The Church Synthesis Problem with Metric

Mark Jenkins[1], Joël Ouaknine[1], Alexander Rabinovich*[2], and James Worrell[1]

1 **Department of Computer Science, University of Oxford, Oxford, UK**
{marj,joel,jbw}@cs.ox.ac.uk

2 **School of Computer Science, Tel Aviv University, Tel Aviv, Israel**
rabinoa@post.tau.ac.il

Abstract

Church's Problem asks for the construction of a procedure which, given a logical specification $\varphi(I, O)$ between input strings I and output 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 gave a procedure to solve Church's problem for MSO specifications and operators computable by finite-state automata.

We consider extensions of Church's problem in two orthogonal directions: (i) we address the problem in a more general logical setting, where not only the specifications but also the solutions are presented in a logical system; (ii) we consider not only the canonical discrete time domain of the natural numbers, but also the continuous domain of reals.

We show that for every fixed bounded length interval of the reals, Church's problem is decidable when specifications and implementations are described in the monadic second-order logics over the reals with order and the +1 function.

1998 ACM Subject Classification F.4.1 Mathematical Logic, F.4.3 Formal Languages

Keywords and phrases Church's Problem, Monadic Logic, Games, Uniformization

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Church's *synthesis problem* [3] is to automatically construct an implementation of a specification relating the inputs and outputs of a state-based system. The specification is assumed to be an $\mathsf{MSO}(<)$-formula $S(I, O)$, which determines a binary relation between input strings I and output strings O. An implementation is a function (or operator) P from strings to strings that *uniformizes* S in the sense that $S(I, P(I))$ holds for all inputs I. Church required that P be computable by a finite-state machine that at every moment $t \in \mathbb{N}$ reads an input symbol $I(t)$ and produces an output symbol $O(t)$. Hence, the output $O(t)$ produced at t depends only on input symbols $I(0), I(1), \ldots, I(t)$ received before t, that is, P should be a *causal* operator. Another property of interest is that the machine computing P be finite-state. In the light of Büchi's proof [1] of the expressive equivalence of $\mathsf{MSO}(<)$ and finite automata, P is finite-state if and only if it is $\mathsf{MSO}(<)$-definable.

Church's synthesis problem can therefore be stated formally as follows.

* This author was partially supported by the ESF GAMES programme and by the EPSRC.

Church Synthesis Problem

Input: an $\mathsf{MSO}(<)$ formula $\varphi(X, Y)$.
Task: Check whether there is a causal operator F such that
$\langle \mathbb{N}, < \rangle \models \forall X \varphi(X, F(X))$ and if so, construct this operator.

This problem, which is more general than the satisfiability problem for MSO over $\langle \mathbb{N}, < \rangle$, was shown decidable in a landmark paper of Büchi and Landweber [2]. Their main theorem is stated as follows:

▶ **Theorem 1** (Büchi and Landweber). *Given an* $\mathsf{MSO}(<)$ *formula* $\varphi(\overline{X}, \overline{Y})$ *one can decide whether there is a causal operator that uniformizes* φ. *If such an operator exists then it can be represented by a finite-state automaton which can be computed from* φ.

Note that this theorem guarantees that whenever φ has a uniformizer then it has a uniformizer that is computable by a finite-state automaton (equivalently, definable in $\mathsf{MSO}(<)$).

In the continuous-time setting, one can naturally consider the synthesis problem over the non-negative reals rather than the naturals. Here we think of a specification as a relation between *signals* rather than words. As specification language one again takes $\mathsf{MSO}(<)$, which has a natural interpretation over the non-negative reals. As implementations one again takes $\mathsf{MSO}(<)$-definable causal operators.

Shelah [13] proved that $\mathsf{MSO}(<)$ is undecidable over the reals if we allow quantification over arbitrary predicates. In Computer Science however, it is natural to restrict to finitely variable predicates, that is, predicates whose characteristic function has finitely many discontinuities in any bounded interval. Under the finite-variability interpretation, MSO is decidable over the reals and there are automata that have the same expressive power [10].

However, the full extension of the Büchi and Landweber theorem fails over the nonnegative reals, even under the finite-variability assumption. For example, the formula that says that Y has at least two points of discontinuity can be uniformized by a causal operator, but not by an $\mathsf{MSO}(<)$-*definable causal operator* (see Example 7 for details).

Nevertheless, we are able to show the following result:

▶ **Theorem 2.** *Given a* $\mathsf{MSO}(<)$ *formula* $\varphi(\overline{X}, \overline{Y})$ *one can decide whether there is an* $\mathsf{MSO}(<)$-*definable causal operator that uniformizes* φ *over* $\langle \mathbb{R}_{\geq 0}, < \rangle$. *If so, the algorithm computes a formula that defines such an operator.*

In the continuous setting a deficiency of $\mathsf{MSO}(<)$ is that it cannot express metric properties such as "the distance between two points is one". Thus we consider specifications expressed in $\mathsf{MSO}(<, +1)$, which extends $\mathsf{MSO}(<)$ with the $+1$ function.

Unfortunately, even with the finitely-variable interpretation, the satisfiability problem over the non-negative reals is undecidable for $\mathsf{MSO}(<, +1)$ [5]. However in [9] we proved that $\mathsf{MSO}(<, +1)$ is decidable for every fixed bounded-length intervals of the reals. The main result of our paper is that Church's synthesis problem is also decidable for $\mathsf{MSO}(<, +1)$ for every fixed bounded-length interval of the reals. Specifically,

▶ **Theorem 3.** *Given an* $\mathsf{MSO}(<, +1)$ *formula* $\varphi(X, Y)$ *and* $N \in \mathbb{N}$, *one can decide whether there is an* $\mathsf{MSO}(<, +1)$-*definable causal operator that uniformizes* φ *over the the interval* $[0, N)$. *If such an operator exists, the algorithm computes a formula that represents the operator.*

In order to prove Theorem 3 we need to consider the Church synthesis problem with *parameters*—additional predicates that the specification may reference but which do not have to be considered in a causal way by the implementation. This problem was considered in [11] for $\mathsf{MSO}(<)$ over $\langle \mathbb{N}, < \rangle$. Here we extend the Church synthesis problem with parameters to the non-negative reals.

Finally, we show that the synthesis problem over bounded intervals of reals is non-elementary, even for specifications expressed in fragments of $\mathsf{MSO}(<, +1)$ with an elementary satisfiability problem.

2 Monadic Second-Order Logic

We consider monadic second-order logic $\mathsf{MSO}(<, +1)$ over a signature consisting of the binary relations $<$ and $+1$ and a countable family of monadic predicate names $P_0, P_1, \ldots$. The vocabulary of $\mathsf{MSO}(<, +1)$ also includes *first-order* variables $t_0, t_1, \ldots$ and monadic *second-order* variables $X_1, X_2, \ldots$. Atomic formulas are of the form $X(t)$, $P(t)$, $t_1 < t_2$, $+1(t_1, t_2)$ or $t_1 = t_2$. Well-formed formulas are obtained from atomic formulas using Boolean connectives, the first-order quantifiers $\exists t$ and $\forall t$, and the second-order quantifiers $\exists X$ and $\forall X$. We denote by $\mathsf{MSO}(<)$ the sub-language consisting of all formulas that do not mention the $+1$ relation.

We are interested in structures of the form $\mathcal{M} = \langle A, <, +1, \mathbf{P}_1, \ldots, \mathbf{P}_m \rangle$ where A is an interval of non-negative reals with the usual order, $+1(x, y)$ holds if and only if $y = x + 1$, and $\mathbf{P}_1, \ldots, \mathbf{P}_m$ are subsets of A that interpret the monadic predicate names $P_1, \ldots, P_m$. (Generally we use boldface to denote interpretations of predicate names.) We omit the standard definition of what it means for a structure to satisfy a sentence. A formula $\varphi(P_1, \ldots, P_m, X_1, \ldots, X_n)$ with free second-order variables among $X_1, \ldots, X_n$ is interpreted in a structure $\langle \mathcal{M}, \mathbf{X}_1, \ldots, \mathbf{X}_n \rangle$ obtained by expanding $\mathcal{M}$ with interpretations of $X_1, \ldots, X_n$.

