exists an MSO*$[<]$ *formula* $\Psi(X, Y)$ *such that:*

(1) *There is no MSO*$[<]$*-definable C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.
(2) *There is a C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$.

Consider the formula $\Psi(X, Y) : = Y$ jumps inside $(0, \infty)$.

It is implemented by $F(X)(t) := \begin{cases} 1, & \text{if } t = 1 \\ 0, & \text{else} \end{cases}$

When X is constant on $[0, \infty)$, Y should jump at $t > 0$, where X is continuous. However, by Lemma 2.3, no *MSO*$[<]$-definable operator implements Ψ. Hence, Ψ is implemented by a C-operator, but it is not implemented by a definable operator.

4. Computability of the Definable Synthesis

In [JORW11] the computability of the definable synthesis problem over the right continuous signals was proved.

Theorem 4.1 (Computability of the Definable Synthesis over the RC Signals)**.** *Given an MSO*$[<]$ *formula* $\Psi(X, Y)$*, it is decidable whether there exists an MSO*$[<]$*-definable C-operator* F *such that* $RC \models \forall X.\ \Psi(X, F(X))$ *and if so, there is an algorithm that constructs an MSO*$[<]$ *formula that defines* F*.*

In this section we extend this theorem to finite variability signals.

Theorem 1.5 (Computability of Definable Synthesis)**.** *Given an MSO*$[<]$ *formula* $\Psi(X, Y)$*, it is decidable whether there exists an MSO*$[<]$*-definable C-operator* F *such that* $FVsig \models \forall X.\ \Psi(X, F(X))$ *and if so, there is an algorithm that constructs an MSO*$[<]$ *formula that defines* F*.*

Our proof is a variation of the proof of Theorem 4.1. In the next subsection we recall a relationship between finite variability signals and ω-strings. In subsection 4.2, we discuss definable operators. Finally, in subsection 4.3, we present an algorithm for the definable synthesis problem.

4.1. FV-signals and ω-strings.

We recall the relationship between finite variability signals and ω-strings.

Let X be a finite variability signal over Σ and let $\widetilde{\tau} := 0 = \tau_0 < \tau_1 < \cdots < \tau_n < \cdots$ be an ω-sequence of reals such that X is constant on every interval (τ_i, τ_{i+1}) and $\lim \tau_i = \infty$. We say that such $\widetilde{\tau}$ is a sample sequence for X. In other words, an unbounded ω-sequence is a sampling sequence for X if it contains all the points of discontinuity of X.

We define:

- $D(X, \widetilde{\tau})$ be an ω-string $(a_0, b_0)(a_1, b_1) \dots (a_i, b_i) \dots$ over $\Sigma \times \Sigma$ defined as $a_i := X(\tau_i)$ and b_i is the value of X in (τ_i, τ_{i+1}). We say that $D(X, \widetilde{\tau})$ **represents** X.
- $D(X) := \{D(X, \widetilde{\tau}) \mid \widetilde{\tau}$ is a sample sequence for $X\}$.
- For a signal language L define $D(L) := \bigcup_{X \in L} D(X)$.

Observe that if an ω-string U represents a signal X and ρ is an order preserving bijection over non-negative reals, then U represents $\rho(X)$.

- For an ω-string $U = (a_0, b_0)(a_1, b_1) \dots (a_i, b_i) \dots$ over $\Sigma \times \Sigma$ and an unbounded ω-sequence $\widetilde{\tau} := 0 = \tau_0 < \tau_1 < \cdots < \tau_n < \cdots$ of reals define a finite variability signal:
  $FV(U, \widetilde{\tau})$ to be a_i at τ_i and b_i in the interval (τ_i, τ_{i+1}).
- $FV(U) := \{FV(U, \widetilde{\tau}) \mid \widetilde{\tau}$ is an unbounded ω-sequence $\}$.
- For an ω-language L define: $FV(L) := \bigcup_{U \in L} FV(U)$.

The following theorem was proved in [Rab02]:

Theorem 4.2. *A finitely variable signal language* S *is MSO*$[<]$*-definable if and only if there is an MSO*$[<]$*-definable* ω*-language* L *such that* $S = FV(L)$*. Moreover, (1) there is an algorithm that from an MSO*$[<]$ *formula* $\varphi(X^{\Sigma})$ *computes an MSO*$[<]$ *formula* $\varphi^D(X^{\Sigma\times\Sigma})$ *such that if* S *is a signal language definable by* φ *then* $D(L)$ *is definable by* φ^D*, and (2) there is an algorithm that from an MSO*$[<]$ *formula* $\varphi(X^{\Sigma\times\Sigma})$ *computes an MSO*$[<]$ *formula* $\varphi^{FV}(X^{\Sigma})$ *such that if* L *is an* ω*-language definable by* φ *then* $FV(L)$ *is definable by* φ^{FV}*.*

Note also that if L is a definable signal language, then $L = FV(D(L))$. However, for a definable ω-language L, it might be the case that $L \subsetneq D(FV(L))$.

Definition 4.3 (Stuttering Equivalence)**.** Two ω-strings are stuttering equivalent if there is a signal X which is represented by both of them. Equivalently, the sets of signals represented by these strings coincide.

Definition 4.4 (Stuttering Free)**.** An ω-string U is stuttering free if there is a signal X such that (1) or (2) holds.

(1) X has infinitely many points of discontinuity and $U = D(X, \widetilde{\tau})$ where $\widetilde{\tau}$ is a sample sequence that contains exactly all points of discontinuity of X, or
(2) X has k points of discontinuity $0 = \tau_0 < \tau_1 < \cdots < \tau_{k-1}$, and $U = D(X, \widetilde{\tau})$, where $\widetilde{\tau} := 0 = \tau_0 < \tau_1 < \cdots < \tau_{k-1} < \tau_k < \cdots$, where τ_i (for $i < k$) are the points of discontinuity of X, and τ_j for $j \geq k$ are arbitrary.

Lemma 4.5.

(1) *For every signal X, all strings in $D(X)$ are stuttering equivalent.*
(2) *Every ω-string V is stuttering equivalent to a unique stuttering free string U.*

Stuttering free ω-strings can be characterized as follows:

Lemma 4.6. *An ω-string U over $\Sigma \times \Sigma$ is stuttering-free if it satisfies the following condition: if i-th letter of U is (a, b) and $i+1$-th letter is (b, b), then for all $j > i$, j-th letter of U is (b, b).*

Hence,

Lemma 4.7 (Stuttering free strings are definable)**.** *There is an $MSO[<]$ formula that defines the set of stuttering free ω-strings over $\Sigma \times \Sigma$.*

4.2. **Definable Operators.** Let $\alpha(X^{\Sigma^2_{in}}, Y^{\Sigma^2_{out}})$ be an $MSO[<]$ formula. It defines an ω-language over $\Sigma^2_{in} \times \Sigma^2_{out}$. To an ω-string over $\Sigma^2_{in} \times \Sigma^2_{out}$ in a natural way corresponds an ω-string over $(\Sigma_{in} \times \Sigma_{out})^2$ and an ω-string over $\Sigma_{in} \times \Sigma_{out}$. Hence, we can consider α as defining an ω-language over $\Sigma_{in} \times \Sigma_{out}$. We say that α defines a C-operator (respectively, SC-operator), if its language is the graph of this operator.

Lemma 4.8. *Let $\alpha(X^{\Sigma^2_{in}}, Y^{\Sigma^2_{out}})$ be a formula such that:*

(1) *If $(X^{\Sigma^2_{in}}, Y^{\Sigma^2_{out}})$ satisfies α, then $X^{\Sigma^2_{in}}$ is stuttering free.*
(2) *For every stuttering free X there is a unique Y such that (X, Y) satisfies α.*

Then, α^{FV} defines an operator on signals. Moreover, if in addition α defines C-operator (respectively, SC-operator) on ω-strings, then α^{FV} defines C-operator (respectively, SC-operator) on FV-signals.

Lemma 4.9. *If there is a definable C-operator that implements $\Psi(X^{\Sigma_{in}}, Y^{\Sigma_{out}})$ over FV-signals, then there is a definable C-operator that implements Ψ^D (over ω).*

Proof. Assume that α defines a C-operator that implements $\Psi(X^{\Sigma_{in}}, Y^{\Sigma_{out}})$. We claim that α^D defines a C-operator that implements Ψ^D (over ω). First, note that α^D implies ψ^D. Now, for every $U \in (\Sigma^2_{in})^\omega$ there is a FV signal X and $\widetilde{\tau}$ such that $U = D(X, \widetilde{\tau})$. Let Y be such that $\alpha(X, Y)$ holds. Since Y changes only when X changes, $\widetilde{\tau}$ is also a sample sequence for Y. Let $V := D(Y, \widetilde{\tau})$. It is clear that $\alpha^D(U, V)$. Therefore, for every U there is V such that $\alpha^D(U, V)$. It remains to show that such V is unique. Indeed, if there are V_1 and V_2 such that both (U, V_1) and (U, V_2) satisfy α^D, then for arbitrary $\widetilde{\tau}$, $FV((U, V_1), \widetilde{\tau})$ and $FV((U, V_2), \widetilde{\tau})$ satisfy α. However, $FV(V_1, \widetilde{\tau}) \neq FV(V_2, \widetilde{\tau})$ and both are α related to $FV(U, \widetilde{\tau})$. This contradicts the fact that α defines an operator on signals.

