The Church Problem for expansions of $(\mathbb{N}, <)$ by unary predicates

Alexander Rabinovich

The Blavatnik School of Computer Science
Tel Aviv University, Tel Aviv, Israel 69978

Abstract

For a two-variable formula B(X,Y) of Monadic Logic of Order (MLO) the Church Synthesis Problem concerns the existence and construction of a finite-state operator Y=F(X) such that B(X,F(X)) is universally valid over Nat.

Büchi and Landweber (1969) proved that the Church synthesis problem is decidable.

We investigate a parameterized version of the Church synthesis problem. In this extended version a formula B and a finite-state operator F might contain as a parameter a unary predicate P.

A large class of predicates P is exhibited such that the Church problem with the parameter P is decidable.

Our proofs use Composition Method and game theoretical techniques.

1. Introduction

Two fundamental results of classical automata theory are decidability of the monadic second-order logic of order (MLO) over $\omega = (\mathbb{N}, <)$ and computability of the Church synthesis problem. These results have provided the underlying mathematical framework for the development of formalisms for the description of interactive systems and their desired properties, the algorithmic verification and the automatic synthesis of correct implementations from logical specifications, and advanced algorithmic techniques that are now embodied in industrial tools for verification and validation.

1.1. Decidable Expansions of ω

Büchi [1] proved that the monadic theory of $\omega = (\mathbb{N}, <)$ is decidable. Even before the decidability of the monadic theory of ω has been proved, it was shown that the expansions of ω by "interesting" functions have undecidable monadic theory. In particular, the monadic theory of $(\mathbb{N}, <, +)$ and the monadic theory of $(\mathbb{N}, <, \lambda x.2 \times x)$ are undecidable [15, 20]. Therefore, most efforts to find decidable expansions of ω deal with expansions of ω by monadic predicates.

Elgot and Rabin [5] found many interesting predicates P for which MLO over $(\mathbb{N}, <, P)$ is decidable. Among these predicates are the set of factorial numbers $\{n! \mid n \in \mathbb{N}\}$, the sets of k-th powers $\{n^k \mid n \in \mathbb{N}\}$ and the sets $\{k^n \mid n \in \mathbb{N}\}$ (for $k \in \mathbb{N}$).

The Elgot and Rabin method has been generalized and sharpened over the years and their results were extended to a variety of unary predicates (see e.g., [18, 16, 3]). In [11, 14] we provided necessary and sufficient conditions for the decidability of monadic (second-order) theory of expansions of the linear order of the naturals ω by unary predicates.

Email address: `rabinoa@post.tau.ac.il` (Alexander Rabinovich)

1.2. Church's Problem

What is known as the "Church synthesis problem" was first posed by A. Church in [4] for the case of $(\omega, <)$. The Church problem is much more complicated than the decidability problem for MLO. Church uses the language of automata theory. It was McNaughton (see [9]) who first observed that the Church problem can be equivalently phrased in game-theoretic language and in recent years many authors took up the generalizations of such games for various applications of the algorithmic theory of infinite games (see e.g., [6, 10, 21]). McNaughton considered games over ω. We consider such games over expansions of ω by unary predicates.

Let $\mathcal{M} = (\mathbb{N}, <, P)$ be the expansion of ω by a unary predicate P. Let $\varphi(X_1, X_2, Z)$ be a formula, where X_1, X_2 and Z are set (monadic predicate) variables. The *McNaughton game* $\mathcal{G}_\varphi^\mathcal{M}$ is defined as follows.

1. The game is played by two players, called Player I and Player II.
2. A *play* of the game has ω rounds.
3. At round $i \in \mathbb{N}$: first, Player I chooses $\rho_{X_1}(i) \in \{0, 1\}$; then, Player II chooses $\rho_{X_2}(i) \in \{0, 1\}$. Both players can observe whether $i \in P$.
4. By the end of the play two predicates $\rho_{X_1}, \rho_{X_2} \subseteq \mathbb{N}$ have been constructed[1]
5. Then, Player I wins the play if $\mathcal{M} \models \varphi(\rho_{X_1}, \rho_{X_2}, P)$; otherwise, Player II wins the play.

What we want to know is: Does either one of the players have a *winning strategy* in $\mathcal{G}_\varphi^\mathcal{M}$? If so, which one? That is, can Player I choose his moves so that, whatever way Player II responds we have $\varphi(\rho_{X_1}, \rho_{X_2}, P)$? Or can Player II respond to Player I's moves in a way that ensures the opposite?

At round i, Player I has access only to $\rho_{X_1}(0) \dots \rho_{X_1}(i-1)$, $\rho_{X_2}(0) \dots \rho_{X_2}(i-1)$ and $P(0) \dots P(i)$.

Hence, a strategy of Player I can be defined as a function which assigns to any finite sequence

$$(\rho_{X_1}(0), \rho_{X_2}(0), P(0)) \dots (\rho_{X_1}(i-1), \rho_{X_2}(i-1), P(i-1)) \, (*, *, P(i))$$

a value in $\{0, 1\}$ which is taken to be $\rho_{X_1}(i)$. (Equivalently, a strategy of Player I in $\mathcal{G}_\varphi^\mathcal{M}$ can be defined as a function which assigns to any finite sequence $\rho_{X_2}(0), \dots, \rho_{X_2}(i-1)$ of moves of Player II the i-th move of Player I. However, information about Player I previous moves is convenient for description of strategies by formulas, and information about previous values of P will be essential for the definition of finite-memory strategies.)

At round i, Player II has access only to $\rho_{X_1}(0) \dots \rho_{X_1}(i)$, $\rho_{X_2}(0) \dots \rho_{X_2}(i-1)$ and $P(0) \dots P(i)$.

Hence, a strategy of Player II can be defined as a function which assigns to any finite sequence

$$(\rho_{X_1}(0), \rho_{X_2}(0), P(0)) \dots (\rho_{X_1}(i-1), \rho_{X_2}(i-1), P(i-1)) \, (\rho_{X_1}(i), *, P(i))$$

a value in $\{0, 1\}$ which is taken to be $\rho_{X_2}(i)$.

Since strategies are functions from finite strings (over a finite alphabet) to $\{0, 1\}$ we can classify them according to their complexity. The recursive strategies, the finite-memory strategies, i.e., the strategies computable by finite-state transducers are defined in a natural way (see Sect. 3).

We investigate the following parameterized version of the Church synthesis problem.

Synthesis Problems for $\mathcal{M} = (\mathbb{N}, <, P)$, where $P \subseteq \mathbb{N}$

Input: an MLO formula $\varphi(X_1, X_2, Z)$.
Task: Check whether Player I has a finite-memory winning strategy in $\mathcal{G}_\varphi^\mathcal{M}$ and if there is such a strategy - construct it.

[1] We identify monadic predicates with their characteristic functions.