We say that a subset $\mathbf{P} \subseteq \mathbb{R}_{\geq 0}$ is *finitely variable* if its characteristic function has finitely many discontinuities in any bounded sub-interval of $\mathbb{R}_{\geq 0}$. Likewise we say that $\mathbf{P}$ is *right-continuous* if its characteristic function is right continuous. In our semantics for $\mathsf{MSO}(<, +1)$ we restrict interpretations of monadic predicate names and variables to be finitely variable and right-continuous. The finite variability restriction is essential: it is known that allowing unrestricted second-order quantification leads to an undecidable satisfiability problem [13]. On the other hand, the assumption of right-continuity is only for simplicity of presentation. In case the domain A is unbounded we also make the simplifying assumption that at least one of the predicates in any structure $\langle A, <, \overline{\mathbf{P}} \rangle$ is not eventually constant.

We also consider discrete structures for $\mathsf{MSO}(<)$ of the form $\langle A, <, \mathbf{P}_1, \ldots, \mathbf{P}_m \rangle$, where A is an initial segment of the natural numbers with the usual order and $\mathbf{P}_1, \ldots, \mathbf{P}_m$ are subsets of A. In line with our assumption for structures over the reals we assume that if $A = \mathbb{N}$ then in any structure $\langle A, <, \overline{\mathbf{P}} \rangle$, at least one of the predicates is not eventually constant.

Let $\Sigma_P = \{0, 1\}^m$ be a finite alphabet. A structure $\langle A, <, \mathbf{P}_1, \ldots, \mathbf{P}_m \rangle$ corresponds to a function $f : A \to \Sigma_P$, where $f(t)_i = 1$ if $t \in \mathbf{P}_i$ and $f(t)_i = 0$ otherwise. If $A \subseteq \mathbb{R}_{\geq 0}$ then f is called a *signal* and if $A \subseteq \mathbb{N}$ then f is a (finite or infinite) word. By assumption, a signal is finitely variable, right-continuous and if its domain is unbounded it is not eventually constant. We denote by Σ^ω the set of all infinite words over an alphabet Σ and by $\mathit{Sig}(\Sigma)$ the set of all signals over Σ with domain $\mathbb{R}_{\geq 0}$. An $\mathsf{MSO}(<)$-sentence $\varphi(\overline{P})$ that mentions predicate names $P_1, \ldots, P_m$ respectively defines a word language $L_{\mathbb{N}}(\varphi) \stackrel{\text{def}}{=} \{w \in (\Sigma_P)^\omega : w \models \varphi\}$ and a signal language $L_{\mathbb{R}}(\varphi) \stackrel{\text{def}}{=} \{f \in \mathit{Sig}(\Sigma_P) : f \models \varphi\}$.

3 Transforming Between Words and Signals

In this paper we answer questions about MSO over the non-negative reals under the finite variability and right-continuous interpretation by reduction to questions about MSO over the naturals. This section presents the foundations of this reduction—semantic translations between signal languages and word languages and corresponding syntactic translations on MSO($<$) formulas. We concentrate here on signals with domain $\mathbb{R}_{\geq 0}$ and on infinite words, though the ideas easily apply to signals with bounded domain and to finite words.

Let $f : \mathbb{R}_{\geq 0} \to \Sigma$ be a signal over alphabet Σ. Recall that by assumption f has a countably infinite and unbounded set of discontinuities. Define a *sampling sequence* for f to be an unbounded strictly increasing sequence of reals $0 = \tau_0 < \tau_1 < \tau_2 < \ldots$ that includes all discontinuities of f. Given a sampling sequence τ we define the word $W_\tau(f) \in \Sigma^\omega$ by $W_\tau(f) = f(\tau_0)f(\tau_1)f(\tau_2)\ldots$. Given a language $L \subseteq \mathit{Sig}(\Sigma)$ we define the corresponding word language $L^\dagger \subseteq \Sigma^\omega$ to comprise all words $W_\tau(f)$ where $f \in L$ and τ is a sampling sequence for f.

Motivated by the translation above we define the relation $\sim$ of *stutter equivalence* on Σ^ω to be the least equivalence relation such that

$$a_0a_1\ldots a_{k-1}a_ka_{k+1}\ldots \sim a_0a_1\ldots a_{k-1}a_ka_ka_{k+1}\ldots$$

for any k. A word language $L \subseteq \Sigma^\omega$ is *stutter-closed* if it saturates $\sim$. It is straightforward that if L is a signal language then the corresponding word language $L^\dagger$ is stutter-closed.

Define the *stutter-closure* of a language $L \subseteq \Sigma^\omega$ to be the smallest stutter-closed language L' that contains L. It is straightforward that L' is ω-regular if L is ω-regular.

Given a word $w = w_0w_1w_2\ldots \in \Sigma^\omega$ and an unbounded strictly increasing sequence of reals $0 = \tau_0 < \tau_1 < \tau_2 < \ldots$, define the signal $S_\tau(w) : \mathbb{R}_{\geq 0} \to \Sigma$ by $S_\tau(t) = w_k$ for the unique interval $[\tau_k, \tau_{k+1})$ containing t. Given a word language $L \subseteq \Sigma^\omega$, the corresponding signal language $L^\star$ comprises all signals $S_\tau(w)$ for some $w \in L$ and unbounded sequence of reals τ.

Define the relation $\sim$ of *stretching equivalence* on the set $\mathit{Sig}(\Sigma)$ of signals over alphabet Σ by $f \sim g$ iff $f = g \circ \rho$ for some order isomorphism $\rho : \mathrm{dom}(f) \to \mathrm{dom}(g)$. A signal language L is *speed-independent* if it saturates $\sim$. It is straightforward that $L_\mathbb{R}(\varphi)$ is speed-independent for any MSO($<$)-formula φ. It is also clear that $L^\star$ is speed-independent for any word language $L \subseteq \Sigma^\omega$.

The operators $(-)^\dagger$ and $(-)^\star$ define a bijection between stutter-closed word languages and speed-independent signal languages:

▶ **Proposition 4.** *If $L \subseteq \Sigma^\omega$ is stutter-closed then $L^{\star\dagger} = L$. If $L \subseteq \mathit{Sig}(\Sigma)$ is speed independent then $L^{\dagger\star} = L$.*

Next we recall from [10] syntactic analogs of the language operators $(-)^\dagger$ and $(-)^\star$. The following results show that the language operators $(-)^\dagger$ and $(-)^\star$ preserve MSO-definability. We briefly justify these constructions and refer the reader to [10] for full details.

▶ **Proposition 5.** *Given an* MSO($<$)*-sentence $\varphi(\overline{P})$ we can compute another* MSO($<$)*-sentence $\varphi^\dagger(\overline{P})$ such that $L_\mathbb{R}(\varphi)^\dagger = L_\mathbb{N}(\varphi^\dagger)$.*

Proof. (Sketch.) In order to define $\varphi^\dagger$ we first rewrite φ by replacing first-order variables with second-order variables. Intuitively we represent an element $t \in \mathbb{R}_{\geq 0}$ by a set with left endpoint t. (Recall that the restriction to finitely variable right-continuous signals means any set interpreting a second-order variable has a least element.) To this end we introduce the following new atomic formulas: $Ls(X, Y)$, which is true if the left endpoint of X is less than the left endpoint of Y; $Eq(X,Y)$, which is true if the left endpoint of X equals the left endpoint of Y; $In(X,Y)$, which is true if the left endpoint of X is an element of Y. These atoms are clearly $\mathsf{MSO}(<)$-definable, moreover it is straightforward to construct a formula equivalent to φ using only these atoms, Boolean connectives and second-order quantification.