Causality and strong causality are immediate. □

Recall that Lemma 2.3 states: if F is a definable operator on finite variability signals, then $F(X)$ is not continuous at t only if X is not continuous at t.

Lemma 4.10. *Let $\Psi(X^{\Sigma_{in}}, Y^{\Sigma_{out}})$ be in MSO[<] and let F be an MSO[<] definable C-Operator (or SC-operator).*
Denote $\Psi^*(X^{\Sigma_{in}}, Y^{\Sigma_{out}}) :=$

$$\Psi \wedge \forall t \in (0, \infty) \textit{ if } X \textit{ is continuous at } t, \textit{ then so is } Y \wedge$$
$$\textit{if } X \textit{ is left-continuous at } t, \textit{ then so is } Y.$$

Then

$$F \textit{ implements } \Psi \iff F \textit{ implements } \Psi^*.$$

4.3. **Algorithm.** Now, we are ready to present an algorithm that verifies whether there is a definable C-operator that implements an *MSO*[<] formula over FV-signals, and if so, it computes an *MSO*[<] formula that defines such an operator.

Algorithm

Let $\Psi(X^{\Sigma_{in}}, Y^{\Sigma_{out}})$ be an *MSO*[<] formula and let Ψ^* be defined as in Lemma 4.10.
Use Büchi-Landweber algorithm to decide whether there is a C-operator that implements $(\Psi^*)^D$ over ω.
If the answer is "NO" then there is no *MSO*[<] definable C-operator that implements Ψ.
If α defines a C-operator that implements $(\Psi^*)^D$ over ω, then let

$$\alpha_{sf} := \alpha(X^{\Sigma^2_{in}}, Y^{\Sigma^2_{out}}) \wedge \text{"}X \text{ is stuttering free."}$$

Let $\beta := (\alpha_{sf})^{FV}$. Then, β defines a C-operator that implements Ψ.

Correctness of the algorithm.

If the answer is "NO" the correctness follows from Lemma 4.9 and Lemma 4.10.

If the algorithm returns β, the correctness follows from Lemma 4.8 and Lemma 4.10. □

The algorithm that decides whether there is a SC-definable operator on FV-signals that implements Ψ is almost the same: replace everywhere "C-operator" by "SC-operator."

Remark 4.11 (Non-causal uniformization)**.** A uniformizer for a binary relation $R \subseteq A \times B$ is a maximal subrelation R^* which is the graph of a partial function. Equivalently, let us call the "cross-section at a" for $a \in A$, the set $R_a = \{b \mid (a, b) \in R\}$, then a uniformizer for R is a relation $R^* \subseteq R$ such that for all $a \in A$, if R_a is non-empty, then there exists exactly one b such that $(a, b) \in R^*$.

For every $\mathsf{MSO}[<]$ formula $\Psi(X, Y)$, there is an $\mathsf{MSO}[<]$ formula $\Phi(X, Y)$ such that the (binary) relation definable by $\Phi(X, Y)$ on ω is a uniformizer for the (binary) relation definable by $\Psi(X, Y)$ on ω [LS98, Rab07]. However, over signals, the relation "there is $t > 0$

such that X is continuous at t, but Y is discontinuous at t" is $\mathsf{MSO}[<]$-definable, yet it has no $\mathsf{MSO}[<]$-definable uniformizer.

Arguments analogous to those presented in the proof of Theorem 1.5 demonstrate that it is decidable whether a relation defined by an $\mathsf{MSO}[<]$ formula $\Psi(X, Y)$ on signals has an $\mathsf{MSO}[<]$-definable uniformizer.

Let $\Psi(X, Y)$ be an $\mathsf{MSO}[<]$ formula and let

$$\Psi^\dagger(X, Y) := \Psi(X, Y) \wedge \forall t \in (0, \infty) \text{ if } X \text{ is continuous at } t, \text{ then so is } Y.$$

Then Ψ has an $\mathsf{MSO}[<]$-definable uniformizer iff $\forall X(\exists Y \Psi \rightarrow \exists Y \Psi^\dagger)$ holds in the signal structure. For every Ψ: $\Psi^\dagger$ has an $\mathsf{MSO}[<]$-definable uniformizer which is computable from Ψ.

5. Games and the Church Problem

Our objective is to demonstrate that the Church Synthesis problem is computable for FV signals. In this section, we present a proof of the decidability of Church's synthesis problem for right-continuous signals. This proof encompasses all the main ideas, but is slightly simpler than the proof for FV signals. In the following section, we outline its extension to FV signals.

Recall that a timed ω-word (or timed ω-sequence) over an alphabet Σ is an ω-sequence $(a_0, \tau_0)(a_1, \tau_1) \cdots (a_i, \tau_i) \cdots$, where $a_i \in \Sigma$ and $0 = \tau_0 < \tau_1 < \cdots < \tau_i < \cdots$ is an unbounded ω-sequence of reals. Such a sequence represents a right continuous signal X defined as $X(\tau) := a_i$ for $\tau \in [\tau_i, \tau_{i+1})$. Note that every right continuous signal is represented by a timed sequence.

We start with a reduction of the Church Synthesis problem to a game. Let $\Psi(X, Y)$ be a formula. Consider the following two-player game $\mathcal{G}_\Psi$. During the play, $\mathcal{I}$ constructs an input signal (equivalently, timed word) X and $\mathcal{O}$ constructs an output Y.

Round 0. Set $t_0 := 0$.

(1) $\mathcal{I}$ chooses a value $a_0 \in \Sigma_{in}$ which holds at t_0 and for a while after t_0.
(2) $\mathcal{O}$ responds with a timed ω-word (equivalently, with a signal) Y'_0 on $[t_0, \infty)$.
 Now, $\mathcal{I}$ has two options:
 A) $\mathcal{I}$ accepts. In this case, the play ends. $\mathcal{I}$ has constructed X which has a_0 at all $t \geq 0$, and $\mathcal{O}$ has constructed $Y := Y'_0$ - the signal chosen by $\mathcal{O}$.
 B) $\mathcal{I}$ interrupts at some $t_1 > t_0$ by choosing $a_1 \neq a_0$ which holds at t_1 and for a while after t_1. Signal X_0 constructed by $\mathcal{I}$ is a_0 in $[t_0, t_1)$ and is a_1 at t_1, Y_0 constructed by $\mathcal{O}$ is the restriction of Y'_0 on the interval $[t_0, t_1)$. Now, proceed to step 1 of round 1.

Round $i + 1$.

(1) $\mathcal{O}$ chooses a timed ω-word (equivalently, a signal) Y'_{i+1} on $[t_{i+1}, \infty)$.
(2) Now, $\mathcal{I}$ has two options:
 A) $\mathcal{I}$ accepts. In this case, the play ends. The signal X constructed by $\mathcal{I}$ coincides with X_i on $[0, t_{i+1})$ and equals a_{i+1} on $[t_{i+1}, \infty)$. The signal Y constructed by $\mathcal{O}$ coincides with Y_i on $[0, t_{i+1})$ and with Y'_{i+1} on $[t_{i+1}, \infty)$.
 B) $\mathcal{I}$ interrupts at some $t_{i+2} > t_{i+1}$ by choosing $a_{i+2} \neq a_{i+1}$ for X at t_{i+2} and for a while after it. In this case, X_{i+1} is the signal that has the value a_{i+2} at t_{i+2}, a_{i+1} on $[t_{i+1}, t_{i+2})$ and extends the signal X_i constructed in the previous steps on $[0, t_{i+1})$, and

Y_{i+1} coincides with Y_i on $[0, t_{i+1})$ and coincides with Y'_{i+1} on $[t_{i+1}, t_{i+2})$. Start round $i + 2$.

After ω rounds, if $\lim t_i = \infty$, we define X (respectively, Y) to be the signal that coincides with X_i (respectively, with Y_i) on $[t_i, t_{i+1})$ for every i.

Winning conditions. $\mathcal{O}$ wins the play iff $\Psi(X, Y)$ holds or $\mathcal{I}$ interrupts infinitely often and the duration of the play is finite, i.e., $\lim t_n < \infty$, where t_n is the time of n-th interruption.

Note that in this game, player $\mathcal{I}$ has much more power than player $\mathcal{O}$. Player $\mathcal{O}$ provides a signal untill infinity and $\mathcal{I}$ observes it and can interrupt it at any time moment she chooses. Nevertheless, we have the following lemma.

Lemma 5.1 (Reduction to games)**.** *$\mathcal{O}$ wins $\mathcal{G}_\Psi$ if and only if there is a C-operator that implements Ψ.*

Example 5.2. Assume that Ψ expresses that Y has a discontinuity at some $t > 0$ where X is continuous. Let us describe a winning strategy for $\mathcal{O}$. At the i-th move $\mathcal{O}$ chooses a signal which has value $X(t_i)$ at $[t_i, t_i + \frac{1}{2^i})$ and $\neg X(t_i)$ at $t \geq t_i + \frac{1}{2^i}$. It is clear that this is a winning strategy for $\mathcal{O}$. Indeed, in order to ensure that Ψ fails, at the i-th round, $\mathcal{I}$ should interrupt at t_{i+1} before $t_i + \frac{1}{2^i}$. Therefore, the duration of such a play is less than $\sum 2^{-i} = 2 < \infty$. Hence, $\mathcal{O}$ wins.