To simplify notations, games and the synthesis problem were previously defined for formulas with three free variables X_1, X_2 and Z. It is easy to generalize all definitions and results to formulas $\psi(X_1,\dots,X_m,Y_1,\dots Y_n,Z_1,\dots,Z_l)$ with many variables. In this generalization at round β, Player I chooses values for $X_1(\beta),\dots,X_m(\beta)$, then Player II replies by choosing the values to $Y_1(\beta),\dots,Y_n(\beta)$ and the structure $\mathcal{M}$ provides the interpretation for $Z_1,\dots Z_l$. Note that, strictly speaking, the input to the synthesis problem is not only a formula, but a formula plus a partition of its free-variables to Player I's variables and Player II's variables and parameter's variables.

In [2], Büchi and Landweber prove the computability of the synthesis problem in $\omega = (\mathbb{N}, <)$ (no parameters).

Theorem 1.1 (Büchi-Landweber, 1969). *Let $\varphi(\bar{X},\bar{Y})$ be a formula, where $\bar{X}$ and $\bar{Y}$ are disjoint lists of variables. Then:*

Determinacy: *One of the players has a winning strategy in the game $\mathcal{G}^{\omega}_{\varphi}$.*

Decidability: *It is decidable* which *of the players has a winning strategy.*

Finite-state strategy: *The player who has a winning strategy, also has a finite-state winning strategy.*

Synthesis algorithm: *We can compute for the winning player in $\mathcal{G}^{\omega}_{\varphi}$ a finite-state winning strategy.*

The determinacy part of the theorem follows from topological arguments. In particular for every expansion $\mathcal{M}$ of ω by unary predicates, the game $\mathcal{G}^{\mathcal{M}}_{\varphi}$ is determined.

Let $\mathcal{M}$ be an expansion of ω by unary predicates. We proved in [12], that there is an algorithm which for every *MLO* formula φ decides who wins $\mathcal{G}^{\mathcal{M}}_{\varphi}$ if and only if the monadic theory of $\mathcal{M}$ is decidable. Moreover, we proved that if the monadic theory of $\mathcal{M}$ is decidable, then the player who has a winning strategy in $\mathcal{G}^{\mathcal{M}}_{\varphi}$ has a recursive *MLO*-definable winning strategy which is computable from φ.

The finite-state strategy part of Theorem 1.1 fails for decidable expansions of ω. For example, let $\mathbf{Fac} = \{n! \mid n \in \mathbb{N}\}$ be the set of factorial numbers. The monadic theory of $\mathcal{M}_{fac} := (\mathbb{N}, <, \mathbf{Fac})$ is decidable by [5]. Let $\varphi(X_1, X_2, Z)$ be a formula which specifies that $t \in X_1$ iff $t + 1 \in Z$ (hence for the game $\mathcal{G}^{\mathcal{M}_{fac}}_{\varphi}$ the moves of Player II are irrelevant). It is easy to see that Player I has a winning strategy in $\mathcal{G}^{\mathcal{M}_{fac}}_{\varphi}$, yet Player I has no finite-state winning strategy in this game. The results of this paper imply that the synthesis problem for $(\mathbb{N}, <, \mathbf{Fac})$ is decidable.

1.3. Main Result

Our main result describes a large class of predicates P such that the synthesis problem for $(\mathbb{N}, <, P)$ is decidable.

An ω-sequence a_i is said to be ultimately periodic with lag l and period d if $a_i = a_{i+d}$ for $i > l$.

Definition 1.2. *Let $\bar{k} = (k_1 < k_2 < \dots k_i < \dots)$ be an increasing ω-sequence of integers.*

1. *$\bar{k}$ is* sparse *if for each d there is n such that $k_{i+1} - k_i > d$ for each $i > n$.*

   $\bar{k}$ is effectively sparse *if there is an algorithm that for each d computes n such that $k_{i+1} - k_i > d$ for each $i > n$.*

2. *$\bar{k}$ is* ultimately reducible *if for every $m > 1$ the sequence k_i mod m is ultimately periodic. $\bar{k}$ is* effectively ultimately reducible *if there is an algorithm that for each m computes a lag and a period of k_i mod m.*

The next definition introduces a generalization of ω-sequences considered by Elgot and Rabin in [5].

Definition 1.3. *Let ER be the class of increasing recursive ω-sequences of integers which are effectively sparse and effectively ultimately reducible.*

Let $P \subseteq \mathbb{N}$ be a predicate. We denote by $Enum(P)$ the sequence $(k_1, k_2 \dots k_i \dots)$ which enumerates the elements of P in the increasing order. Often we do not distinguish between P and $Enum(P)$, In particular we say that a predicate is ER predicate if $Enum(P)$ is in ER. The class ER contains many interesting predicates. It contains the set Fact=$\{n! \mid n \in \mathbb{N}\}$ of factorial numbers, the sets $\{k^n \mid n \in \mathbb{N}\}$, the sets $\{n^k \mid n \in \mathbb{N}\}$. It has nice closure properties, e.g., if $\bar{k}$ and $\bar{l}$ are in ER then $\{k_i + l_i \mid i \in \mathbb{N}\}$, $\{k_i \times l_i \mid i \in \mathbb{N}\}$, and $\{k_i^{l_i} \mid i \in \mathbb{N}\}$ are in ER.

In [18], Siefkes introduced ER predicates and generalized Elgot-Rabin contraction method to prove that for every ER predicate P the monadic theory of $\mathcal{M} = (\mathbb{N}, <, P)$ is decidable. Our main results show that the synthesis problem for each predicate $P \in ER$ is decidable.

Theorem 1.4 (Main). *Let P be an ER predicate and let $\mathcal{M} = (\mathbb{N}, <, P)$. There is an algorithm that for every MLO formula $\varphi(X_1, X_2, Z)$ decides whether Player I or Player II has a finite-memory winning strategy in $\mathcal{G}_\varphi^{\mathcal{M}}$, and if so constructs such a strategy.*

Our algorithm is based on game theoretical techniques and the composition method developed by Feferman-Vaught, Shelah and others.

1.4. Organization of the paper

The article is organized as follows. The next section recalls standard definitions about the monadic second-order logic of order, and summarizes elements of the composition method.

In Section 3, we introduce game-types, define games on game types and show that these game are reducible to the McNaughton games.

Section 4 consider games over finite chains. Sufficient conditions are provided for existence of a finite state strategies which uniformly wins over a class of finite chains.

Section 5 describes an algorithm for the synthesis problem over the expansions of ω by ER predicates, and proves the soundness of the algorithm, i.e., if the algorithm outputs a strategy for $\mathcal{G}_\varphi^{\mathcal{M}}$, then it is a finite state strategy which wins φ over $\mathcal{M}$. In Section 6 we prove the completeness of our algorithm: if a player has a finite state winning strategy in $\mathcal{G}_\varphi^{\mathcal{M}}$, then the algorithm will find such a strategy.