Having performed this translation, we proceed to define $\varphi^\dagger$ by structural induction on φ. The new atomic formulas $Ls(X,Y)$, $Eq(X,Y)$ and $In(X,Y)$ go unchanged, as do negation and conjunction. The only non-trivial element of the transformation concerns second-order quantification. If $\varphi \stackrel{\text{def}}{=} \exists X \psi$ we let $\varphi^\dagger$ be the formula that defines the stutter closure of the language $L_{\mathbb{N}}(\exists X \psi^\dagger)$. (We have already observed above that this language is definable.) After this process it is trivial to replace occurrences of $Ls(X,Y)$, $Eq(X,Y)$ and $In(X,Y)$ with their $\mathsf{MSO}(<)$-equivalents. ◀

▶ **Proposition 6.** *Given an* $\mathsf{MSO}(<)$*-sentence* $\varphi(\overline{P})$ *we can compute another* $\mathsf{MSO}(<)$*-sentence* $\varphi^\star(\overline{P})$ *such that* $L_{\mathbb{R}}(\varphi^\star) = L_{\mathbb{N}}(\varphi)^\star$.

Proof. (Sketch.) It is straightforward to write an $\mathsf{MSO}(<)$-formula $\mathit{Sample}(\overline{P}, D)$ that is satisfied when each discontinuity of the characteristic signal of $\overline{P}$ is also a discontinuity of the characteristic signal of D. Given $\varphi(\overline{P})$ let $\varphi'(\overline{P}, D)$ be obtained by relativizing all first-order quantification in φ to the set of discontinuities in D. We now define

$$\varphi^\star := \exists D(\mathit{Sample}(\overline{P}, D) \wedge \varphi'(\overline{P}, D)) .$$ ◀

4 Church's Problem with Parameters

Recall that a binary relation R is *uniformized* by a partial function f if $f \subseteq R$ and $\mathrm{dom}(f) = \mathrm{dom}(R)$. In this paper we are interested in uniformizing MSO-definable relations by MSO-definable functions. The problem of uniformizing MSO-definable relations on the structure $\langle \mathbb{N}, < \rangle$ was first studied over fifty years ago by Church [3], motivated by the problem of synthesizing circuits from relational input-output specifications. Later Rabinovich [11] and Hänsch, Slaats and Thomas [4] considered the problem of uniformization over *labelled chains* $\langle \mathbb{N}, <, \overline{\mathbf{P}} \rangle$.

Our eventual goal is to study uniformization of MSO-definable relations over the structure $\langle A, <, +1 \rangle$ for A a bounded interval of reals. We delay a treatment of the $+1$ relation until the following section. Here we lay the groundwork by considering uniformization of labelled chains $\langle \mathbb{R}_{\geq 0}, <, \overline{\mathbf{P}} \rangle$. This extends the treatment of the labelled case from [4, 11] to dense orders.

4.1 The Uniformization Problem

Consider a second-order language over a signature including the binary relation symbol $<$ and monadic predicate names $P_1, \ldots, P_m$. Let $\mathcal{M} = \langle A, <, \mathbf{P}_1, \ldots, \mathbf{P}_m \rangle$ be a labelled chain. We say that an MSO-formula $\psi(\overline{P}, \overline{X}, \overline{Y})$ uniformizes an MSO-formula $\varphi(\overline{P}, \overline{X}, \overline{Y})$ over $\mathcal{M}$ if $\mathcal{M}$ satisfies the following sentences:

1. $\forall \overline{X} \forall \overline{Y} \left(\psi(\overline{P}, \overline{X}, \overline{Y}) \rightarrow \varphi(\overline{P}, \overline{X}, \overline{Y}) \right)$
2. $\forall \overline{X} \exists^{=1} \overline{Y} \psi(\overline{P}, \overline{X}, \overline{Y})$

We say that $\psi(\overline{P}, \overline{X}, \overline{Y})$ uniformizes $\varphi(\overline{P}, \overline{X}, \overline{Y})$ over a class of chains $\mathcal{C}$ if $\psi(\overline{P}, \overline{X}, \overline{Y})$ uniformizes $\varphi(\overline{P}, \overline{X}, \overline{Y})$ over each individual chain in $\mathcal{C}$. Notice that the above conditions can only hold if $\varphi(\overline{P}, \overline{X}, \overline{Y})$ is a total relation, however there is no loss of generality in considering only uniformization for total relations.

We say that a formula $\psi(\overline{P},\overline{X},\overline{Y})$ satisfying 1 above is *faithful* to φ and a formula satisfying 2 above is *functional.* We furthermore say that ψ is *causal* if the following sentence holds in $\mathcal{M}$:

3. $\forall \overline{X}\,\overline{Y}\,\overline{U}\,\overline{V}\,\forall t\ \left[\psi(\overline{P},\overline{X},\overline{Y}) \wedge \psi(\overline{P},\overline{U},\overline{V}) \wedge \left(\forall s \le t\,(\overline{X}(s)=\overline{U}(s))\right) \Rightarrow \overline{Y}(t)=\overline{V}(t)\right]$

Intuitively a function is causal if its output at any time only depends on its input in the past—a reasonable assumption for any realizable function.

Roughly speaking, the uniformization problem is to determine whether a given formula $\varphi(\overline{P},\overline{X},\overline{Y})$ has a uniformizer over a given structure, or class of structures, and if so to compute such a uniformizer. We are interested here in uniformizers which are definable by an MSO formula and this proves an important restriction over structures with real-valued domains. The following example illustrates a case where one can easily think of a uniformizer for φ, but no such uniformizer can be definable in MSO.

▶ **Example 7.** There is a formula $\varphi(X,Y)$ (even without parameters) such that there is a causal operator which uniformizes φ over the reals however, no MSO-definable causal operator uniformizes it.

Proof. Let $\varphi(X,Y)$ be an MSO($<$)-formula which says that Y has at least two points of discontinuity. It is clear that the operator F which ignores its input and sets $F(X)(t)=1$ if and only if $t \in ([0,1)\cup[2,3))$ uniformizes this formula, however we prove below there is no MSO-definable uniformizer $\psi(X,Y)$.

Let $\psi(X,Y)$ be an MSO($<$)-formula that defines a functional operator. Interpret X by the constant false signal $\mathbf{X}$ and let $\mathbf{Y}$ be the unique interpretation of Y such that $\psi(\mathbf{X},\mathbf{Y})$ holds. For any order isomorphism $\rho:\mathbb{R}_{\ge 0}\to\mathbb{R}_{\ge 0}$, since every MSO($<$)-formula is speed-independent, $\psi(\mathbf{X}\circ\rho,\mathbf{Y}\circ\rho)$ also holds. But $\mathbf{X}\circ\rho=\mathbf{X}$; as ψ is functional we must also have $\mathbf{Y}\circ\rho=\mathbf{Y}$. It is easy to see that this entails that $\mathbf{Y}$ be constant, contrary to the requirement that it have two discontinuities. ◀

Motivated by failures such as this, we seek to compute the set of parameter values for which there exists a definable uniformizer along with a single formula which defines a uniformizer for all such parameter values. We formally state the main result of this section as follows:

▶ **Theorem 2.** *Given an* MSO($<$)*-formula* $\varphi(\overline{P},\overline{X},\overline{Y})$ *one can compute a sentence* $\theta(\overline{P})$ *and formula* $\psi(\overline{P},\overline{X},\overline{Y})$ *such that for every structure* $\mathcal{M}=\langle\mathbb{R}_{\ge 0},<,\overline{\mathbf{P}}\rangle$*,* φ *has a causal uniformizer over* $\mathcal{M}$ *if and only if* $\mathcal{M}\models\theta$ *and in this case* ψ *is such a causal uniformizer.*

In Theorem 2 we call φ the *winning condition*, ψ the *uniformizer* and θ the *domain formula.* We call the predicate names $\overline{P}$ *parameters* and $\overline{X},\overline{Y}$ *variables.*

We sketch a proof of Theorem 2 in Sections 4.2 and 4.3.

4.2 From Signals to Words

Let L_1 be a speed-independent language of signals over alphabet $\Sigma_P=\{0,1\}^m$ and let $L_2=(L_1)^\dagger$ be the corresponding stutter-closed language of words. We identify a signal f with the corresponding structure $\langle\mathbb{R}_{\ge 0},<,\mathbf{P}_1,\ldots,\mathbf{P}_m\rangle$, where f is the characteristic function of $\overline{\mathbf{P}}$; we similarly identify a word w with the corresponding structure $\langle\mathbb{N},<,\overline{\mathbf{P}}\rangle$. We further identify a language L with the class of stuctures that correspond to the elements of L.