Proof of Lemma 5.1.

$\Rightarrow$-direction. Let σ be an $\mathcal{O}$-winning strategy. Consider the play where $\mathcal{I}$ ignores $\mathcal{O}$'s moves and constructs an input signal X. Let Y be the signal constructed by $\mathcal{O}$ in this play, This defines a C-operator, and $\Psi(X, Y)$ holds.

$\Leftarrow$-direction. Assume that F is a C-operator that implements Ψ. Define a winning strategy for $\mathcal{O}$ in an obvious way.

Round 0. If $\mathcal{I}$ chooses a_0, then $\mathcal{O}$ replies by $F(X_0)$ the output produced to the input signal which is a_0 everywhere. If $\mathcal{I}$ interrupts at $t_1 > t_0$ and chooses $a_1 \neq a_0$, then $\mathcal{O}$ replies by Y'_1, where Y'_1 is the restriction of $Y_1 = F(X_1)$ on $[t_1, \infty)$, where X_1 is a_0 on $[0, t_1)$ and a_1 on $[t_1, \infty)$.

Continue in a similar way for the other rounds. This strategy is winning for $\mathcal{O}$. □

5.1. **An automaton Game.** Let $\mathcal{A}$ be a deterministic parity automaton equivalent to Ψ^D. We modify the game $\mathcal{G}_\psi$ as follows. Player $\mathcal{O}$ chooses signals (equivalently, timed words) over the set $Q := Q_\mathcal{A}$ of states of $\mathcal{A}$, instead of timed words over Σ_{out}. Of course, the timed words should be consistent with the behavior of $\mathcal{A}$. A play is winning for $\mathcal{O}$, if the constructed ω-word satisfies the acceptance conditions of $\mathcal{A}$. Alternatively, we can pretend that $\mathcal{O}$ chooses timed words over Σ_{out}. However, we are interested in the corresponding timed words over Q.

Here is a modified game $\mathcal{G}_\mathcal{A}$ on the automaton $\mathcal{A}$ equivalent to Ψ^D.

$\mathcal{G}_{\mathcal{A}}$ Game

Round 0: Set $t_0 := 0$ and $q_0 := q_{init}$.

(1) $\mathcal{I}$ chooses a value $a_0 \in \Sigma_{in}$ which holds at t_0 and for a while after t_0.

(2) $\mathcal{O}$ responds with a timed ω-word (equivalently, with a signal) $Y'_0 = (q_1, \tau_1), \cdots (q_k, \tau_k) \cdots$ over Q such that $\tau_1 > t_0$ and $q_{j+1} = \delta_{\mathcal{A}}(q_j, (a_0, b_j))$ for some $b_j \in \Sigma_{out}$ and all $j \geq 0$.
Now, $\mathcal{I}$ has two options:
A) $\mathcal{I}$ accepts. In this case, the play ends. $\mathcal{O}$ has constructed $Y := Y'_0$.
B) $\mathcal{I}$ interrupts at some $t_1 > t_0$ by choosing $a_1 \neq a_0$ which holds at t_1 and for a while after t_1. Proceed to step 1 of round 1.

Round $i+1$: Set $q_0 := Y'_i(t_{i+1})$.

(1) $\mathcal{O}$ chooses a timed ω-sequence $Y'_{i+1} := (q_1, \tau_1), \cdots (q_k, \tau_k) \cdots$ over Q such that $\tau_1 > t_{i+1}$ and $q_{k+1} = \delta_{\mathcal{A}}(q_k, (a_{i+1}, b_k))$ for some $b_k \in \Sigma_{out}$ and all $k \geq 0$.

(2) Now, $\mathcal{I}$ has two options.
A) $\mathcal{I}$ accepts. In this case, the play ends. The signal Y constructed by $\mathcal{O}$ coincides with Y'_j on $[t_j, t_{j+1})$ for $j \leq i$ and with Y'_{i+1} on $[t_{i+1}, \infty)$.
B) $\mathcal{I}$ interrupts at some $t_{i+2} > t_{i+1}$ by choosing $a_{i+2} \neq a_{i+1}$ for X at t_{i+2} and for a while after it. In this case, start round $i+2$.

Winning Condition: If the play lasted ω rounds, set $Y := Y'_i$ for $t \in [t_i, t_{i+1})$. For plays of finite length Y was defined by option A). $\mathcal{O}$ wins if either the ω-sequence of states of Y produced by $\mathcal{O}$ is accepted by $\mathcal{A}$, or there were ω rounds, but $\lim t_i$ is finite.

The following is immediate.

Lemma 5.3 ($\mathcal{G}_\Psi$ and $\mathcal{G}_{\mathcal{A}}$ are equivalent)**.** *$\mathcal{O}$ wins in $\mathcal{G}_\Psi$ if and only if $\mathcal{O}$ wins in $\mathcal{G}_{\mathcal{A}}$.*

Now, we are going to restrict $\mathcal{O}$ moves. The first restriction is a time schedule. When $\mathcal{O}$ chooses a timed sequence $(q_0, \tau_0)(q_1, \tau_1) \cdots (q_i, \tau_i) \cdots$ we require that $\tau_{i+1} - \tau_i = \tau_{j+1} - \tau_j$ for all i, j. We call such a sequence (signal) *simple timed*. Another restriction is on the sequence of values. We require that this sequence is ultimately periodic.

In the next subsection we describe the ultimately periodic sequences that we need.

5.2. **Ultimately Periodic Sequences.**

Definition 5.4 (Ultimately periodic strings)**.** An ω-string w is ultimately periodic if $w = uv^\omega$ for some (finite) strings u and v; v (respectively, u) is the periodic (respectively, pre-periodic or lag) part of w; the length of v (respectively, of u) is period (respectively, lag) of w.

We denote by $u[n]$ the n-th letter of a string u and by $u[k, m)$ the set of letters $\{u[i] \mid k \leq i < m)\}$.

Definition 5.5 ($\equiv_{state}$-equivalence on finite strings)**.** Let $\mathcal{A}$ be an automaton over a set Q of states and let u, v be finite strings over Q. We say that u is equivalent to v (notation $u \equiv_{state} v$) if and only if

(1) The first state of u is the same as the first state of v, and the last state of u is the same as the last state of v.

(2) For every m there is n such that $u[m] = v[n]$ and the states that appear in $u[0, m)$ are the same as the states that appear in $v[0, n)$.
(3) Symmetrically, for every n there is m such that $u[m] = v[n]$ and the states that appear in $u[0, m)$ are the same as the states that appear in $v[0, n)$.
(4) u is run of $\mathcal{A}$ if and only if v is a run of $\mathcal{A}$.

For the sake of formality, we have to use notations $u \equiv^{\mathcal{A}}_{state} v$ to emphasize that this equivalence relation is parameterized by $\mathcal{A}$. However, $\mathcal{A}$ will be clear from the context or unimportant and we will use $\equiv_{state}$ instead of $\equiv^{\mathcal{A}}_{state}$.

The next lemma summarizes properties of $\equiv_{state}$-equivalence:

Lemma 5.6 (Properties of $\equiv_{state}$)**.** *For every $\mathcal{A}$:*
(1) *$\equiv_{state}$ has finitely many equivalence classes.*
(2) *$\equiv_{state}$ is a congruence with respect to concatenation, i.e., if $u_1 \equiv_{state} v_1$ and $u_2 \equiv_{state} v_2$, then $u_1u_2 \equiv_{state} v_1v_2$.*
(3) *For every ω-string π there are $u_0, \dots, u_n, \dots$ such that $\pi = u_0 \cdots u_n \cdots$ and $u_i \equiv_{state} u_j \equiv_{state} u_iu_{i+1}$ for all $i, j > 0$ and $u_0 \equiv_{state} u_0u_1$.*

Proof. (1) and (2) are immediate. (3) will be proved in the Appendix. □

Definition 5.7 (Equivalence on ω-strings)**.** Let $\mathcal{A}$ be an automaton over a set Q of states, and let u, v be ω-strings over Q. We say that u is equivalent to v iff
(1) $\mathit{Inf}(u) = \mathit{Inf}(v)$, i.e., a state appears infinitely often in u iff it appears infinitely often in v.
(2) for every m there is n such that $u[m] = v[n]$ and the states that appear in $u[0, m)$ are the same as the states that appear in $v[0, n)$,
(3) Symmetrically, for every n there is m such that $u[m] = v[n]$ and the states that appear in $u[0, m)$ are the same as the states that appear in $v[0, n)$.
(4) u is a run of $\mathcal{A}$, if and only if v is a run of $\mathcal{A}$.

The next lemma is immediate. It deals with the concatenations of ω-sequences of finite strings.