In Section 7 we consider strategies with look-ahead. A strategy with a look-ahead h at i-th round can observe whether $i + h \in P$. We show determinacy of McNaughton games over ER predicates by finite-memory strategies with look-ahead, i.e., for such games one of the players has a winning finite-memory strategy with look-ahead. The proofs in Section 7 relies on the definability results in [12], and are entirely independent from our proof of the computability of finite-memory synthesis problem. To understand these proofs, the reader should only familiarize himself/herself with the notations and definitions of Section 2.

Further results and open questions are discussed in Section 8.

An extended abstract of this paper was published in [13].

2. Preliminaries and Background

We use i, j, n, k, l, m, p, q for natural numbers. We use $\mathbb{N}$ for the set of natural numbers and ω for the first infinite ordinal. We use the expressions "*chain*" and "*linear order*" interchangeably. A chain with m elements will be denoted by m.

We use $\mathbb{P}(A)$ for the set of subsets of A.

2.1. The Monadic Logic of Order (MLO)

2.1.1. Syntax

The syntax of the monadic second-order logic of order - *MLO* has in its vocabulary *individual* (first order) variables $t_1, t_2 \dots$, monadic *second-order* variables $X_1, X_2 \dots$ and one binary relation $<$ (the order).

Atomic formulas are of the form $X(t)$ and $t_1 < t_2$. Well formed formulas of the monadic logic *MLO* are obtained from atomic formulas using Boolean connectives $\neg, \vee, \wedge, \rightarrow$ and the first-order quantifiers $\exists t$ and $\forall t$, and the second-order quantifiers $\exists X$ and $\forall X$. The quantifier depth of a formula φ is denoted by $\mathrm{qd}(\varphi)$.

We use upper case letters X, Y, Z,... to denote second-order variables; with an overline, $\bar{X}$, $\bar{Y}$, etc., to denote finite tuples of variables.

2.1.2. Semantics

A *structure* is a tuple $\mathcal{M} := (A, <^{\mathcal{M}}, \bar{P}^{\mathcal{M}})$ where: A is a non-empty set, $<^{\mathcal{M}}$ is a binary relation on A, and $\bar{P}^{\mathcal{M}} := \left(P_1^{\mathcal{M}}, \dots, P_l^{\mathcal{M}}\right)$ is a *finite* tuple of sub*sets* of A.

If $\bar{P}^{\mathcal{M}}$ is a tuple of l sets, we call $\mathcal{M}$ an *l-structure*. If $<^{\mathcal{M}}$ linearly orders A, we call $\mathcal{M}$ an *l-chain*. When the specific l is unimportant, we simply say that $\mathcal{M}$ is a *labeled* chain.

Suppose $\mathcal{M}$ is an l-structure and φ a formula with free-variables among $X_1, \dots, X_l$. We define the relation $\mathcal{M} \models \varphi$ (read: $\mathcal{M}$ *satisfies* φ) as usual, understanding that the second-order quantifiers range over *subsets* of A.

Let $\mathcal{M}$ be an l-structure. The *monadic theory* of $\mathcal{M}$, $MTh(\mathcal{M})$, is the set of all formulas with free-variables among $X_1, \dots, X_l$ satisfied by $\mathcal{M}$.

From now on, we omit the superscript in '$<^{\mathcal{M}}$' and '$\bar{P}^{\mathcal{M}}$'. We often write $(A, <) \models \varphi(\bar{P})$ meaning $(A, <, \bar{P}) \models \varphi$.

For a chain $\mathcal{M} = (A, <, \bar{P})$ and a subset I of A, we denote by $\mathcal{M} \restriction I$ the subchain of $\mathcal{M}$ over the set I.

2.2. Elements of the composition method

Our proofs make use of the technique known as the composition method developed by Feferman-Vaught and Shelah [8, 17]. To fix notations and to aid the reader unfamiliar with this technique, we briefly review the definitions and results that we require. A more detailed presentation can be found in [19] or [7].

Let $n, l \in \mathbb{N}$. We denote by $\mathfrak{Form}_l^n$ the set of *MLO* formulas with free variables among $X_1, \dots, X_l$ and of quantifier depth $\leq n$.

Definition 2.1. *Let $n, l \in \mathbb{N}$ and let $\mathcal{M}, \mathcal{N}$ be l-structures. The* n-theory *of $\mathcal{M}$ is $Th^n(\mathcal{M}) := \{\varphi \in \mathfrak{Form}_l^n \mid \mathcal{M} \models \varphi\}$. If $Th^n(\mathcal{M}) = Th^n(\mathcal{N})$, we say that $\mathcal{M}$ and $\mathcal{N}$ are* n-equivalent *and write $\mathcal{M} \equiv^n \mathcal{N}$.*

Clearly, $\equiv^n$ is an equivalence relation. For any $n \in \mathbb{N}$ and $l > 0$, the set $\mathfrak{Form}_l^n$ is infinite. However, it contains only finitely many semantically distinct formulas. So, there are finitely many $\equiv^n$-equivalence classes of l-structures. In fact, we can compute characteristic formulas for the $\equiv^n$-equivalence classes:

Lemma 2.2 (Hintikka Lemma). *For $n, l \in \mathbb{N}$, we can compute a* finite *set $Char_l^n \subseteq \mathfrak{Form}_l^n$ such that:*

- *For every $\equiv^n$-equivalence class C there is a* unique *$\tau \in Char_l^n$ such that for every l-structure $\mathcal{M}$: $\mathcal{M} \in C$ iff $\mathcal{M} \models \tau$.*
- *Every MLO formula $\varphi(X_1, \dots X_l)$ with $\mathrm{qd}(\varphi) \leq n$ is equivalent to a (finite) disjunction of characteristic formulas from $Char_l^n$. Moreover, there is an algorithm which for every formula $\varphi(X_1, \dots X_l)$ computes a finite set $G \subseteq Char_l^{\mathrm{qd}(\varphi)}$ of characteristic formulas, such that φ is equivalent to the disjunction of all the formulas from G.*

*Any member of $Char_l^n$ we call a (n, l)-*Hintikka formula *or (n, l)-characteristic formula. We use τ, τ_i, τ^j to range over the characteristic formulas and G, G_i, G' to range over sets of characteristic formulas.*

Definition 2.3 (n-Type). *For $n, l \in \mathbb{N}$ and an l-structure $\mathcal{M}$, we denote by $type_n(\mathcal{M})$ the unique member of $Char_l^n$ satisfied by $\mathcal{M}$ and call it the* n-type *of $\mathcal{M}$.*

Thus, $type_n(\mathcal{M})$ determines $Th^n(\mathcal{M})$ and, indeed, $Th^n(\mathcal{M})$ is computable from $type_n(\mathcal{M})$.