We reduce the problem of computing a uniformizer of an MSO($<$)-formula $\varphi(\overline{P},\overline{X},\overline{Y})$ over the class of signals L_1 to the problem of computing a uniformizer for the corresponding formula $\varphi^\dagger(\overline{P},\overline{X},\overline{Y})$ over the class of words L_2. Superficially these two problems are quite different since L_1 is a class of dense orders and L_2 a class of discrete orders. The key fact that makes this reduction work is that for each signal $f \in L_1$ we include in L_2 the whole stutter-closed class of words representing f. Given this, the reduction is simply stated:

▶ **Theorem 8.** *If $\psi(\overline{P}, \overline{X}, \overline{Y})$ is a causal uniformizer for $\varphi(\overline{P}, \overline{X}, \overline{Y})$ over L_1 then $\psi^\dagger(\overline{P}, \overline{X}, \overline{Y})$ is a stutter-closed causal uniformizer for $\varphi^\dagger(\overline{P}, \overline{X}, \overline{Y})$ over L_2. Conversely if $\psi(\overline{P}, \overline{X}, \overline{Y})$ is a stutter-closed causal uniformizer for $\varphi^\dagger(\overline{P}, \overline{X}, \overline{Y})$ over L_2, then $\psi^\star(\overline{P}, \overline{X}, \overline{Y})$ is a causal uniformizer for $\varphi(\overline{P}, \overline{X}, \overline{Y})$ over L_1.*

The proof of Theorem 8 relies on the relations between speed-independent signal languages and stutter-closed word languages developed in Section 3.

We will use Theorem 8 to reduce the problem of uniformizing classes of signals, considered in Theorem 2, to the problem of computing stutter-closed uniformizers of stutter-closed formulas over words. We therefore undertake to prove the following Theorem.

▶ **Theorem 9.** *Given a stutter-closed* MSO$(<)$*-formula $\varphi(\overline{P}, \overline{X}, \overline{Y})$ one can compute a stutter-closed sentence $\theta(\overline{P})$ and stutter-closed formula $\psi(\overline{P}, \overline{X}, \overline{Y})$ such that for any stutter-closed language of words $L \subseteq (\Sigma_P)^\omega$, φ has a causal uniformizer over L if and only if $L \models \theta$ and in this case ψ is such a causal uniformizer.*

Considering φ over signals, we observe that the word language defined by $\varphi^\dagger$ is always stutter-closed. We therefore first apply Theorem 9 then Theorem 8 to $\varphi^\dagger$ to derive Theorem 2.

4.3 Stutter-Closed Uniformizers

Say that a formula $\psi(\overline{P}, \overline{X}, \overline{Y})$ defines a *stutter-preserving* relation on $\mathcal{M} = \langle \mathbb{N}, <, \overline{\mathbf{P}} \rangle$ if $\mathcal{M}$ satisfies $\forall \overline{X}, \overline{Y}(\psi(\overline{P}, \overline{X}, \overline{Y}) \rightarrow \mathit{StutPres}(\overline{P}, \overline{X}, \overline{Y}))$, where

$$\mathit{StutPres}(\overline{P}, \overline{X}, \overline{Y}) \overset{\text{def}}{=} \forall n(\overline{X}(n) = \overline{X}(n+1) \wedge \overline{P}(n) = \overline{P}(n+1) \rightarrow \overline{Y}(n) = \overline{Y}(n+1)) \quad (1)$$

In other words, in the function defined by ψ the output $\overline{Y}$ can only change when either the input $\overline{X}$ or parameters $\overline{P}$ change.

The following is straightforward.

▶ **Proposition 10.** *Let $L \subseteq (\Sigma_P)^\omega$ be a stutter-closed language of words. If $\psi(\overline{P}, \overline{X}, \overline{Y})$ is functional over L and stutter-closed then it is also stutter-preserving over L.*

Say that an ω-word $u = u_0 u_1 \ldots$ is *stutter-free* if $u_i \neq u_{i+1}$ for all i. Recall that we assume that for any structure $\langle \mathbb{N}, <, \overline{\mathbf{P}} \rangle$ one of the predicates $\mathbf{P}_i$ is not eventually constant. This means that the characteristic ω-word of the structure is stutter equivalent to a unique stutter-free word.

Proof of Theorem 9. We define a game based on φ such that the uniformizer ψ defines a winning strategy in this game. Our proof is based on the construction of Hänsch, Slaats and Thomas [4] but requires non-trivial modification to handle various issues related to stuttering.

Step 1: *definition of game arena $\mathcal{G}$.* Define a formula $\varphi'(\overline{P}, \overline{X}, \overline{Y})$ by

$$\varphi'(\overline{P}, \overline{X}, \overline{Y}) \overset{\text{def}}{=} (\varphi(\overline{P}, \overline{X}, \overline{Y}) \wedge \mathit{StutPres}(\overline{P}, \overline{X}, \overline{Y})) \vee \mathit{EvConst}(\overline{P}), \quad (2)$$

where *StutPres* is the formula in (1) expressing stutter preservation and $\mathit{EvConst}(\overline{P})$ expresses that $\overline{P}$ is eventually constant. The inclusion of *StutPres* is justified by the observation in Proposition 10 that a stuttering-closed uniformizer is stutter-preserving. The inclusion of

$EvConst(\overline{P})$ is connected with our semantic assumption that the characteristic words of structures over $\mathbb{N}$ are not eventually constant.

The formula φ' mentions predicate names $\overline{P} = (P_1, \ldots, P_m)$ and free variables $\overline{X} = (X_1, \ldots, X_n)$ and $\overline{Y} = (Y_1, \ldots, Y_\ell)$. Then interpretations for φ' over domain $\mathbb{N}$ are ω-words over the alphabet $\{0,1\}^m \times \{0,1\}^n \times \{0,1\}^\ell$. The first step is to construct a deterministic parity automaton $\mathcal{A}$ over this alphabet that accepts precisely those words that satisfy φ'. We transform this automaton into a parity game arena $\mathcal{G}$ by separating each transition $s \xrightarrow{(p,x,y)} t$ of $\mathcal{A}$ into a pair of transitions $s \longrightarrow (s,p,x)$ and $(s,p,x) \longrightarrow t$ controlled by Player 1 and Player 2 respectively. The priorities of the states in $\mathcal{G}$ are inherited from $\mathcal{A}$.

Step 2: *definition of parity game* $\mathcal{G}^\pi$. Next, given a stutter-free ω-word π over alphabet $\{0,1\}^m$, representing an interpretation of the parameters, we transform the arena $\mathcal{G}$ into an infinite-state parity game denoted $\mathcal{G}^\pi$. This game is stratified into finite levels—one level for each letter of π, the states at each level being a copy of those of $\mathcal{G}$. Multiple rounds of the game can be played at each level, with Player 1 controlling passage from one level to the next.

The states of $\mathcal{G}^\pi$ are pairs consisting of a state of $\mathcal{G}$ and a *level number* i. For each Player-1 edge $s \longrightarrow (s,p,x)$ in $\mathcal{G}$ and index $i \in \mathbb{N}$ we include a Player-1 edge $(s)_i \longrightarrow (s,p,x)_i$ if $p = \pi_i$ and an edge $(s)_i \longrightarrow (s,p,x)_{i+1}$ if $p = \pi_{i+1}$. For each Player-2 edge $(s,p,x) \longrightarrow t$ in $\mathcal{G}$ and index $i \in \mathbb{N}$ we include an edge $(s,p,x)_i \longrightarrow (t)_i$ in $\mathcal{G}^\pi$. Finally we add a new initial Player-1 state $(\hat{s})_0$ to $\mathcal{G}^\pi$, where s is the initial state of $\mathcal{G}$. From this state there is an edge to a state $(s,p,x)_0$ if $p = \pi_0$.

Note that due to the disjunct $EvConst(\overline{P})$ in (2) Player 1 cannot win $\mathcal{G}^\pi$ by choosing never to leave a given level.