Lemma 5.8. *if $u_i \equiv_{state} v_i$ for $i \in \omega$, then $u_0u_1 \dots$ is equivalent to $v_0v_1 \dots$*

From Lemma 5.8 and Lemma 5.6 we obtain:

Lemma 5.9. *For every automaton $\mathcal{A}$ over a set Q of states there is a finite set of ultimately periodic strings $UP(Q)$ (=$UP_{\mathcal{A}}(Q)$ computable from $\mathcal{A}$) such that for every ω-word π over Q there is $\rho = uv^{\omega} \in UP(Q)$ which is equivalent to π, and $v \equiv_{state} vv$ and $u \equiv_{state} uv$. Moreover, there are $u_0, u_1, \cdots, u_i, \cdots$ such that $\pi = u_0u_1 \cdots u_i \cdots$ and $u_0 \equiv_{state} u$ and $u_i \equiv_{state} v$ for every $i > 0$.*

Proof. For each $\equiv_{state}$-equivalence class (over finite strings) choose a (shortest) string u in it. This string is called a representative of its $\equiv_{state}$-equivalence class. Let $u_1, \dots, u_k$ be the representatives. A $\equiv_{state}$-equivalence class C is idempotent if $uv \equiv_{state} u$ for every $u, v \in C$, equivalently $uu \equiv_{state} u$ for an element $u \in C$. Define $UP(Q) := \{u_iu_j^{\omega} \mid u_i$ and u_j are representatives, and u_j is in an idempotent class and $u_iu_j \equiv_{state} u_i\}$. Lemma 5.6(3) implies that every ω-string is equivalent to a string in $UP(Q)$. Lemma 5.6(3) also implies the "moreover part" of Lemma 5.9. □

Notation 5.10. Let $d_Q := \max\{|u| \mid u$ is a shortest string of its $\equiv_{state}$-equivalence class$\}$. Then, d_Q is an upper bound on the length of periods v and lags of ω-strings in $UP(Q)$.

Lemma 5.11. *Assume $v \equiv_{state} vv$ and $u \equiv_{state} uv$ and $w = uv^\omega = q_1q_2\cdots$. Let n and m be the lengths of u and v respectively. Let S_u be the set of states that appear in u. Then, for every $l \geq n$, the set of states that appear in $q_1\cdots q_l$ is S_u. Moreover, for every $l, l' > n$ if $q_l = q_{l'}$, then $q_1\cdots q_l \equiv_{state} q_1\cdots q_{l'}$.*

5.3. **Restrictions on strategies of $\mathcal{O}$.**

Definition 5.12 (Simple Strategies)**.** A timed ω-word $(a_0,t_0)(a_1,t_1)\cdots$ is simple timed if there is $\Delta > 0$ such that $t_i - t_0 = \Delta \times i$ for all i. An $\mathcal{O}$-strategy is *simple* (for $\mathcal{A}$) if it chooses only simple timed $UP_{\mathcal{A}}(Q)$-sequences, i.e., the sequences of the form $(a_0,0)(a_1,\Delta)\cdots(a_i,\Delta\times i)\cdots$, where $a_0a_1\cdots \in UP_{\mathcal{A}}(Q)$.

Lemma 5.13 (Reduction to simple strategies in $\mathcal{G}_{\mathcal{A}}$)**.** *$\mathcal{O}$ has a winning strategy if and only if it has a simple winning strategy.*

Proof. Let σ be a winning strategy for $\mathcal{O}$ in $\mathcal{G}_{\mathcal{A}}$. We are going to construct a simple winning strategy. First, for a timed ω-sequence $u := (a_0,0)(a_1,t_1)\cdots$ let $first(u) := t_1$ be its first positive point and $untimed(u)$ be the corresponding untimed ω-sequence $a_0a_1\cdots$.

By u^s we denote the simple timed sequence v such that $untimed(u)$ is equivalent to $untimed(v) \in UP(Q)$ and the scale of v is $t_1 := first(u)/d_Q$ where d_Q is greater than or equal to the length of the lag of $untimed(v)$.

We are going to show that a strategy τ which replaces u moves of $\mathcal{O}$ in σ by the corresponding u^s moves is winning.

Consider a play π consistent with τ. We construct a play π^σ consistent with σ such that the states visited infinitely often in π are the same as the state visited infinitely often in π^σ, and the duration of π^σ is greater than or equal to the duration of π. Hence, if π^σ is winning, then π is winning.

For every prefix π_n of π where $\mathcal{O}$ moves, we construct the corresponding prefix π^σ_n of π^σ, where $\mathcal{O}$ moves and the following invariants hold:

I-last state: π_n and π^σ_n end in the same states.

I-recurrent states: The infix between π_{n-1} and π_n has the same states as the infix between π^σ_{n-1} and π^σ_n. I.e., let v_n (respectively, v^σ_n) be such that $\pi_n = \pi_{n-1}v_n$ (respectively, $\pi^\sigma_n = \pi^\sigma_{n-1}v^\sigma_n$); then a state occurs in v_n if and only if it occurs in v^σ_n.

I-duration: The duration of π^σ_n is greater than or equal to the duration of π_n.

Assume that a prefix π_n was constructed and the corresponding prefix is π^σ_n (they end in the same state). If σ prescribes for $\mathcal{O}$ to choose u, let u^s be the corresponding choice for τ.

Now, if $\mathcal{I}$ accepts u^s, then we are done; both plays are accepting or rejecting. If $\mathcal{I}$ interrupts at a state q, we consider two cases.

A) $\mathcal{I}$ interrupted u^s at time t at a position inside the period. Then, we can find $t' > t$ such that u is at the state q at t' and the corresponding prefixes are $\equiv_{state}$-equivalent. In fact, by Lemma 5.9 and Lemma 5.11, there are infinitely many such positions in u. Now, we constructed π^σ_{n+1} prefix with the duration greater than the duration of corresponding π_{n+1} prefix.

B) $\mathcal{I}$ interrupts u^s at time t in a state q at a position in the lag; hence, $t < t_1 = \mathit{first}(u)$. We choose the corresponding interruption point in u. It is at time greater than or equal to t_1. Hence, the invariants are preserved.

It remains to show that π is winning for $\mathcal{O}$ if π^σ is winning for $\mathcal{O}$.

If the plays are finite, then π is winning iff π^σ is winning.

If the plays are infinite, then (1) by the recurrent state invariant the same states appear infinitely often in π and in π^σ, and (2) if $\mathcal{I}$ causes time to converge in π^σ to $t^\sigma < \infty$, then in π time converges to a value bounded by $t^\sigma < \infty$. Since $\mathcal{O}$ wins π^σ, we obtain that $\mathcal{O}$ wins π. □

Now, we will play explicitly on a finite arena. This game corresponds to $\mathcal{G}_\mathcal{A}$-game when $\mathcal{O}$ is restricted to use only simple strategies.

Let $\mathcal{A} := (Q, \Sigma_{in}, \Sigma_{out}, \delta, q_{init}, pr)$ be an automaton. We construct the following arena:

Arena $\mathcal{G}_\mathcal{A}^{arena}$

Nodes: $\{q_{new}\} \cup Q \times \Sigma_{in} \cup Q \times \Sigma_{in} \times UP(Q)$, where $Q_\mathcal{I} := \{q_{new}\} \cup Q \times \Sigma_{in} \times UP(Q)$ is the set of $\mathcal{I}$ nodes, and $Q_\mathcal{O} := Q \times \Sigma_{in}$ is the set of $\mathcal{O}$ nodes. The set F of final nodes is $\{(q, a, u) \mid$ the maximal priority that appears infinitely often in u is even$\}$.

Edges:

(1) q_{new} to (q_{init}, a) for every $a \in \Sigma_{in}$.

(2) (q_0, a) to (q_0, a, u), where $u = (q_1, q_2, \dots) \in UP(Q)$ such that $\exists b_i \in \Sigma_{out}.\ q_{i+1} = \delta(q_i, (a, b_i))$ for all i.

(3) There is an edge from (q, a, u) to (q', b) for $u \in UP(Q)$, if $a \neq b \in \Sigma_{in}$ and there is n such that $u(n) = q'$; this edge will be labeled by p - the maximal priority among $u(1), \dots, u(n)$ and it will be called small if $u(1), \dots, u(n)$ is in the lag part of u; otherwise, it will be called big.

This arena is a finite graph (in (3) construction of edges depends on $n \in \mathbb{N}$, however, due to periodicity of u, the number of edges is finite and they are computable). There might be parallel edges from (q, a, u) to (q', b) which will have labels (p, big) or $(p, small)$ for a priority p. If k is the number of priorities, then from (q, a, u) to (q', b) for every $p \leq k$ there is at most one edge $(p, small)$; and by Lemma 5.11, there is a unique p' such that there is an edge (p', big) from (q, a, u) to (q', b); moreover, this p' is the maximal of the priorities assigned to the states of u.

Now, consider the following game.