Definition 2.4 (Sum of chains). (1) *Let $l \in \mathbb{N}$, $\mathcal{I} := (I, <^{\mathcal{I}})$ a chain and $\mathfrak{S} := (\mathcal{M}_\alpha \mid \alpha \in I)$ a sequence of l-chains. Write $\mathcal{M}_\alpha := (A_\alpha, <^\alpha, {P_1}^\alpha, \dots, {P_l}^\alpha)$ and assume $A_\alpha \cap A_\beta = \emptyset$ whenever $\alpha \neq \beta$ are in I. The* ordered sum *of $\mathfrak{S}$ is the l-chain*

$$\sum_{\mathcal{I}} \mathfrak{S} := (\bigcup_{\alpha \in I} A_\alpha, <^{\mathcal{I},\mathfrak{S}}, \bigcup\nolimits_{\alpha \in I} {P_1}^\alpha, \dots, \bigcup_{\alpha \in I} {P_l}^\alpha), \text{ where}$$

if $\alpha, \beta \in I$, $a \in A_\alpha$, $b \in A_\beta$, then $b <^{\mathcal{I},\mathfrak{S}} a$ iff $\beta <^{\mathcal{I}} \alpha$ or $\beta = \alpha$ and $b <^\alpha a$.

If the domains of the $\mathcal{M}_\alpha$'s are not disjoint, replace them with isomorphic l-chains that have disjoint domains, and proceed as before.

(2) *If for all $\alpha \in I$, $\mathcal{M}_\alpha$ is isomorphic to $\mathcal{M}$ for some* fixed *$\mathcal{M}$, we denote $\sum_{\mathcal{I}} \mathfrak{S}$ by $\mathcal{M} \times \mathcal{I}$.*

(3) *If $\mathcal{I} = (\{0,1\}, <)$ and $\mathfrak{S} = (\mathcal{M}_0, \mathcal{M}_1)$, we denote $\sum_{\mathcal{I}} \mathfrak{S}$ by $\mathcal{M}_0 + \mathcal{M}_1$.*

We will use only special cases of this definition in which the index chain $\mathcal{I}$ and the summand chains $\mathcal{M}_\alpha$ are finite or of the order type ω.

The next proposition says that taking ordered sums preserves $\equiv^n$-equivalence.

Proposition 2.5. *Let $n, l \in \mathbb{N}$. Assume:*

1. *$(I, <^{\mathcal{I}})$ is a linear order,*
2. *$\left(\mathcal{M}^0_\alpha \mid \alpha \in I\right)$ and $\left(\mathcal{M}^1_\alpha \mid \alpha \in I\right)$ are sequences of l-chains, and*
3. *for every $\alpha \in I$, $\mathcal{M}^0_\alpha \equiv^n \mathcal{M}^1_\alpha$.*

Then, $\sum_{\alpha \in I} \mathcal{M}^0_\alpha \equiv^n \sum_{\alpha \in I} \mathcal{M}^1_\alpha$.

This allows us to define the sum of formulas in $Char^n_l$ with respect to any linear order.

Definition 2.6 (Sum of types). (1) *Let $n, l \in \mathbb{N}$, $\mathcal{I} := (I, <^{\mathcal{I}})$ a chain, $\mathfrak{H} := (\tau_\alpha \mid \alpha \in I)$ a sequence of (n,l)-Hintikka formulas. The* ordered sum *of $\mathfrak{H}$, (notations $\sum_{\mathcal{I}} \mathfrak{H}$ or $\sum_{\alpha \in \mathcal{I}} \tau_\alpha$), is* an *element τ of $Char^n_l$ such that:*

if $\mathfrak{S} := (\mathcal{M}_\alpha \mid \alpha \in I)$ is a sequence of l-chains and $type_n(\mathcal{M}_\alpha) = \tau_\alpha$ for $\alpha \in I$, then

$$type_n(\sum_{\mathcal{I}} \mathfrak{S}) = \tau.$$

(2) *If for all $\alpha \in I$, $\tau_\alpha = \tau$ for some* fixed *$\tau \in Char^n_l$, we denote $\sum_{\alpha \in \mathcal{I}} \tau_\alpha$ by $\tau \times \mathcal{I}$.*

(3) *If $\mathcal{I} = (\{0,1\}, <)$ and $\mathfrak{H} = (\tau_0, \tau_1)$, we denote $\sum_{\alpha \in \mathcal{I}} \tau_\alpha$ by $\tau_0 + \tau_1$.*

The following fundamental result of Shelah can be found in [17]:

Theorem 2.7 (Composition Theorem). *Let $\varphi(X_1, \ldots, X_l)$ be a formula, let $n = \mathrm{qd}(\varphi)$ and let $\{\tau_1, \ldots, \tau_m\} = Char^n_l$. Then, there is a formula $\psi(Y_1, \ldots, Y_m)$ such that for every chain $\mathcal{I} = (I, <^{\mathcal{I}})$ and every sequence $(\mathcal{M}_\alpha \mid \alpha \in I)$ of l-chains the following holds:*

$$\sum_{\alpha \in I} \mathcal{M}_\alpha \models \varphi \text{ iff } \mathcal{I} \models \psi(Q_1, \ldots Q_m), \text{ where}$$

$Q_j = \{\alpha \in I \mid M_\alpha \models \tau_j\}$. Moreover, ψ is computable from φ.

The next Theorem is an important consequence of the Composition Theorem:

Theorem 2.8 (Addition Theorem). *The function which maps the pairs of characteristic formulas to their sum is a recursive function. Formally, the function $\lambda n, l \in \mathbb{N}. \lambda \tau_0, \tau_1 \in Char^n_l . \tau_0 + \tau_1$ is recursive.*

We often use the following well-known lemmas (see e.g., [7]):

Lemma 2.9. *For every $n \in \mathbb{N}$ there is $N_0(n)$ such that for every sentence φ of quantifier depth at most n and every $m \geq N_0$:*

φ is satisfiable over the m-element chain iff it is satisfiable over the $m + N_0$-element chain, i.e., $m \equiv^n m + N_0$.

Furthermore, N_0 is computable from n.

Lemma 2.10. *For every $n \in \mathbb{N}$ there is $N_1(n)$ such that for every $\mathcal{M} = (A, <, P)$: if $n_1 > n_2 \geq N_1$ and $n_1 = n_2$ mod N_1, then $\mathcal{M} \times n_1 \equiv^n \mathcal{M} \times n_2$. Moreover, N_1 is computable from n.*

3. Game types

In this section we introduce game-types; their role for games is similar to the role of types for *MLO*. We define games on game types and show that these games are reducible to McNaughton games. But first we introduce a terminology, define finite-memory strategies and fix some notational conventions.

Let $\mathcal{M} := (\mathbb{N}, <, \bar{P})$ be an l-chain and let $\rho := (\rho_{X_1}(0), \rho_{X_2}(0)) \dots (\rho_{X_1}(i), \rho_{X_2}(i)) \dots$ be a play. We denote by $\mathcal{M}^\frown\rho$ the expansion of $\mathcal{M}$ by the predicates ρ_{X_1} and ρ_{X_2}. We say that the m-type of ρ is τ if $\tau = \mathit{type}_m(\mathcal{M}^\frown\rho)$. Whenever $\mathcal{M}$ is clear from the context we write $\mathit{type}_m(\rho)$ for $\mathit{type}_m(\mathcal{M}^\frown\rho)$.