The key property of the game $\mathcal{G}^\pi$, which depends on the fact that $\mathcal{A}$ is stutter-closed, is as follows:

Player 2 wins $\mathcal{G}^\pi$ if and only if $\varphi(\overline{P}, \overline{X}, \overline{Y})$ has a causal uniformizer over the class of all infinite words π' that are stutter equivalent to π.

The easier direction in the proof of the above claim is the right-to-left implication: one can easily show that a causal uniformizer for $\varphi(\overline{P}, \overline{X}, \overline{Y})$ over the stutter equivalence class of π yields a winning strategy in $\mathcal{G}^\pi$. Below we concentrate on the left-to-right implication.

Step 3: *coding and testing strategies.* As $\mathcal{G}^\pi$ is a parity game it is determined and has memoryless winning strategies. To compute winning strategies we first divide the set of game states into levels S_i which contain those nodes annotated with level number i. We can encode the possible levels by a finite alphabet Σ and thus represent the game as an ω-word $\sigma \in \Sigma^\omega$ in which σ_i represents level S_i. Note that σ can be produced as the output of a transducer $\mathcal{S}$ on input π.

A memoryless strategy for Player 2 in $\mathcal{G}^\pi$ maps each node $(s,p,x)_i$ to a node $(t)_i$ and can be represented by an word γ over a finite alphabet Γ whose letters encode the finite sub-strategy for each level i. We build a deterministic parity automaton $\mathcal{T}$ that takes as input pairs of words π and γ and accepts if and only if the strategy γ is winning in $\mathcal{G}^\pi$. The automaton $\mathcal{T}$ incorporates the transducer $\mathcal{S}$ to transform the input word π into a word σ representing the game $\mathcal{G}^\pi$ in the manner described above. For each level i of $\mathcal{G}^\pi$ and level-i Player-2 strategy γ_i, automaton $\mathcal{T}$ computes the finite set of all possibilities over Player-1 moves for the first state, last state and lowest-priority intermediate state of the level-i segment of of a play of $\mathcal{G}^\pi$.

The strategy tester automaton $\mathcal{T}$ is equivalent to an $\mathsf{MSO}(<)$-formula $\chi(\overline{P}, \overline{S})$, where $\overline{S}$ encodes letters of the strategy alphabet Γ. Note that χ is only satisfied by stutter-free interpretations of $\overline{P}$.

Step 4: *selecting a winning strategy.* The formula $\chi(\overline{P},\overline{S})$ allows us to detect when $\overline{S}$ encodes a winning strategy in $\mathcal{G}^\pi$. A key issue here is that there may be more than one winning strategy for a given set of parameters π: a uniformizer corresponds to a particular winning strategy.

We can compute an $\mathsf{MSO}(<)$-formula that picks a winning strategy using a result of Lifsches and Shelah [8] on the computability of selectors in $\mathsf{MSO}(<)$. We say that a formula $\alpha(\overline{P},\overline{S})$ is a *selector* for a formula $\beta(\overline{P},\overline{S})$ over a structure $\mathcal{M} = \langle \mathbb{N}, <, \overline{\mathbf{P}} \rangle$ if:

1. $\mathcal{M} \models \exists^{\leq 1}\overline{S}\,\alpha(\overline{P},\overline{S})$;
2. $\mathcal{M} \models \forall \overline{S}(\alpha(\overline{P},\overline{S}) \rightarrow \beta(\overline{P},\overline{S}))$;
3. $\mathcal{M} \models (\exists \overline{S}\beta(\overline{P},\overline{S})) \rightarrow (\exists \overline{S}\alpha(\overline{P},\overline{S}))$.

▶ **Lemma 11** (Selector Lemma [12]). *There is an algorithm that for every formula $\beta(\overline{P},\overline{S})$ constructs a formula $\alpha(\overline{P},\overline{S})$ such that α is a selector for β over all structures $\mathcal{M} = \langle \mathbb{N}, <, \overline{\mathbf{P}} \rangle$.*

Applying Lemma 11 we can compute a selector $\chi'(\overline{P},\overline{S})$ for $\chi(\overline{P},\overline{S})$. Then $\chi'(\overline{P},\overline{S})$ is satisfied when $\overline{P}$ is interpreted by a stutter-free word π and $\overline{S}$ is interpreted by a word representing a winning strategy γ in $\mathcal{G}^\pi$.

Step 5: *definition of $\theta(\overline{P})$ and $\psi(\overline{P},\overline{X},\overline{Y})$.*

Similar to the definition of the strategy tester automaton we can compute a formula $\mathit{Strat}(\overline{P},\overline{X},\overline{Y},\overline{S})$ that is true precisely when in $\mathcal{G}^\pi$, for π the unique stutter-free word equivalent to $\overline{P}$, the sequence of moves $\overline{X}$ by Player 1 generates the sequence of responses $\overline{Y}$ by Player 2 given that his strategy on the k-th round is $\overline{S}(k)$.

Note that $\mathit{Strat}(\overline{P},\overline{X},\overline{Y},\overline{S})$ defines a stutter-closed language. Closure under removing stutters follows from the fact that $\overline{S}$ encodes a memoryless strategy and (it can be assumed without loss of generality that) $\mathcal{A}$ doesn't change state when its input stutters. Closure under adding stutters follows similarly using in addition the fact that $\overline{S}$ encodes a stutter-preserving strategy.

We also define $\chi''(\overline{P},\overline{S})$ to be the stutter-closure of the strategy selection operator $\chi'(\overline{P},\overline{S})$. To define χ'' consider positions where $\overline{P}$ changes and state that χ' holds over this set of positions (just relativization) and that $\overline{S}$ does not change between changes of $\overline{P}$.

Finally we are able to define

$$\begin{aligned} \theta(\overline{P}) &\stackrel{\text{def}}{=} \exists \overline{S}\chi''(\overline{P},\overline{S}) \\ \psi(\overline{P},\overline{X},\overline{Y}) &\stackrel{\text{def}}{=} \exists \overline{S}\left(\chi''(\overline{P},\overline{S}) \wedge \mathit{Strat}(\overline{P},\overline{X},\overline{Y},\overline{S})\right) . \end{aligned}$$

Then both θ and ψ are stutter-closed since both χ'' and Strat are stutter-closed.

By construction, $\theta(\overline{\mathbf{P}})$ holds if and only if there exists $\overline{\mathbf{S}}$ that encodes a winning strategy for Player 2 in $\mathcal{G}^\pi$, where π the unique stutter-free word equivalent to the characteristic ω-word $u_{\overline{\mathbf{P}}}$. Since Strat encodes plays of this game that follow this winning strategy, ψ uniformizes φ.

This concludes the proof of Theorem 9. ◀

5 Uniformizing Metric Formulas

In this section we show decidability of the uniformization problem for $\mathsf{MSO}(<,+1)$ over bounded real time domains.

Note that as an immediate corollary of Theorem 2, we can establish an analogous result over bounded domains.

▶ **Corollary 12.** *Let $\mathbb{T} = [0, N)$ be a bounded interval of reals. Given an* MSO($<$)*-formula $\varphi(\overline{P}, \overline{X}, \overline{Y})$ one can compute a sentence $\theta(\overline{P})$ and formula $\psi(\overline{P}, \overline{X}, \overline{Y})$ such that for every structure $\mathcal{M} = \langle \mathbb{T}, <, \overline{\mathbf{P}} \rangle$, φ has a causal uniformizer over $\mathcal{M}$ if and only if $\mathcal{M} \models \theta$ and in this case ψ is such a causal uniformizer.*

We now seek to apply this result to formulas of MSO($<, +1$) by first removing all references to the $+1$ relation using the following translation.

5.1 Eliminating the Metric

Given an MSO($<, +1$)-formula φ, we define a straightforward syntactic transformation into an MSO($<$)-formula $\overline{\varphi}$ such that there is a natural bijection between models of φ with domain $[0, N)$ and models of $\overline{\varphi}$ with domain $[0, 1)$.