Timed game $\mathcal{G}_\mathcal{A}^{arena}$ on our finite arena

Initial step. $\mathcal{I}$ chooses $a \in \Sigma_{in}$ and the play proceeds to (q_{init}, a); t_0 is set to 0.
From a state (q, a) at the i-th move at time t_i player $\mathcal{O}$ chooses an edge to (q, a, u) and a time scale Δ_i. It defines a sequence Y_i on $[t_i, \infty)$ as $Y(t_i + \Delta_i \times n) = q$ if q is the n-th state of u.
From a state (q, a, u), at $i+1$-move Player $\mathcal{I}$ either (A) accepts and the play ends, or (B) interrupts at $t_{i+1} > t_i$ by chosen $b \neq a$. Then, the play moves to (q_n, b) by an edge with priority p if $t_i + \Delta_i \times (n-1) < t_{i+1} \leq t_i + \Delta_i \times n$ and p is the maximal of priorities among $q_1, \cdots, q_n$.
Winning Condition. $\mathcal{O}$ wins a play π if either π is finite (this happens if $\mathcal{I}$ has chosen option (A)) and its last state is in F, or π is infinite and $\lim t_i$ is finite, or the maximal priority that appears infinitely often is even.

Note that in the above game the arena is finite, yet each player has uncountable many moves, when he/she chooses time. The following is immediate.

Lemma 5.14. *$\mathcal{O}$ wins $\mathcal{G}_\mathcal{A}^{arena}$ if and only if $\mathcal{O}$ has a winning simple strategy for $\mathcal{G}_\mathcal{A}$-game.*

Next, we are going to restrict further the strategies of player $\mathcal{O}$ in $\mathcal{G}_{\mathcal{A}}^{arena}$.

A strategy is called **almost positional**, if at every vertex of the arena it always chooses the same untimed move (time scales might be different).

Lemma 5.15 (Almost positional winning strategy in $\mathcal{G}_{\mathcal{A}}^{arena}$)**.** *$\mathcal{O}$ has a simple winning strategy iff $\mathcal{O}$ has an almost positional simple winning strategy.*

Proof. First, note that $\mathcal{G}_{\mathcal{A}}^{arena}$ game is determined, i.e., one of the player has a winning strategy.

Let σ be a simple strategy for $\mathcal{O}$. For every vertex s in $Q_{\mathcal{O}}$, let $Consistent(\sigma, s)$ be the set of possible (untimed) moves from s which are consistent with σ, i.e., $u \in Consistent(\sigma, s)$ if there is a (finite) run $\pi := s_0 s_1 \dots s_n$ consistent with σ such that $s = s_n$ and σ prescribes to choose u' after π, where $u = untimed(u')$. Let out_s^σ be the cardinality of $Consistent(\sigma, s)$.

Assume σ is winning. We prove by induction on $\sum out_s^\sigma$ that $\mathcal{O}$ has an almost positional simple winning strategy..

Basis. $out_s^\sigma = 1$ for every s. Hence, σ is almost positional.

Inductive step. Assume $out_s^\sigma > 1$. Let u be a move from s consistent with σ.

Let arena G_u be obtained from $\mathcal{G}_{\mathcal{A}}^{arena}$ when only u moves (with different scales) are available from s, and on the other nodes $\mathcal{G}_{\mathcal{A}}^{arena}$ and G_u coincide. Similarly, let arena $G_{\neg u}$ be obtained from $\mathcal{G}_{\mathcal{A}}^{arena}$ when u moves are forbidden and only the moves from $Consistent(\sigma, s) \setminus \{u\}$ are allowed from s, and on the other nodes $\mathcal{G}_{\mathcal{A}}^{arena}$ and G_u coincide.

If $\mathcal{O}$ wins in G_u or in $G_{\neg u}$, by the inductive hypothesis $\mathcal{O}$ has an almost positional winning strategy in G_u or in $G_{\neg u}$, and this strategy will be winning in $\mathcal{G}_{\mathcal{A}}^{arena}$.

Assume $\mathcal{I}$ wins in these games by strategies μ_u and $\mu_{\neg u}$. We are going to show that this implies that $\mathcal{I}$ has a winning strategy in $\mathcal{G}_{\mathcal{A}}^{arena}$ and this contradicts that σ is a winning strategy for $\mathcal{O}$.

Let πv be a finite play in G from s. We say that a move v is a G_u move if the last move from s in πv was u, otherwise v is a $G_{\neg u}$ move.

Let $View_u(\pi)$ (respectively, $View_{\neg u}(\pi)$) be the sequence of G_u (respectively, $G_{\neg u}$) moves in π.

Define a strategy μ for $\mathcal{I}$ as follows. Assume π ends in a position for $\mathcal{I}$, then

$$\mu(\pi) = \begin{cases} \mu_u(View_u(\pi)) & \text{if the last move of } \pi \text{ is a } G_u \text{ move} \\ \mu_{\neg u}(View_{\neg u}(\pi)) & \text{otherwise} \end{cases}$$

Consider a play π which is consistent with μ. If from some moment on π stays in G_u (respectively, $G_{\neg u}$), then from this moment it is consistent with μ_u (respectively, with $\mu_{\neg u}$), and therefore it is winning for $\mathcal{I}$.

Assume π infinitely often switches between G_u and $G_{\neg u}$. Since $View_u(\pi)$ (respectively, $View_{\neg u}(\pi)$) is consistent with μ_u (respectively, with $\mu_{\neg u}$) we obtain that $View_u(\pi)$ and $View_{\neg u}(\pi)$ are winning plays for $\mathcal{I}$. Hence, their durations are infinite. The maximal priority that occurs infinitely often in $View_u(\pi)$ is odd and the maximal priority that occurs infinitely often in $View_{\neg u}(\pi)$ is odd. Hence, the maximal priority that occurs infinitely often in π is odd, and π is winning play for $\mathcal{I}$. □

Finally, we restrict/eliminate the time scale that Player $\mathcal{O}$ chooses.

Lemma 5.16 (2^{-i} scale)**.** *$\mathcal{O}$ has an almost positional simple winning strategy if and only if $\mathcal{O}$ has an almost positional simple winning strategy when the i-th time move has the scale 2^{-i}.*

Proof. Let σ be an almost positional simple winning strategy for $\mathcal{O}$ and σ^{exp} be the corresponding almost positional simple strategy for $\mathcal{O}$ when the i-th time move has scale 2^{-i}.

Assume π^{exp} is a play consistent with σ^{exp} which is winning for $\mathcal{I}$. Toward a contradiction, we are going to construct π consistent with σ which is winning for $\mathcal{I}$.

We will keep the following invariant: After the i-th move of $\mathcal{I}$ the corresponding subplays π_i and π_i^{exp} (1) have the same last state, (2) pass through the same set of states, and (3) infinitely often the duration of π_i is greater than the duration of π_i^{exp}. Moreover, (4) if $\pi_{i+1} = \pi_i w_{i+1}$ and $\pi_{i+1}^{exp} = \pi_i^{exp} w'_{i+1}$, then w_{i+1} is state equivalent to w'_{i+1}.

Let u^{exp} be the $i+1$-th move of $\mathcal{O}$. Assume $\mathcal{I}$ interrupts it at a state q at time t. If q is inside the period, we can find q which at time $t' > t$ (far enough in the u move) and interrupts u at the same q with the same past. If q is inside the prefix lag, we pick up the corresponding q in the prefix of u. Hence, $\pi_{i+1} = \pi_i w_{i+1}$ and $\pi_{i+1}^{exp} = \pi_i^{exp} w'_{i+1}$ and w_{i+1} is state equivalent to w'_{i+1}.

It is clear that invariants (1), (2) and (4) are preserved. If the duration of π^{exp} is infinite, then there are infinitely many interruptions inside the periods of $\mathcal{O}$ moves. Hence, for each such interruption at n-th move we can choose the corresponding interruption in u that ensures that the duration of π_n is greater than the duration of π_n^{exp}. We assumed that π^{exp} is winning for $\mathcal{I}$. Therefore, the duration of π^{exp} is infinite and the third property holds.

To sum up, we proved that if there is a play consistent with σ^{exp} which is winning for $\mathcal{I}$, then there is a play consistent with σ which is winning for $\mathcal{I}$ a contradiction to the assumption that σ is a winning strategy for $\mathcal{O}$. ☐

5.4. **Decidability.** We proved that $\mathcal{O}$ has a winning strategy iff $\mathcal{O}$ has an almost positional simple winning strategy where the i-th time move has the scale 2^{-i}. There are finitely many such strategies. Hence, to decide whether $\mathcal{O}$ has a winning strategy it is enough to check whether such a strategy is winning for $\mathcal{O}$.

Let σ be an almost positional strategy for $\mathcal{O}$ with the time scale 2^{-i} on the i-th move. It chooses exactly one edge from every vertex (q, a). Consider the sub-graph G_σ of $\mathcal{G}_{\mathcal{A}}^{arena}$ where from (q, a) only the edge specified by σ is chosen (all other edges from (q, a) are removed). This is an arena for one player $\mathcal{I}$. Recall that the edges from the nodes of the form (q, a, u) are labeled (see the definition of the arena on page 17). The edge which corresponds to the move from (q, a, u) when $\mathcal{I}$ interrupts at q_n is assigned a priority p, where p is the maximal priority assigned to the states $q_1, \cdots, q_n$; this edge is small if $q_1 \cdots q_n$ is inside lag of u; otherwise, it is big. The edge will carry a label $(p, small)$ or (p, big). Note "big" and "small" are just names that indicate whether an interruption appears in the period part or in the lag part, and they do not affect the order of priorities.