A strategy for Player I for games over l-chains is a transducer which consists of a set Q - memory states, an initial state q_{init}, the memory update functions $\mu_1 : Q \times \{0,1\}^l \to Q$ and $\mu_2 : Q \times \{0,1\} \to Q$, and the output function $\theta : Q \to \{0,1\}$.

A strategy is finite-memory (or finite-state) if its set of memory states is finite.

During a play at round i, Player I first updates the state according to μ_1 and the values of predicates $\bar{P}(i)$, then outputs its value according to θ, and then after a move of Player II update the state according to μ_2. Hence, a play $\rho := (\rho_{X_1}(0), \rho_{X_2}(0)) \dots (\rho_{X_1}(i), \rho_{X_2}(i)) \dots$ is consistent with such a strategy if there are $q_0, q'_0, \dots, q_i, q'_i$ such that $q_0 = \mu_1(q_{init}, \bar{P}(0))$, $\rho_{X_1}(i) = \theta(q_i)$, $q'_i = \mu_2(q_i, \rho_{X_2}(i))$ and $q_{i+1} = \mu_1(q'_i, \bar{P}(i+1))$.

Notational Conventions

1. In Hintikka's Lemma we considered formulas with the free variables among $X_1, \dots, X_l$. It can be extended trivially to formulas with free second-order variables in any finite list $\bar{V}$. In particular we use $\mathit{Char}^k(X, Y, Z)$ for the set of Hintikka formulas of quantifier depth k with free variables X, Y, Z.

2. Whenever we deal with the synthesis problem over an l-chain $\mathcal{M} = (\mathbb{N}, <, P_1, \dots, P_l)$, we will often replace variables Z_i by the predicate P_i; in particular we will write "$\varphi(X_1, X_2, P_1, \dots, P_l)$" instead of "$\varphi(X_1, X_2, Z_1, \dots, Z_l)$"

3. By Lemma 2.2, for every formula $\varphi(X_1, X_2, P)$ of a quantifier depth n there is $G \subseteq \mathit{Char}^n(X_1, X_2, P)$ such that φ is equivalent to the disjunction of all formulas from G. Moreover, G is computable from φ. We often identify φ with this set G and write "$\mathcal{G}_G^{\mathcal{M}}$" instead of "$\mathcal{G}_\varphi^{\mathcal{M}}$".

Definition 3.1. *Let $\mathcal{M}$ be an l-chain, st be a strategy, and $G \subseteq \mathit{Char}^m(X_1, X_2, \bar{P})$. st* wins *G over $\mathcal{M}$ iff the m-type of every play (on $\mathcal{M}$) consistent with st is in G.*

Definition 3.2 (Game Types). *Let $n \in \mathbb{N}$.*

Game type of a chain *Let $\mathcal{M} := (A, < \bar{P})$ be an l-chain, where $(A, <)$ is finite or of order type ω. The n-game-type of $\mathcal{M}$ is defined as:*
$\textit{game-type}_n(\mathcal{M}) := \{G \subseteq \mathit{Char}^n(X_1, X_2, \bar{P}) \mid$ Player I wins $\mathcal{G}_G^{\mathcal{M}}\}$.

Formal game-type *A formal (n, l)-game-type is an element[2] of $\mathbb{P}(\mathbb{P}(\mathit{Char}^n(X_1, X_2, \bar{P})))$, where $\bar{P}$ is an l-tuple $(P_1, \dots, P_l)$ of variables. We denote by Gtype_l^n the set of formal (n, l)-game-types.*

Let F be a function from $\mathbb{N}$ into Gtype_l^n and $G \subseteq \mathit{Char}^n(X_1, X_2, \bar{P})$. We consider the following ω-game $\mathrm{Game}(F, G)$.

Game(F, G): The game has ω rounds and it is defined as follows:

Round i: Player I chooses $G_i \in F(i)$. Then, Player II chooses $\tau_i \in G_i$.

Winning conditions: Let τ_i $(i \in \mathbb{N})$ be the sequence of moves of Player II in the play. Player I wins the play if $\sum_{i \in \mathbb{N}} \tau_i \in G$.

The following lemma is immediate:

[2] recall that $\mathbb{P}(A)$ stands for the set of subsets of A.

Lemma 3.3. *If $\forall i\big(F_1(i) \subseteq F_2(i)\big)$, $G_1 \subseteq G_2$ and Player I wins Game(F_1, G_1), then Player I wins Game(F_2, G_2).*

The following proposition plays an important role in our proofs:

Proposition 3.4. *Assume that $F(i)$ $(i \in \mathbb{N})$ is ultimately periodic. Then, it is decidable which of the players wins Game(F, G), Moreover, the winner has a finite-memory winning strategy which is computable from G.*

PROOF. We provide a reduction from Game(F, G) to a McNaughton game over ω. Let $Char^n(X_1, X_2.\bar{P}) := \{\tau_1, \dots, \tau_m\}$. For every $G' \subseteq Char^n(X_1, X_2.\bar{P})$

- Let $\varphi_{G'}(X_1, X_2)$ be $\bigvee_{\tau \in G'} \tau$ - the disjunction of all formulas from G'.
- Let $\psi_{G'}(Y_1, \dots, Y_m)$ be constructed from $\varphi_{G'}$ as in the Composition Theorem (Theorem 2.7).

Let $\{G_1, \dots, G_k\}$ be the set of all formal (n, l)-game-types. Define formula $\varphi_{F,G}(X_1, \dots, X_k, Y_1, \dots, Y_m)$ as the disjunction of 1-3

1. (a) For all t exactly one of $X_i(t)$ $(i = 1, \dots, k)$ holds and
   (b) For all t: $X_i(t) \rightarrow \big(G_i \in F(t)\big)$ and
   (c) $\psi_G(Y_1, \dots Y_m)$.
2. There is t such that not exactly one of $Y_j(t)$ holds.
3. There is t and $i \in \{1, \dots, k\}$ such that $X_i(t)$ and $\neg Y_j(t)$ for every $\tau_j \in G_i$.

Note that F is ultimately periodic and therefore *MLO* definable. Hence, 1(b) can be expressed in *MLO*. All other conditions are clearly expressible in *MLO*.