With each monadic predicate X that appears in φ, we associate a collection $X_0, \ldots, X_{N-1}$ of N fresh monadic predicates. Intuitively, each X_i is a predicate on $[0, 1)$ that represents X over the subinterval $[i, i+1)$. Formally, an interpretation of X over domain $[0, N)$ yields interpretations of the X_i over $[0, 1)$ by defining $X_i(t)$ if and only if $X(i + t)$. Note that this correspondence yields a bijection between interpretations of X on $[0, N)$ and interpretations of $X_0, \ldots, X_{N-1}$ on $[0, 1)$.

We can assume that φ does not contain any (first- or second-order) existential quantifiers, by replacing them with combinations of universal quantifiers and negations if need be. It is also convenient to rewrite φ into a formula that makes use of a unary function $+1$ instead of the $+1$ relation. To this end, replace every occurrence of $+1(x, y)$ in φ by $(x < N - 1 \wedge x + 1 = y)$.

Next, replace every instance of $\forall x\, \psi$ in φ by the formula

$$\forall x\, (\psi[x/x] \wedge \psi[x + 1/x] \wedge \ldots \wedge \psi[x + (N-1)/x])\,,$$

where $\psi[t/x]$ denotes the formula resulting from substituting every free occurrence of the variable x in ψ by the term t. Intuitively, this transformation is legitimate since first-order variables in our target formula will range over $[0, 1)$ rather than $[0, N)$.

Having carried out these substitutions, use simple arithmetic to rewrite every term in φ as $x + k$, where x is a variable and $k \in \mathbb{N}$ is a non-negative integer constant.

Every inequality occurring in φ is now of the form $x + k < N - 1$ or $x + k_1 < y + k_2$. Replace every inequality of the first kind by **true** if $k + 2 \leq N$ and by **false** otherwise, and replace every inequality of the second kind by (i) $x < y$, if $k_1 = k_2$; (ii) **true**, if $k_1 < k_2$; and (iii) **false** otherwise.

Every equality occurring in φ is now of the form $x + k_1 = y + k_2$. Replace every such equality by $x = y$ if $k_1 = k_2$, and by **false** otherwise.

Every use of monadic predicates in φ now has the form $X(x + k)$, for $k \leq N - 1$. Replace every such predicate by $X_k(x)$.

Finally, replace every occurrence of $\forall X\, \psi$ in φ by $\forall X_0\, \forall X_1\, \ldots \forall X_{N-1}\, \psi$. The resulting formula is the desired $\overline{\varphi}$. Note that $\overline{\varphi}$ does not mention the $+1$ function, and is therefore indeed a non-metric (i.e., purely order-theoretic) sentence in MSO($<$). The following proposition is then clear.

▶ **Proposition 13** ([9]). *$\langle [0, N), <, +1, \mathbf{P} \rangle \models \varphi$ if and only if $\langle [0, 1), <, \mathbf{P}_0, \ldots, \mathbf{P}_{N-1} \rangle \models \overline{\varphi}$.*

5.2 Main Result

The following result, concerning the computability of uniformizers in MSO($<, +1$), is the main result of the paper. We state this problem for unlabelled intervals, although considering labelled intervals is essential in the proof. Our proof technique generalises straightforwardly to handle a more general result involving uniformisation of $\mathsf{MSO}(<,+1)$ over labelled intervals [6].

▶ **Theorem 3.** *Let $\mathbb{T} = [0, N)$ be a bounded interval of reals. Given an $\mathsf{MSO}(<,+1)$-formula $\varphi(\overline{X}, \overline{Y})$ one can decide whether φ has an $\mathsf{MSO}(<,+1)$-definable causal uniformizer over $\langle \mathbb{T}, <, +1\rangle$ and if so one can compute such a uniformizer $\psi(\overline{X}, \overline{Y})$.*

Proof. To simplify notation, we consider the special case where φ has only X and Y as free variables.

Step 1. Applying the transformation described in Section 5.1 to $\varphi(X, Y)$ yields an $\mathsf{MSO}(<)$-formula $\overline{\varphi}(X_0, \dots, X_{N-1}, Y_0, \dots, Y_{N-1})$, such that there is a natural bijection between the models of φ over $[0, N)$ and the models of $\overline{\varphi}$ over $[0, 1)$, where $X_i(t)$ holds if and only if $X(i + t)$ holds for $i = 0, 1, \dots, N - 1$ and $0 \le t < 1$.

Step 2. We reduce the problem of uniformizing φ over $\langle \mathbb{T}, <, +1\rangle$ to an N-*phase* uniformization procedure applied to $\overline{\varphi}$. In the first phase, we construct a causal operator to determine the values of Y_0 from the values of X_0. The second phase then constructs a causal operator to determine the values of Y_1 from those of X_1, treating the values of X_0 and Y_0 generated in the previous phase as parameters that are already fixed. At the end of the N-th phase we wish $\overline{\varphi}$ to be satisfied by the values of $\overline{X}$ and $\overline{Y}$ we have determined. Our claim is that we can construct a series of functions in such a scenario if and only if we can uniformize φ over $\langle \mathbb{T}, <, +1\rangle$.

We formalise the above idea by defining N uniformization problems $G_0, G_1 \dots, G_{N-1}$ involving only $\mathsf{MSO}(<)$ over $[0, 1)$. The basic data of each problem G_k, $0 \le k < N$ are illustrated in Figure 1.

We define the problems G_k by backward induction, starting with G_{N-1}. The winning condition in this problem is

$$\varphi_{N-1}(X_0, \dots, X_{N-1}, Y_0, \dots, Y_{N-1}) \stackrel{\text{def}}{=} \overline{\varphi}(X_0, \dots, X_{N-1}, Y_0, \dots, Y_{N-1}) \, ,$$

where $X_0, \dots, X_{N-2}$ and $Y_0, \dots, Y_{N-2}$ are considered as parameters and X_{N-1} and Y_{N-1} as variables. Applying Corollary 12 we obtain a domain formula θ_{N-1} and uniformizer ψ_{N-1} for φ_{N-1}.

Suppose that we have defined G_k, with basic data as given in Figure 1. Then we define G_{k-1} as follows. The formula to be uniformized, denoted φ_{k-1}, is defined to be the domain formula θ_k from the preceding problem G_k. In G_{k-1} we consider $X_0, \dots, X_{k-2}$ and $Y_0, \dots, Y_{k-2}$ as parameters and X_{k-1} and Y_{k-1} as variables. The domain formula θ_{k-1} and uniformizer ψ_{k-1} are then obtained by applying Corollary 12 to the problem G_{k-1}.

Notice that the domain formula θ_0 is equivalent to either the sentence **true** or the sentence **false**. Below we show that $\theta_0 \equiv$ **true** just in case $\varphi(X, Y)$ has a uniformizer over $\langle \mathbb{T}, <, +1\rangle$.

Step 3: *Definition of uniformizer $\psi(X, Y)$ for $\varphi(X, Y)$.* Let

$$\overline{\psi} \stackrel{\text{def}}{=} \psi_0 \wedge \cdots \wedge \psi_{N-1} \, .$$

Then $\overline{\psi}$ is a formula in variables $X_0, \dots, X_{N-1}$ and $Y_0, \dots, Y_{N-1}$. The $\mathsf{MSO}(<,+1)$-formula $\psi(X, Y)$ is obtained from $\overline{\psi}$ by replacing every occurrence of $X_i(t)$ with $X(i + t)$ and $Y_i(t)$ with $Y(i+t)$. The transformation from $\overline{\psi}$ to ψ can be seen as the reverse of the transformation in Section 5.1, motivating our choice of notation.

This completes the description of the procedure to decide whether $\varphi(X, Y)$ has a uniformizer and if so to construct such a uniformizer. We turn now to the correctness of this construction.

Uniformization Problem G_k	
Input:	
Winning condition	$\varphi_k(X_0,\dots,X_k,Y_0,\dots,Y_k)$
Parameters	$X_0,\dots,X_{k-1},Y_0,\dots,Y_{k-1}$
Variables	X_k, Y_k
Output:	
Domain formula	$\theta_k(X_0,\dots,X_{k-1},Y_0,\dots,Y_{k-1})$
Uniformizer	$\psi_k(X_0,\dots,X_k,Y_0,\dots,Y_k)$

Figure 1 Basic data for G_k.