Lemma 5.17 (Deciding whether a strategy is winning)**.** *Let σ be an almost positional winning strategy for $\mathcal{O}$ with the time scale 2^{-i} on the i-th move. Player $\mathcal{I}$ wins against σ if and only if in G_σ either (A) there is a reachable node $(q, a, u) \notin F$ or (B) there is a reachable cycle C with at least one edge labeled by a big priority and the maximal priority in C is odd.*

Proof. It is clear that if (A) holds, then $\mathcal{I}$ wins against σ. She only needs to reach such a node $(q, a, u) \notin F$ and then stop playing. If (B) holds, then $\mathcal{I}$ can reach such a cycle C and then move along its edges. Moreover, when she goes over an edge labeled by a big priority p, she can choose a time t which is at distance at least one from the time when the corresponding edge to (q, a, u) is taken (by Lemma 5.11, she can choose times as big as she wishes). This will ensure that every round along C takes at least one time unit; therefore, $\lim t_i = \infty$. Since the maximal priority on C is odd, this play is winning for $\mathcal{I}$.

Now, we assume that $\mathcal{I}$ wins and show that (A) or (B) holds. If (A) does not hold, then a winning play for $\mathcal{I}$ must be infinite. We are going to find a cycle as described in (B). If only small priorities appear infinitely often, then the duration of the play is finite and it cannot be winning for $\mathcal{I}$. Indeed, assume that after t_j only small priorities appear in the play. Recall that d_Q is an upper bound on the length of lags for $UP(Q)$ moves. Then, $t_{i+1} - t_i < d_Q \times 2^{-i}$ for all $i \geq j$. Hence, $\lim t_i \leq t_j + 2d_Q < \infty$ and $\mathcal{I}$ loses.

Therefore, there is an edge e_b with a big priority that appears infinitely often. Let p be the maximal priority that appears infinitely often (it is odd). Then, some edge e_p with p appears infinitely often, and there is a sub-play that starts and ends at e_p and passes through e_b. This sub-play is a cycle that passes through an edge with a big priority and the maximal priority of its edges is p and p is odd. □

There are only finitely many almost positional strategies for $\mathcal{O}$ with the time scale 2^{-i} on the i-th move (in $\mathcal{G}^{arena}_{\mathcal{A}}$), and by Lemma 5.17, it is decidable whether such a strategy is winning, and $\mathcal{O}$ wins iff it has such a winning strategy. Hence, we obtain:

Lemma 5.18. *It is decidable whether $\mathcal{O}$ has a wining strategy $\mathcal{G}^{arena}_{\mathcal{A}}$.*

Recall $\mathcal{O}$ has a winning strategy in $\mathcal{G}^{arena}_{\mathcal{A}}$ iff $\mathcal{O}$ has a winning strategy in $\mathcal{G}_{\mathcal{A}}$ iff $\mathcal{O}$ has a winning strategy in $\mathcal{G}_{\Psi}$ iff there is a causal operator that implements Ψ. Therefore, we obtain:

Theorem 5.19 (Decidability of Synthesis for RC Signals)**.** *The Church synthesis problem over RC signals is decidable.*

6. The Church Synthesis Problem over FV Signals is Decidable

In this section a proof of the decidability of Church's synthesis problem over FV signals is outlined.

Theorem 1.6 (Decidability of Synthesis)**.** *Given an MSO$[<]$ formula $\Psi(X,Y)$, it is decidable whether there exists a C-operator F such that $FVsig \models \forall X.\ \Psi(X, F(X))$.*

The proof is very similar to the proof of Theorem 5.19, and does not require novel insights or techniques. Here, we explain the differences.

We start with a game. Consider the following two-player game $\mathcal{G}_{\Psi}$. During the play $\mathcal{I}$ constructs a (timed ω-sequence for) input signal X and $\mathcal{O}$ constructs an output Y.

Round 0. Set $t_0 := 0$.

(1) $\mathcal{I}$ chooses a value $a_0 \in \Sigma_{in}$ which holds at t_0.
(2) $\mathcal{O}$ responds with $b_0 \in \Sigma_{out}$.
(3) $\mathcal{I}$ chooses $a_0^d \in \Sigma_{in}$ to hold for a while after t_0 (here and below the superscript d in b^d indicates that the duration of b is non-singular, i.e., this input b lasts for a positive amount of time).
(4) $\mathcal{O}$ responds with a timed ω-sequence (equivalently, with a signal) Y_0 on (t_0, ∞).

Now, $\mathcal{I}$ has two options:

A) $\mathcal{I}$ accepts. In this case, the play ends. $\mathcal{I}$ has constructed X which is a_0 at $t_0 = 0$ and a_0^d at all $t > 0$, and $\mathcal{O}$ has constructed Y which is b_0 at t_0 and Y_0 - the signal chosen by $\mathcal{O}$ in (t_0, ∞).

B) $\mathcal{I}$ interrupts at some $t_1 > t_0$ by choosing left or right discontinuity in the input.

Discontinuity from the left: $\mathcal{I}$ chooses $a_1 \neq a_0^d$ and the game proceeds to step (1) of round 1, or

Discontinuity from the right: $\mathcal{I}$ sets $a_1 = a_0^d$ and $a_1^d \neq a_0^d$ and the game proceeds to step (3) of round 1.

Round $i+1$.

(1) $\mathcal{O}$ chooses $b_{i+1} \in \Sigma_{out}$ to hold at t_{i+1}.
(2) $\mathcal{I}$ chooses a_{i+1}^d to hold for a positive amount of time after t_{i+1}.
(3) $\mathcal{O}$ responds with a timed ω sequence Y'_{i+1} (equivalently, with a signal) on (t_{i+1}, ∞).
(4) Now, $\mathcal{I}$ has two options.

A) $\mathcal{I}$ accepts. In this case the play ends. The signal X constructed by $\mathcal{I}$ is a_j at t_j, is a_j^d on $[t_j, t_{j+1})$ for $j \leq i$, is a_{i+1} at t_{i+1} and is equal a_{i+1}^d on (t_{i+1}, ∞). The signal Y constructed by $\mathcal{O}$ is defined similarly, by concatenating the signals constructed by $\mathcal{O}$ at rounds $0, \dots, i+1$.

B) $\mathcal{I}$ interrupt at some $t_{i+2} > t_{i+1}$ by choosing:

(a) $a_{i+2} \neq a_{i+1}^d$ in this case round $i+2$ starts at step (1) (in this case X is left discontinuous at t_{i+1}) or by
(b) $a_{i+2} = a_{i+1}^d$ and $a_{i+2}^d \neq a_{i+1}^d$. In this case round $i+2$ starts at step (3) (in this case X is right discontinuous at t_{i+2}).

The signal X constructed by $\mathcal{I}$ corresponds to the timed sequence $(a_0, a_0^d, 0)(a_1, a_1^d, t_1)\dots (a_i, a_i^d, t_i)\dots$, and the signal Y constructed by $\mathcal{O}$ is the concatenatin of the signals constructed by $\mathcal{O}$ in each round..

Winning conditions. $\mathcal{O}$ wins the play iff $\Psi(X, Y)$ holds or $\mathcal{I}$ interrupts infinitely often and the duration of the play is finite, i.e., $\lim t_n < \infty$, where t_n is the time of n-th interruption.

Similarly, to Lemma 5.1, we have

Lemma 6.1 (Reduction to games). *$\mathcal{O}$ wins $\mathcal{G}_\Psi$ if and only if there is a C-operator that implements Ψ.*

Note that in this game $\mathcal{O}$ has two types of moves. In step (1) she chooses a letter in Σ_{out} and in Step (3) she chooses a Σ_{out}-signal over open interval (t, ∞). Similarly to the proof of Theorem 5.19, we are going to restrict the $\mathcal{O}$ moves in step (3). The first restriction is a time schedule. When $\mathcal{O}$ chooses a timed sequence (equivalently a signal), we require that the distance between consecutive time points in the sequence is the same. We call such a sequence *simple timed*. Another restriction is on the sequence of values. We require that this sequence is ultimately periodic. But first, we consider a game over an automaton in which $\mathcal{O}$ chooses signals over states instead of signals over Σ_{out}. She chooses the run which corresponds to a choice of an input over Σ_{out}.

Consider the game $G_\mathcal{A}$ over a deterministic parity automaton $\mathcal{A} := (Q, \Sigma_{in} \times \Sigma_{out}, \delta, q_{init}, pr)$ equivalent to Ψ^D. In this game $\mathcal{O}$ will choose signals over Q instead of signals over Σ_{out}.

Game $G_\mathcal{A}$.

Round 0. Set $t_0 := 0$.

(1) $\mathcal{I}$ chooses a value $a_0 \in \Sigma_{in}$ which holds at t_0.
(2) $\mathcal{O}$ responds with $q_0 := \delta(q_{init}, (a_0, b_0))$ for some $b_0 \in \Sigma_{out}$.
(3) $\mathcal{I}$ chooses $a_0^d \in \Sigma_{in}$ to hold for a positive amount of time after t_0.

(4) $\mathcal{O}$ responds with a signal Y_0 on (t_0, ∞) over Q such that Y_0 is a run of $\mathcal{A}$ (from q_0) when the signal over Σ_{in} is fixed to be a_0^d.