Consider the McNaughton game $\mathcal{G}^{\omega}_{\varphi_{F,G}}$. The second disjunct forces Player II at each round to assign the value 1 exactly to one of Y_j, and the third disjunct forces Player II to reply to the choice of X_i of Player I by choosing Y_j such that $\tau_j \in G_i$. It is clear that Player I (respectively, Player II) has a winning strategy in Game(F, G) iff Player I (respectively, Player II) has a winning strategy in $\mathcal{G}^{\omega}_{\varphi_{F,G}}$. By the Büchi-Landweber theorem, $\mathcal{G}^{\omega}_{\varphi_{F,G}}$ is determinate, and it is decidable who wins the game and the winner of $\mathcal{G}^{\omega}_{\varphi_{F,G}}$ has a finite-memory winning strategy. This finite-memory strategy corresponds to a finite-memory winning strategy in Game(F, G). □

4. Winning strategies over classes of finite chains

In the introduction we defined McNaughton's games over expansions of ω. In this subsection we will consider the games over expansions of finite chains. These games are defined similarly. The only difference is that these games are of finite length. Games over an l-chains with m elements have m rounds.

The main result of this section is Proposition 4.7. It deals with conditions for existence of a finite-memory strategy which uniformly wins over a class of finite chains.

The following lemma says that there is a sentence which uniformly expresses that Player I has a winning strategy in the game with winning condition φ.

Lemma 4.1. *For every φ there is a formula win(φ) such that for every finite l-chain $\mathcal{M}$, Player I has a winning strategy in $\mathcal{G}^{\mathcal{M}}_{\varphi}$ iff $\mathcal{M} \models$ win(φ). Furthermore, win(φ) is computable from φ.*

PROOF. (Sketch) In [11] we proved much stronger result (Theorem 2.3 in [11]) which says that there is a formula win_{φ} such that if $\mathcal{M}$ is an expansion of ω, then Player I has a winning strategy in $\mathcal{G}^{\mathcal{M}}_{\varphi}$ if and only if $\mathcal{M} \models win_{\varphi}$. It is easy to transfer the result from ω-chains to finite chains. Alternatively, it is easy to simplify this proof for finite chains. □

Recall that we identify a subset G of $Char^m(X_1, X_2, \bar{P})$ with the disjunction $\vee_{\tau \in G}\tau$. In particular, for $G \subseteq Char^m(X_1, X_2, \bar{P})$ we write $win(G)$ for $win(\vee_{\tau \in G}\tau)$.

For $C \subseteq \mathbb{P}(Char^m(X_1, X_2, \bar{P}))$ we write $Win(C)$ for $\wedge_{G \in C} win(G)$. $Win(C)$ expresses that Player I has a winning strategy for every $G \in C$.

Definition 4.2 (Residual). *For $\tau \in Char^m$ and $G \subseteq Char^m$, define $res_\tau(G)$ as $res_\tau(G) := \{\tau' \mid \tau + \tau' \in G\}$; define $Res(G)$ as $Res(G) := \{res_\tau(G) \mid \tau \in G\}$.*

Assume that ρ is a partial play of type τ. Player I can win $res_\tau(G)$ after ρ iff she has a strategy which ensures that every extension of ρ wins G.

Definition 4.3 (A winning strategy over a class of chains). *Let st be a strategy of Player I and $\mathcal{C}$ be a class of chains. We say that st wins φ over $\mathcal{C}$ iff st is a winning strategy in $\mathcal{G}_\varphi^{\mathcal{M}}$ for every $\mathcal{M} \in \mathcal{C}$.*

Lemma 4.4. *Assume that $\mathcal{M}_0$ and $\mathcal{M}_1$ are finite l-chains. If $\mathcal{M}_0 \models win(G)$ and $\mathcal{M}_1 \models Win(Res(G))$ then Player I has a finite-memory strategy which wins G over the class $\{\mathcal{M}_0 + \mathcal{M}_1 \times k \mid k \in \mathbb{N}\}$ of l-chains.*

PROOF. Let k_0 and k_1 be the length of $\mathcal{M}_0$ and $\mathcal{M}_1$ respectively. Consider the following strategy of Player I:

Play first k_0 rounds according to his winning strategy for $win(G)$. For every $j \in \mathbb{N}$ if the m-type of the play after $k_0 + jk_1$ rounds is τ then play the next k_1 rounds according to the winning strategy for $win(res_\tau(G))$.

It is easy to show by the induction on j that if a play ρ is played according to this strategy, then after $k_0 + jk_1$ rounds its m-type is in G. Therefore, it is a winning strategy for Player I.

Player I needs only a finite memory to keep the information about the m-type of the play τ_i up to each round i. After a round i she should add to τ_{i-1} the type of the play during the round i, i.e., to add to τ_{i-1} the m-type of one element chain expanded by the predicates $\rho_{X_1}(i)$, $\rho_{X_1}(i)$ and $P(i)$. Player I can calculate in a finite memory whether the current round number is $k_0 + jk_1$ for some $j \in \mathbb{N}$. Hence, this strategy is a finite-memory strategy. □

Definition 4.5 (Game type realized by a strategy). *Let $\mathcal{M}$ be an l-chain, st be a strategy, and $G \subseteq Char^m(X_1, X_2, \bar{P})$. st* realizes *$G$ on $\mathcal{M}$ if it wins $\mathcal{G}_G^{\mathcal{M}}$ and for every m-type $\tau \in G$ there is a play ρ consistent with st such that $type_m(\mathcal{M}^\frown\rho) = \tau$,*

In other words st realizes G in $\mathcal{M}$, if st wins $\mathcal{G}_G^{\mathcal{M}}$ and there is no $G_1 \subsetneq G$ such that st wins $\mathcal{G}_{G_1}^{\mathcal{M}}$. Recall that for $n \in \mathbb{N}$ we also denote by n the finite chain with n elements.

Lemma 4.6.

1. *If for $n_1 < n_2$ a strategy realizes G over chains n_1 and n_2, then $Win(Res(G))$ is satisfiable over the chain $n_2 - n_1$.*
2. *If for $n_1 < n_2$ a strategy realizes G over n_1 and wins G over n_2, then $Win(Res(G))$ is satisfiable over $n_2 - n_1$.*

PROOF. (1) follows from (2). (2) follows from the definition of *Win* and Definitions 4.2 and 4.5. □

Proposition 4.7. *For $m \in \mathbb{N}$, let n be an upper bound on the quantifier depth of win(G) for every $G \subseteq Char_2^m$, and let N_0 be computed from n as in Lemma 2.9. For every $i \in [0, N_0 - 1)$ the following are equivalent:*

1. *Player I has a finite-memory strategy which wins G over the class $\{t > N_0 \mid t \bmod N_0 = i\}$ of finite chains.*
2. *Player I has a finite-memory strategy which wins G over an infinite subclass of $\{t > N_0 \mid t \bmod N_0 = i\}$.*

3. *There is a finite-memory strategy which realizes* $G_1 \subseteq G$ *over* n_1 *and over* n_2 *for some* $n_2 > n_1 \geq N_0$ *such that* n_1 *mod* $N_0 = n_2$ *mod* $N_0 = i$.

4. *There is* $G_1 \subseteq G$ *such that* $N_0 + i \models win(G_1)$*, and* $N_0 \models win(G')$ *for every* $G' \in Res(G_1)$.