Suppose that $\theta_0 \equiv$ **true**. We show that $\psi(X,Y)$ defines a causal uniformizer for $\varphi(\overline{X},\overline{Y})$ on $\langle \mathbb{T},<,+1\rangle$.

Let $\mathbf{X} \subseteq \mathbb{T}$. We must show that there is a unique $\mathbf{Y} \subseteq \mathbb{T}$ such that $\psi(\mathbf{X},\mathbf{Y})$ and for this $\mathbf{Y}$ also $\varphi(\mathbf{X},\mathbf{Y})$. Write $\overline{\mathbf{X}} = \mathbf{X}_0,\dots,\mathbf{X}_{N-1}$ for the tuple of subsets of $[0,1)$ defined by $\mathbf{X}_i(t)$ if and only if $\mathbf{X}(i+t)$.

Since $\theta_0 \equiv$ **true** we know that ψ_0 uniformizes φ_0. Thus there exists $\mathbf{Y}_0 \subseteq [0,1)$ such that $\psi_0(\mathbf{X}_0,\mathbf{Y}_0)$ and $\varphi_0(\mathbf{X}_0,\mathbf{Y}_0)$ both hold. But $\theta_1 \equiv \varphi_0$, so $\theta_1(\mathbf{X}_0,\mathbf{Y}_0)$ also holds. Since ψ_1 uniformizes φ_1 there exists $\mathbf{Y}_1 \subseteq [0,1)$ such that $\psi_1(\mathbf{X}_0,\mathbf{X}_1,\mathbf{Y}_0,\mathbf{Y}_1)$ and $\varphi_1(\mathbf{X}_0,\mathbf{X}_1,\mathbf{Y}_0,\mathbf{Y}_1)$ both hold. Continuing in this vein we successively generate predicates $\mathbf{Y}_0,\dots,\mathbf{Y}_{N-1}$ such that

$$\psi_0(\mathbf{X}_0,\mathbf{Y}_0) \wedge \psi_1(\mathbf{X}_0,\mathbf{X}_1,\mathbf{Y}_0,\mathbf{Y}_1) \wedge \cdots \wedge \psi_{N-1}(\mathbf{X}_0,\dots,\mathbf{X}_{N-1},\mathbf{Y}_0,\dots,\mathbf{Y}_{N-1})\,.$$

Thus by definition of $\overline{\psi}$ we have $\overline{\psi}(\mathbf{X}_0,\dots,\mathbf{X}_{N-1},\mathbf{Y}_0,\dots,\mathbf{Y}_{N-1})$.

Now define $\mathbf{Y} \subseteq \mathbb{T}$ by having $\mathbf{Y}(t+i)$ hold if and only if $\mathbf{Y}_i(t)$ holds for $i=0,\dots,N-1$ and $0 \le t < 1$. Then by definition of ψ we have $\psi(\mathbf{X},\mathbf{Y})$. Furthermore, since ψ_{N-1} uniformizes φ_{N-1} we also have that

$$\varphi_{N-1}(\mathbf{X}_0,\dots,\mathbf{X}_{N-1},\mathbf{Y}_0,\dots,\mathbf{Y}_{N-1})\,.$$

But φ_{N-1} was defined to be $\overline{\varphi}$ and thus by Proposition 13 we have that $\varphi(\mathbf{X},\mathbf{Y})$ also holds.

The fact that ψ is functional and causal can easily be obtained from the corresponding properties of $\psi_0,\dots,\psi_{N-1}$ in the above construction. Reversing the above argument also allows us to deduce that if $\varphi(X,Y)$ has a uniformizer over $\langle \mathbb{T},<,+1\rangle$ then $\theta_0 \equiv$ **true**: given a uniformizer $\psi(X,Y)$ for φ one successively generates uniformizers for φ_{N-1} down to φ_0. ◀

6 Lower Bounds

Define a family of functions $\exp_k : \mathbb{N} \to \mathbb{N}$ by $\exp_0(n) = n$ and $\exp_{k+1}(n) = 2^{\exp_k(n)}$. A function $f : \mathbb{N} \to \mathbb{N}$ is *non-elementary* if it grows faster than any $\exp_k$.

The procedure for uniformizing MSO$(<,+1)$-formulas over bounded time domain $\mathbb{T} = [0,N)$ described in Section 5.2 has non-elementary complexity. This blow-up arises not only from the non-elementary transformation of MSO$(<,+1)$ to automata—repeated application of Corollary 12 leads to an N-fold exponential blow-up.

In this section we give a non-elementary lower bound for the bounded uniformization problem for FO$(<,+1)$ that holds even for formulas of a fixed quantifier alternation depth (for which satisfiability over bounded intervals is elementary[1]). This is proven by reduction from the language emptiness problem for star-free regular expressions. The construction we outline below can also be used to show that uniformization is non-elementary also for the temporal logic MTL for which satisfiability over bounded intervals is EXPSPACE-complete [9]. We have used a related idea to show that the language emptiness problem for alternating timed automata over bounded time domains is non-elementary [7].

A *star-free regular expression* over alphabet Σ is built from the symbols $\emptyset$ and σ, for any $\sigma \in \Sigma$, using the operations of union $(+)$, concatenation $(\cdot)$, and complementation $(\neg)$. Such an expression E denotes a language $L(E) \subseteq \Sigma^*$ which is defined as follows:

$L(\emptyset) = \emptyset$ and $L(\sigma) = \{\sigma\}$;
$L(\neg E) = \Sigma^* \setminus L(E)$;
$L(E + E') = L(E) \cup L(E')$;
$L(E \cdot E') = L(E) \cdot L(E')$.

The *operator depth* $\mathrm{odp}(E)$ of a star-free regular expression E is defined as follows:
$\mathrm{odp}(\emptyset) = \mathrm{odp}(\sigma) = 1$;
$\mathrm{odp}(\neg E) = \mathrm{odp}(E)$;
$\mathrm{odp}(E + E') = \max\{\mathrm{odp}(E), \mathrm{odp}(E')\} + 1$;
$\mathrm{odp}(E \cdot E') = \max\{\mathrm{odp}(E), \mathrm{odp}(E')\} + 1$.
Note that negation does not count toward the operator depth.

The following result was shown in [14].

▶ **Theorem 14.** *The language emptiness problem for star-free regular expressions is non-elementary.*

Given a star-free regular expression E over alphabet Σ and a word $w = w_0 w_1 \dots w_{n-1} \in \Sigma^*$ we define the *membership game* $\mathbb{G}(w, E)$. This is a two-player game with N rounds, where N is the operator depth of E. The two players are *Prover*, who is trying to show $w \in E$, and *Refuter*, who is trying to show $w \notin E$. The positions of the game are triples (b, e, F) where b and e are positions in the word w and F has the form G or $\neg G$ for G a sub-expression of E. The initial position is $(0, n, E)$. If the position at the start of a given round is (b, e, F) the goal of Prover is to show that $w_b w_{b+1} \dots w_{e-1} \in F$. The round proceeds as follows:

- If $F \equiv F_1 \cdot F_2$ then Prover moves first by choosing an index i with $b \leq i \leq e$. Refuter responds by selecting either (b, i, F_1) or (i, e, F_2) as the position in the next round;
- If $F \equiv \neg(F_1 \cdot F_2)$ then Refuter moves first by choosing an index i with $b \leq i \leq e$. Prover responds by selecting either $(b, i, \neg F_1)$ or $(i, e, \neg F_2)$ as the position in the next round;
- If $F \equiv F_1 + F_2$ then Prover selects either (b, e, F_1) or (b, e, F_2) as the position in the next round;
- If $F \equiv \neg(F_1 + F_2)$ then Refuter selects either $(b, e, \neg F_1)$ or $(b, e, \neg F_2)$ as the position in the next round.

The positions (b, e, σ), $(b, e, \neg\sigma)$, $(b, e, \emptyset)$ and $(b, e, \neg\emptyset)$ are terminal and are classified as winning for Prover or Refuter according to whether $u_b, u_{b+1} \dots, u_{e-1}$ is a member of the corresponding expression.