Now, $\mathcal{I}$ has two options.

A) $\mathcal{I}$ accepts. In this case, the play ends. $\mathcal{I}$ has constructed X which is a_0 at $t_0 = 0$ and a_0^d at all $t > 0$, and $\mathcal{O}$ has constructed Y which is q_0 at t_0 and is Y_0 - the signal chosen by $\mathcal{O}$ in (t_0, ∞).

B) $\mathcal{I}$ interrupts at some $t_1 > t_0$ by choosing left or right discontinuity in the input. Set q_1 to be the left limit of Y_0 at t_1.

Discontinuity from the left: $\mathcal{I}$ chooses $a_1 \neq a_0^d$ and the game proceeds to step (1) of round 1, or

Discontinuity from the right: $\mathcal{I}$ sets $a_1 = a_0^d$ and $a_1^d \neq a_0^d$ and the game proceeds to step (3) of round 1.

Round $i+1$.

(1) $\mathcal{O}$ updates $q_{i+1} := \delta(q_{i+1}, (a_{i+1}, b_{i+1}))$ for some $b_{i+1} \in \Sigma_{out}$ to hold at t_{i+1}.
(2) $\mathcal{I}$ chooses a_{i+1}^d to hold for a positive amount of time after t_{i+1}.
(3) $\mathcal{O}$ responds with a signal Y_{i+1} over Q on (t_{i+1}, ∞) which is a run of $\mathcal{A}$ from q_{i+1} when the signal over Σ_{in} is fixed to be a_{i+1}^d.
(4) Now, $\mathcal{I}$ has two options:

A) $\mathcal{I}$ accept. In this case the play ends. The signal Y constructed by $\mathcal{O}$ coincides with Y_l on (t_l, t_{l+1}), is equal to q_{l+1} at t_{l+1} (for $l \leq i$), and coincides with Y_{i+1} on (t_{i+1}, ∞).

B) $\mathcal{I}$ interrupt at some $t_{i+2} > t_{i+1}$. Set q_{i+2} to be the left limit of Y_{i+1} at t_{i+2}. $\mathcal{I}$ chooses either

(a) $a_{i+2} \neq a_{i+1}^d$ and then round $i+2$ starts at step (1) (in this case X is left discontinuous at t_{i+2}) or by
(b) $a_{i+2} = a_{i+1}^d$ and $a_{i+2}^d \neq a_{i+1}^d$. In this case round $i+2$ starts at step (3) (in this case X is right discontinuous at t_{i+2}).

Winning conditions. $\mathcal{O}$ wins if either the sequence of states produced by $\mathcal{O}$ is accepted by $\mathcal{A}$, or there were infinitely many moves, but $\lim t_i$ is finite.

Similarly, to Lemma 5.3, we have

Lemma 6.2 ($\mathcal{G}_\psi$ and $\mathcal{G}_\mathcal{A}$ are equivalent)**.** *$\mathcal{O}$ wins in $\mathcal{G}_\Psi$ if and only if $\mathcal{O}$ wins in $\mathcal{G}_\mathcal{A}$.*

Exactly as in Subsection 5.2 we define $\equiv_{state}$-equivalence over finite strings and an equivalence over ω-strings. We define a finite set of ultimately periodic ω-strings $UP(Q)$ ($= UP_\mathcal{A}(Q)$) and say that a move $(q_0, t_0)(q_1.t_1) \dots (q_i, t_i) \dots$ is simple if $q_0 q_1 \dots q_i \dots \in UP(Q)$ and $t_{i+1} = t_i + \Delta$ for some Δ. A strategy is simple if it uses only simple moves. We have

Lemma 6.3 (Reduction to simple strategies)**.** *$\mathcal{O}$ has a winning strategy if and only if $\mathcal{O}$ has a simple winning strategy.*

Now, we will play explicitly on a finite (node) arena. This game corresponds to the game when $\mathcal{O}$ is restricted to use only simple strategies.

Let $\mathcal{A} := (Q, \Sigma_{in} \times \Sigma_{out}, \delta, q_{init}, pr)$ be a parity automaton. We construct the following arena. It has more states than an arena for right continuous game, because we need to deal also with signals which are discontinuous from the right.

The nodes (vertex) of $\mathcal{O}$ are of the form (q, a) or $(q, \dagger, a)$ where $q \in Q_\mathcal{A}$ and $a \in \Sigma_{in}$. At a moment t a play is in (q, a) if the automaton is in state q and $\mathcal{I}$ chooses a to hold at the moment t. From this state $\mathcal{O}$ should choose an output at t and move to a node of the form $(q', \dagger)$. When a play is at $(q, \dagger, a)$ at t it indicates that $\mathcal{A}$ was at state q and $\mathcal{I}$ chooses for

$a \in \Sigma_{in}$ to holds for an interval of positive length after t. From this node $\mathcal{O}$ should respond with a signal on (t, ∞). Our arena $\mathcal{G}_{\mathcal{A}}^{arena}$ is described below.

Arena $\mathcal{G}_{\mathcal{A}}^{arena}$

Nodes: $\{q_{new}\} \cup Q \times \Sigma_{in} \cup Q \times \{\dagger\} \cup Q \times \{\dagger\} \times \Sigma_{in} \cup Q \times \Sigma_{in} \times UP(Q)$, where $Q_{\mathcal{I}} := \{q_{new}\} \cup Q \times \{\dagger\} \times \Sigma_{in} \cup Q \times \Sigma_{in} \times UP(Q)$ is the set of $\mathcal{I}$ nodes, and the rest is the set of $\mathcal{O}$ nodes.

The set F of final nodes is $\{(q, \dagger, u) \mid$ the maximal priority that appears infinitely often in u is even$\}$.

Edges:

(1) q_{new} to (q_{init}, a) for every $a \in \Sigma_{in}$.
(2) (q, a) to $(q', \dagger)$ if $q' = \delta(q, (a, b))$ for some $b \in \Sigma_{out}$.
(3) $(q, \dagger)$ to $(q, \dagger, a)$ for every $a \in \Sigma_{in}$ (here $\mathcal{I}$ chooses a to hold for a while).
(4) $(q_0, \dagger, a)$ to (q_0, a, u), where $u = (q_1, q_2, \dots) \in UP(Q)$ such that $\exists b_i \in \Sigma_{out}.\ q_{i+1} = \delta(q_i, (a, b_i))$ for all i.
(5) There is an edge from (q, a, u) to (q', b) for $u \in UP(Q)$, if $b \neq a$ and $q' = u(n)$ for $n = 2m + 1$ (this corresponds to discontinuity from the left).
(6) There is an edge from (q, a, u) to $(q', \dagger, b)$ for $u \in UP(Q)$, if $b \neq a$ and $q' = u(n)$ for $n = 2m$ (this corresponds to discontinuity from the right).

In (5) and (6) the edge will be labeled by p - the maximal priority among $u(1), \dots, u(n)$ and it will be called small if $u(1), \dots, u(n)$ is in the lag part of u; otherwise, it will be called big. The nodes also have priorities which are inherited from their Q components, i.e., $pr(q, a) = pr(q, a, u) = pr(q, \dagger) = pr(q, \dagger, a) = pr_{\mathcal{A}}(q)$.

Now, consider the following game.

Timed game on $\mathcal{G}_{\mathcal{A}}^{arena}$

Initial step. $\mathcal{I}$ chooses $a \in \Sigma_{in}$ and the play proceeds to (q_{init}, a); t_0 is set to 0.
When the play is at node (q, a), $\mathcal{O}$ chooses an edge to $(q', \dagger)$ and the play is now at $(q', \dagger)$.
From $(q, \dagger)$, $\mathcal{I}$ can move to $(q, \dagger, a)$.
From $(q, \dagger, a)$, $\mathcal{O}$ chooses an edge to (q, a, u) and a time scale τ_i. It defines a signal Y_i on (t_i, ∞) as $Y(t_i + \tau_i \times n) = u(2n)$ and $Y_i(t) = u(2n + 1)$ if $t \in (t_i + \tau_i \times n, t_i + \tau_i \times (n + 1))$.
From a state (q, a, u) at $i+1$-move Player $\mathcal{I}$ either (A) accepts and the play ends, or (B) interrupts at $t_{i+1} > t_i$ by choosing $b \neq a$ and the left or the right discontinuity at t_{i+1}. Let q be the left limit of Y_i at t_{i+1}.
If the left discontinuity is chosen, then the play moves to (q, b). If the right discontinuity is chosen, the play moves to $(q, \dagger, b)$.
Winning Condition. $\mathcal{O}$ wins a play π if either π is finite (this happens if $\mathcal{I}$ has chosen option (A)) and its last state is in F or π is infinite and $\lim t_i$ is finite, or the maximal priority that appears infinitely often is even.

Lemma 6.4. *$\mathcal{O}$ wins $\mathcal{G}_{\mathcal{A}}^{arena}$ if and only if $\mathcal{O}$ has a winning simple strategy for $\mathcal{G}_{\mathcal{A}}$-game.*

Next, like in the proof of Theorem 5.19, we are going to restrict further the strategies of $\mathcal{O}$ in $\mathcal{G}_{\mathcal{A}}^{arena}$.