PROOF. The implication (1) $\Rightarrow$ (2) is immediate.

(2) $\Rightarrow$ (3). If a strategy wins G over $\mathcal{M}$ then it realizes a subset of G. Since the set of subset of G is finite, it follows that there is a subset of G which is realized infinitely often and therefore at least twice.

(3) $\Rightarrow$ (4) follows from Lemmas 2.9 and 4.6.

(4) $\Rightarrow$ (1) follows from Lemma 4.4. □

Proposition 4.7 is crucial for the design of our algorithm, due the decidability of (4).

5. Algorithm

Let P be an *ER* predicate and let $\mathcal{M} = (\mathbb{N}, <, P)$. We are going to prove that there is an algorithm that for every *MLO* formula $\varphi(X_1, X_2, Z)$ decides whether Player I has a finite-memory winning strategy in $\mathcal{G}^{\mathcal{M}}_{\varphi}$, and if so constructs such a strategy. It is easy to modify our proofs and to show that it is decidable whether Player II has a finite-memory winning strategy.

For every *MLO* formula $\varphi(X_1, X_2, P)$, first construct a set of the characteristic formulas G such that φ is equivalent to their disjunction and then use the following algorithm.

Synthesis algorithm over $\mathcal{M} := (\mathbb{N}, <, P)$ where P is in ER
Instance: $m \in \mathbb{N}$. *Task:* Find the set $Out = \{G \subseteq Char^m(X_1, X_2, P) \mid$ Player I has a finite-memory winning strategy in $\mathcal{G}^{\mathcal{M}}_{G}\}$, and for each $G \in Out$ construct a finite-memory strategy $st(G)$ which wins G over $\mathcal{M}$.

In this section we describe an algorithm for the synthesis problem for the expansions of ω by ER predicates.

First we describe ideas which underline the algorithm and then provide its detailed description.

Let $\bar{k} := k_0 < k_1 < \cdots < k_i < \dots$ be the enumeration of the elements of an ER predicate P in the increasing order and let $\mathcal{M} := (\mathbb{N}, <, P)$. Recall that $\mathcal{M} \bigm| I$ is the subchain of $\mathcal{M}$ over the set I. We can represent $\mathcal{M} \bigm| [k_l, \infty)$ as the following sums of chains.

$$\mathcal{M} \bigm| [k_l, \infty) = \sum_{s \in \omega} \mathcal{M} \bigm| [k_{l+s}, k_{l+s+1}) = \sum_{s \in \omega} \big(\mathcal{M} \bigm| [k_{l+s}, k_{l+s}] + \mathcal{M} \bigm| (k_{l+s}, k_{l+s+1})\big)$$

Note that $\mathcal{M} \bigm| [k_{l+s}, k_{l+s}]$ is isomorphic to the one element chain $(\{0\}, <, \{0\})$ and $\mathcal{M} \bigm| (k_{l+s}, k_{l+s+1})$ is isomorphic to an $(k_{l+s+1} - k_{l+s} - 1)$-element linear order expanded by the empty predicate.

Since, $\bar{k}$ is sparse and for every m the sequence k_l *mod* m is ultimately periodic we obtain (by Lemma 2.9) that the sequence of n-equivalence classes of $\mathcal{M} \bigm| (k_{l+s}, k_{l+s+1})$ is also ultimately periodic.

These observations together with Proposition 2.5 imply that for every n there is a lag l_n and a period p_n such that for $l > l_n$:

$$\mathcal{M} \bigm| [k_l, \infty) \equiv^n \mathcal{M} \bigm| [k_{l+p_n}, \infty)$$

Let st be a finite-memory strategy and $G \subseteq Char^m$ a winning condition. It is expressible by an *MLO* formula that st wins G.

Therefore, the ω-sequence $U^l_{st} := \{G \subseteq Char^m \mid st \text{ wins } G \text{ on } \mathcal{M} \bigm| [k_l, \infty)\}$ is also ultimately periodic. We will show that the ω-sequence $U^l := \{G \subseteq Char^m \mid$ there is a finite-memory strategy st which wins G on $\mathcal{M} \bigm| [k_l, \infty)\}$ is also ultimately periodic.

Similar arguments show that the sequence $V^l := \{G \subseteq Char^m \mid$ there is a finite-memory strategy st which wins G on $\mathcal{M} \bigm| [0, k_l)\}$ is ultimately periodic.

Our algorithm computes the (finite description) of ultimately periodic ω-sequences $\{U^l\}_{l=0}^{\infty}$ and $\{V^l\}_{l=0}^{\infty}$. From $\{U^l\}_{l=0}^{\infty}$ and $\{V^l\}_{l=0}^{\infty}$ we can compute the desirable *Out*. Indeed, let l and p be a join lag and period of these sequences. Then, $G \in Out$ iff there is $G_i \in V^{l+p}$ such that for every $\tau \in G_i$ we have that the residual (see Definition 4.2) $res_\tau(G)$ is in U^{l+p}. Indeed if there is such G_i then we can play the first k_{l+p} step according to a strategy which wins G_i on $\mathcal{M} \restriction [0, k_{l+p})$. This will ensure that after k_{l+p} steps our play will be of some m-type $\tau \in G_i$. Then we switch to a finite memory strategy which wins $res_\tau(G)$ on $\mathcal{M} \restriction [k_{l+p}, \infty)$. This will ensure that the m-type of the whole play will be in G. (Actually in the computation of Out we only used that we can compute a lag and period l and p of ω-sequences $\{U^i\}_{i=0}^{\infty}$ and $\{V^i\}_{i=0}^{\infty}$, and the elements U^{l+p} and V^{l+p} of these sequences.)

Note that there is a strategy which wins G on a finite chain $\mathcal{M}'$ iff there is a finite-state strategy which wins G on $\mathcal{M}'$. Lemma 4.1 states that it is *MLO* definable who has a winning strategy to win a game on a finite chain. This allows us to compute $\{V^l\}_{l=0}^{\infty}$.

The computation of $\{U^l\}_{l=0}^{\infty}$ is more subtle. Here Proposition 4.7 plays a crucial role by characterizing who has a finite state winning strategy over a periodic class of finite (unlabeled) chains.

In the rest of this section we provide a detailed description of our synthesis algorithm. We also prove the soundness of the algorithm, i.e., if $G \in Out$, then there is a finite-state strategy which wins G over $\mathcal{M}$. In the next section we show the reverse implication.

Conventions. Let $\tau(X_1, X_2)$ be an m-type for $m > 0$. There is the unique m-type $\tau^*(X_1, X_2, P)$ such that $\tau \rightarrow (\tau^*(X_1, X_2, P) \wedge \forall t \neg P(t))$. The m-type of a 2-chain $\mathcal{M}$ is τ iff the m-type of the expansion of $\mathcal{M}$ by the empty predicate is τ^*. We often will not distinguish between τ and the corresponding τ^*. In particular, for m-type $\tau_1(X_1, X_2, P)$ we write $\tau + \tau_1$ instead of $\tau^* + \tau_1$. We also lift this correspondence to sets of m-types; for a set $G \subseteq Char_2^m$ we sometimes use G for the set $G^* := \{\tau^* \mid \tau \in G\}$. It will be always clear from the context whether we refer to the type of a chain or to the type of the chain expanded by the empty predicate.