It is clear that Prover has a winning strategy in $\mathbb{G}(w, E)$ if and only if $w \in L(E)$.

For any regular expression F, let $\mathit{Sub}(F)$ be the set of sub-expressions of E along with their negations. Given that a position in $\mathbb{G}(w, E)$ is a triple from the set $\Pi \stackrel{\text{def}}{=} \{0, \dots, n\}^2 \times \mathit{Sub}(E)$,

[1] The paper [9] provides an elementary reduction of the satisfiability problem for $\mathsf{FO}(<, +1)$ over bounded intervals to the problem for $\mathsf{FO}(<)$ which does not change the quantifier alternation depth.

a *play* of $\mathbb{G}(w,E)$ can be represented as a word in Π^* denoting a sequence of successive positions. The idea of our reduction is to encode plays as signals over a domain $[0,N+1)$ and to construct a formula of $\mathsf{FO}(<,+1)$ that is satisfied by a signal if and only it encodes a winning play for Prover. In this encoding successive game positions are encoded in successive unit-length subintervals of the domain.

Our encoding represents plays using monadic predicates P_b, P_e, $P_\#$, $P_{S_{Pr}}$, $P_{L_{Pr}}$, $P_{R_{Pr}}$, $P_{S_{Rf}}$, $P_{L_{Rf}}$, $P_{R_{Rf}}$ and two families of predicates P_σ, $\sigma \in \Sigma$ and P_F, $F \in \mathit{Sub}(E)$. For a signal to encode a play of $\mathbb{G}(w,E)$ we require, among other things, that:

- The predicates P_F, $F \in \mathit{Sub}(E)$, hold on intervals $[k,k+1)$ for $k=0,1,\ldots,N$ and are mutually exclusive.
- Exactly one of the predicates P_σ, $\sigma \in \Sigma$, and $P_\#$ holds at any given point. Moreover these predicates hold in sequence $P_{w_0}, P_{w_1}, \ldots, P_{w_{n-1}}, P_\#$ over the interval $[0,1)$.
- If $s=t+1$ then P_σ holds at s if and only if P_σ holds at t; likewise $P_\#$ holds at s if and only if $P_\#$ holds at t.
- In each successive unit interval $[k,k+1)$ the predicate P_b holds in one sub-interval over which some predicate P_σ or $P_\#$ also holds. The same restriction applies to P_e.
- If $P_{E_1 \cdot E_2}$ holds on $[k,k+1)$ then $P_{S_{Pr}}$ holds in exactly one sub-interval in this time unit and either $P_{L_{Rf}}$ holds at the same time as $P_\#$ during this time unit and P_{E_1} holds in the next time unit, or $P_{R_{Rf}}$ holds at the same time as $P_\#$ during this time unit and P_{E_2} holds in the next time unit.

Notice how the third clause ensures that a copy of the word w is propagated between successive time units, cf. Figure 2.

A game position (i,j,F) is encoded in a unit-length subinterval $[k,k+1)$ by having P_F hold throughout the interval, P_b hold at the same time as P_{w_i} and P_e hold at the same time as P_{w_j} (where we take $w_n=\#$). The idea is that the propositions $P_{S_{Pr}}$, $P_{L_{Pr}}$ and $P_{R_{Pr}}$ encode moves of Prover and the propositions $P_{S_{Rf}}$, $P_{L_{Rf}}$ and $P_{R_{Rf}}$ encodes moves of Refuter; the respective position of these S propositions indicate the position around which the input word is split while the L and R propositions indicate whether that player wished to continue playing in the left or right subword.

Given a star-free regular expression E we can define a formula $\varphi_E(\overline{X},\overline{Y})$ such that φ_E has a uniformizer if and only if there exists a word $w \in \Sigma^*$ such that Prover has a winning strategy in the game $\mathbb{G}(w,E)$. The tuple $\overline{X}$ just includes the predicates $P_{S_{Rf}}$, $P_{L_{Rf}}$ and $P_{R_{Rf}}$ while the tuple $\overline{Y}$ includes all the other predicates mentioned above. We define φ_E such that it is true on any signal that represents a play of $\mathbb{G}(w,E)$ that is winning for Prover according to the encoding defined above. For signals that do not encode such plays φ_E is only satisfied if the predicates $P_{S_{Rf}}$, $P_{L_{Rf}}$ or $P_{R_{Rf}}$ do not obey the above rules (intuitively Refuter broke the rules of the game). Details of this encoding can be found in [6].

▶ **Theorem 15.** *The time-bounded uniformization problem for* $\mathsf{FO}(<,+1)$ *is non-elementary for formulas of quantifier alternation depth at most three.*

7 Conclusion

In this paper, we considered extensions of Church's synthesis problem to the continuous-time domain of the reals. We proved that under the finite-variability and right-continuous assumption, Church's problem is decidable when we require that the uniformizer be definable in the same logic as the specification. This result holds over unbounded intervals when only the $<$ relation is available and over every fixed bounded-length interval when the $+1$ relation is also used.

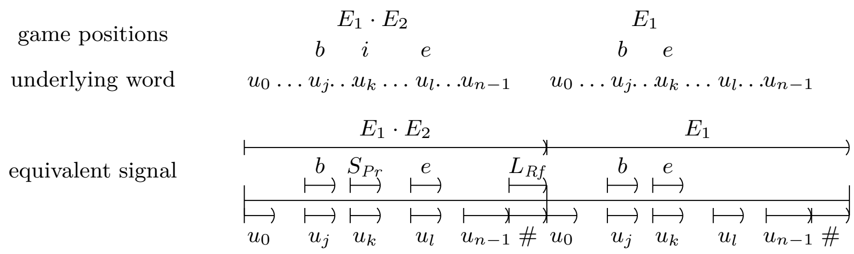

Figure 2 A signal encoding a play

References

1 J. R. Büchi. On a decision method in restricted second-order arithmetic. In *Proceedings of the 1960 International Congress on Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford University Press, 1962.

2 J. R. Büchi and L. H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the AMS*, 138(27):295–311, 1969.

3 A. Church. Applications of recursive arithmetic to the problem of circuit synthesis. In *Summaries of the Summer Institute of Symbolic Logic*, volume 1, pages 3–50. Cornel Univ, Ithaca, 1957.

4 P. Hänsch, M. Slaats, and W. Thomas. Parametrized regular infinite games and higher-order pushdown strategies. In *Proceedings of FCT 09*, volume 5699 of *Lecture Notes in Computer Science*. Springer, 2009.

5 Y. Hirshfeld and A. Rabinovich. Logics for real time: Decidability and complexity. *Fundam. Inform.*, 62(1), 2004.

6 M. Jenkins, J. Ouaknine, A. Rabinovich, and J. Worrell. The Church synthesis problem with metric (full version).
`http://www.cs.ox.ac.uk/people/mark.jenkins/church-metric.pdf`.

7 M. Jenkins, J. Ouaknine, A. Rabinovich, and J. Worrell. Alternating timed automata over bounded time. In *LICS*, pages 60–69. IEEE Computer Society, 2010.

8 S. Lifsches and S. Shelah. Uniformization and Skolem functions in the class of trees. *J. Symb. Log.*, 63(1):103–127, 1998.

9 J. Ouaknine, A. Rabinovich, and J. Worrell. Time-bounded verification. In *Proceedings of CONCUR 09*, volume 5710 of *Lecture Notes in Computer Science*. Springer, 2009.

10 A. Rabinovich. Finite variability interpretation of monadic logic of order. *Theoretical Computer Science*, 275(1-2):111–125, 2002.

11 A. Rabinovich. The Church synthesis problem with parameters. *Logical Methods in Computer Science*, 3(4), 2007.

12 A. Rabinovich. On decidability of monadic logic of order over the naturals extended by monadic predicates. *Inf. Comput.*, 205(6):870–889, 2007.

13 S. Shelah. The monadic theory of order. *The Annals of Mathematics*, 102(3):379–419, 1975.

14 L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. In *Proceedings of STOC 73*, pages 1–9, New York, NY, USA, 1973. ACM.