A strategy is called almost positional, if at every node it always chooses the same untimed move (time scales might be different).

Lemma 6.5 (Almost positional winning strategy in $\mathcal{G}_{\mathcal{A}}^{arena}$)**.** *$\mathcal{O}$ has a simple winning strategy if and only if $\mathcal{O}$ has an almost positional simple winning strategy.*

Finally, we restrict the time scale that Player $\mathcal{O}$ chooses. We say that a strategy of $\mathcal{O}$ is well-scaled if the i-th time move from a state of the form $(q, \dagger, a)$ has the scale 2^{-i}.

Lemma 6.6 (2^{-i} scale)**.** *$\mathcal{O}$ has an almost positional simple winning strategy if and only if $\mathcal{O}$ has a well-scaled almost positional simple winning strategy.*

Lemma 6.7 (Deciding whether a well-scaled strategy is winning)**.** *Let σ be a well-scaled almost positional winning strategy for $\mathcal{O}$. It is decidable whether σ is winning.*

Since there are only finitely many well-scaled almost positional winning strategies, we obtain that

Lemma 6.8. *It is decidable whether $\mathcal{O}$ has a winning strategy in timed game on $\mathcal{G}_{\mathcal{A}}^{arena}$.*

Finally, Theorem 1.6 follows from Lemmas 6.1-6.4 and Lemma 6.8.

7. Conclusion

We investigated a generalization of the Church synthesis problem to the continuous time domain of the non-negative reals. We demonstrated that in continuous time there are phenomena which are very different from the canonical discrete time domain of natural numbers. We proved the decidability the Church synthesis problem when the specifications are given by $MSO[<]$ formulas.

One can consider a specification language stronger than $MSO[<]$. For example, let $\mathcal{M} := \langle \mathcal{R}^{\geq 0}, SIG, <_R, P_1, \dots, P_k \rangle$ be an extension of Sig structure by unary predicates $P_1, \dots, P_k$. If $MSO[<, P_1, \dots, P_k]$ decidable, then the Church synthesis problem for specifications given by $MSO[<, P_1, \dots, P_k]$ formulas is decidable.

The extension of $MSO[<]$ by $+1$ function is undecidable. Hence, the Church synthesis problem for $MSO[<, +1]$ is undecidable. There are decidable fragments of $MSO[<, +1]$ which use metric constraints in a more restricted form. The most widespread such decidable metric language is Metric Interval Temporal Logic (MITL) [AFH96]. We do not know whether the Church problem is decidable for MITL specifications. Proof techniques from [DGRR09] might be usefull to show decidability/undecidability of the Church synthesis problem for sub-logics of MITL.

The satisfiability problem for $MSO[<]$ is non-elementary. The complexity of our algorithm for the Church synthesis problem is non-elementary because any translation of $MSO[<]$-formulas to equivalent parity automata is non-elementary. Another reason for the high complexity of the algorithm is that $\mathcal{O}$ chooses her moves from $UP(Q)$ and the size of $UP(Q)$ is triple exponential (in the size of Q).

Fattal considered specifications of signal languages given by deterministic parity automata [Fat23]. He reduced the Church problem to games in which $\mathcal{O}$ selects, on her move, a single letter from the output alphabet along with its duration. He provided an NP upper bound for the synthesis problem when the specifications are given by deterministic parity automata. For deterministic Büchi automata specifications, Fattal found a polynomial-time algorithm. However, even for specifications given by deterministic parity automata with three priorities, it remains unknown whether a polynomial-time algorithm exists. (In the discrete case, for every fixed bound p on the priorities of a parity automaton, there is a polynomial algorithm.)

Acknowledgement

We would like to express our gratitude to Ori Lahav and the anonymous referees for their insightful comments.

References

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994. doi:10.1016/0304-3975(94)90010-8.

[AFH96] Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The benefits of relaxing punctuality. *Journal of the ACM*, 43(1):116–146, 1996. doi:10.1145/227595.227602.

[Büc60] J. Richard Büchi. On a decision method in restricted second order arithmetic. *Proceedings of the International Congress on Logic, Methodology and Philosophy of Science*, pages 1–11, 1960.

[BL69] J. Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969. URL: http://www.jstor.org/stable/1994916.

[Bou05] Patricia Bouyer. Synthesis of timed systems. http://www.lsv.fr/~bouyer/files/bouyer-games.pdf, 2005. Slides of invited lecture, Spring School on Infinite Games and Their Applications, Bonn, Germany.

[CHP11] Krishnendu Chatterjee, Thomas Henzinger, and Vinayak Prabhu. Timed parity games: Complexity and robustness. *Logical Methods in Computer Science*, 7, 12 2011. doi:10.2168/LMCS-7(4:8)2011.

[DGRR09] Laurent Doyen, Gilles Geeraerts, Jean-François Raskin, and Julien Reichert. Realizability of real-time logics. In *Proceedings of the 7th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, volume 5810 of *Lecture Notes in Computer Science*, pages 14–16. Springer, 2009. doi:10.1007/978-3-642-04368-0_2.

[Fat23] Daniel Fattal. Selection and uniformization over continuous time. Master's thesis, Tel-Aviv university, 2023.

[FLTM02] M. Faella, S. La Torre, and A. Murano. Dense real-time games. In *Proceedings 17th Annual IEEE Symposium on Logic in Computer Science*, pages 167–176, 2002. doi:10.1109/LICS.2002.1029826.

[JORW11] Mark Jenkins, Joël Ouaknine, Alexander Rabinovich, and James Worrell. The church synthesis problem with metric. In Marc Bezem, editor, *Computer Science Logic, 25th International Workshop / 20th Annual Conference of the EACSL, CSL 2011, September 12-15, 2011, Bergen, Norway, Proceedings*, volume 12 of *LIPIcs*, pages 307–321. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2011. doi:10.4230/LIPIcs.CSL.2011.307.

[LS98] Shmuel Lifsches and Saharon Shelah. Uniformization and skolem functions in the class of trees. *The Journal of Symbolic Logic*, 63(1):103–127, 1998. doi:10.2307/2586688.

[Rab02] A. Rabinovich. Finite variability interpretation of monadic logic of order. *Theoretic]al Computer Science*, 275:111–125, 2002.

[Rab07] Alexander Rabinovich. On decidability of monadic logic of order over the naturals extended by monadic predicates. *Information and Computation*, 205(6):870–889, 2007. doi:10.1016/j.ic.2006.12.004.

[She75] S. Shelah. The monadic theory of order. *The Annals of Mathematics*, 102(3):379–419, 1975. URL: http://www.jstor.org/stable/1971037.

[TB73] Boris Trakhtenbrot and Y. M. Barzdin. *Finite Automata: Behavior and Synthesis*. North-Holland, Amsterdam, 1973.

[Tra95] B. Trakhtenbrot. Origins and metamorphoses of the trinity: logic, nets, automata. In *Proceedings of Tenth Annual IEEE Symposium on Logic in Computer Science*, pages 506–507, 1995. doi:10.1109/LICS.1995.523284.

Appendix A. Proof of Lemma 5.6(3)

A coloring of ω is a function f from the set of unordered pairs of distinct elements of ω into a finite set T of colors. For $x < y$, we will write $f(x, y)$ instead of $f(\{x, y\})$. The coloring f is additive if $f(x,y) = f(x',y')$ and $f(y,z) = f(y',z')$ imply that $f(x,z) = f(x',z')$. In this case a partial function $+$ is defined on T such that if $x < y < z$, then $f(x,y) + f(y,z) = f(x,z)$. A set J is t-homogeneous (for f and the color t) if $f(x,y) = t$ for every $x < y \in J$.

Ramsey's theorem for additive coloring [She75] states:

Theorem A.1. *If f is an additive coloring of ω, then there is $t \in T$ and an infinite t-homogeneous subset J of ω. Moreover, there is $t_0 \in T$ such that $f(0,j) = t_0$ for every $j \in J$.*

Now, we are ready to prove Lemma 5.6(3).

Let the set of colors be the set of $\equiv_{state}$-equivalence classes. Given π. Define coloring f as $f(i,j) :=$ the equivalence class of the substring of π on the interval $[i,j)$. By Lemma 5.6(2), this is an additive coloring. Let $J = \{j_0 < j_1 < \cdots\}$ be an infinite homogeneous subset of color v (such a set exists by Theorem A.1).

Define u_0 to be the substring of π over $[0, j_0)$, and u_j to be the substring of π over $[u_j, u_{j+1})$ for $j > 0$. Let v_0 be the equivalence class of u_0.

By Theorem A.1 we obtain u_i and $u_i u_{i+1}$ and u_j are all in the v-equivalence class for all $i, j > 0$, and $u_0 \equiv_{state} u_0 u_1$ are both in v_0 equivalence class. Therefore, $u_i \equiv_{state} u_j \equiv_{state} u_i u_{i+1}$ for all $i, j > 0$ and $u_0 \equiv_{state} u_0 u_1 = u_0 u_1 \cdots u_j$. fot all $j \geq 0$.

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, 171 Second St, Suite 300, San Francisco, CA 94105, USA, or Eisenacher Strasse 2, 10777 Berlin, Germany