Now we are going to describe our algorithm.

Step 1

1. Compute $One := \{G \subseteq Char^m(X_1, X_2, P) \mid$ Player I has a strategy which wins G over the one element structure $(\{0\}, <, \{0\})\}$.

   For $G \in One$, we denote by $st_1(One, G)$ the corresponding winning strategy.

2. Let N_0 be defined from m as in Proposition 4.7. For $i = 0, \dots N_0 - 1$ compute $CWIN^i := \{G \subseteq Char^m(X_1, X_2) \mid$ Player I has a finite-memory strategy which wins G over the class $\{t > N_0 \mid t \ mod \ N_0 = i\}\}$. This set is computable by condition (4) of Proposition 4.7.

   For $G \in CWIN^i$, we denote by $st_1(i, G)$ the corresponding finite-memory winning strategy; this strategy is computable by Lemma 4.4, since the condition (4) of Proposition 4.7 holds.

Step 2 Let $\bar{k} := k_0 < k_1 < \dots < k_i < \dots$ be the enumeration of the elements of P in the increasing order. Compute l and p such that for every n greater than l:

1. $k_{n+1} - k_n > N_0$ and
2. $(k_{n+1} - k_n) \ mod \ N_0 = (k_{n+p+1} - k_{n+p}) \ mod \ N_0$
3. For $j < p$, set $d_j := (k_{l+j+1} - k_{l+j} - 1) \ mod \ N_0$.

(To compute such l and p we need our assumption that $P \in ER$.)

Step 3 Let $F : \mathbb{N} \rightarrow \text{Gtype}^m(X_1, X_2, P)$ be defined as follows:

$$F(i) = \begin{cases} One & \text{if } i \text{ is even} \\ CWIN^{d_j} & \text{if } i = 2s+1 \text{ and } s \ mod \ p = j \end{cases}$$

Note that F is a periodic sequence.

Use Proposition 3.4 to compute the set $U := \{G \subseteq Char^m(X_1, X_2, P) \mid$ Player I has a finite-memory strategy which wins $Game(F, G)\}$.

For $G \in U$, we denote by $st_{main}(F, G)$ the corresponding finite-memory winning strategy.

Now, for $G \in U$ we describe a finite-memory strategy $st_3(F, G)$ which wins G over the class $\{\mathcal{M}_i := \mathcal{M} \restriction [k_{l+pi}, \infty) \mid i \in \mathbb{N}\}$ of chains.

We organize our description of how strategy $st_3(F, G)$ behaves on $\mathcal{M}_i := \mathcal{M} \restriction [k_{l+pi}, \infty)$ in sessions. For $s \in \mathbb{N}$, the session $2s$ is played on the one element subchain of $\mathcal{M}_i$ isomorphic to $(\{0\}, <, \{0\})$; the session $2s + 1$ will be played on the subchain $\mathcal{M} \restriction (k_{l+pi+s}, k_{l+pi+s+1})$ which is isomorphic to the $(k_{l+pi+s+1} - k_{l+pi+s} - 1)$-element linear order expanded by the empty predicate.

Session 0. Let G_0 be the first move of $st_{main}(F, G)$. Then Player I will move according to his winning strategy in $st_1(One, G_0)$. After a move of Player II, the m-type of the partial play ρ_0 is some $\tau_0 \in G_0$.

Session $2s + 1$. Let G_{2s+1} be the move of Player I according to $st_{main}(F, G)$ after a partial play $G_0\tau_0G_1\tau_1 \dots G_{2s}\tau_{2s}$. Then Player I will play according to his strategy in $st_1(d_{(s \bmod p)}, G_{2s+1})$ until she reads one on P (recall that d_j, were defined in Step 2). At this point the type of a subplay ρ_{2s+1} during this round will be $\tau_{2s+1} \in G_{2s+1}$.

Session $2s$. *($s > 0$)* Let G_{2s} be the move of Player I according to $st_{main}(F, G)$ after a partial play $G_0\tau_0G_1\tau_1 \dots G_{2s-1}\tau_{2s-1}$. Player I will move according to his winning strategy in $st_1(One, G_{2s})$. After a move of Player II, the m-type of the partial play ρ_{2s} during this session will be some $\tau_{2s} \in G_{2s}$.

Observe that this is indeed a finite-memory strategy. Like in the proof of Lemma 4.4, Player I can compute in a finite memory at each session s the m-type τ_s of the subplay during session s, and then after this session supply only this m-type to $st_{main}(F, G)$ (and not the whole history $G_0\tau_0 \dots G_s\tau_s$).

This strategy wins G because the sequence $G_0\tau_0 \dots G_s\tau_s \dots$ played over the sessions is consistent with the winning strategy $st_{main}(F, G)$ in $Game(F, G)$.

Step 4 We are going to compute the set $V := \{G \subseteq Char^m(X_1, X_2, P) \mid$ Player I has a strategy which wins G over $\mathcal{M} \restriction [0, k_{l+pi})$ for some $i \in \mathbb{N})\}$.

Let n be the quantifier depth of $win(G)$.

By our choice of N_0, l and p (in Step 1 and Step 2) we know that for every i:

$$\mathcal{M} \restriction [k_{l+i}, k_{l+i+1}) \equiv^n \mathcal{M} \restriction [k_{l+i+p}, k_{l+i+1+p})$$

Hence, for every i:

$$\begin{aligned}\mathcal{M} \restriction [k_{l+pi}, k_{l+pi+p}) = \textstyle\sum_{s=0}^{p-1} \mathcal{M} \restriction [k_{l+pi+s}, k_{l+pi+s+1}) \equiv^n \\ \equiv^n \textstyle\sum_{s=0}^{p-1} \mathcal{M} \restriction [k_{l+s}, k_{l+s+1}) = \mathcal{M} \restriction [k_l, k_{l+p})\end{aligned}$$

Let $N_1 := N_1(n)$ be defined as in Lemma 2.10. From the above equivalence, Lemma 2.10 and Proposition 2.5, it follows that for every i there is $j \leq N_1$ such that

$$\mathcal{M} \restriction [k_l, k_{l+pi}) \equiv^n \mathcal{M} \restriction [k_l, k_{l+pj})$$

and hence, $\mathcal{M} \restriction [0, k_{l+pi}) \equiv^n \mathcal{M} \restriction [0, k_{l+pj})$.

Therefore, $V = \{G \subseteq Char^m(X_1, X_2, P) \mid \mathcal{M} \restriction [0, k_{l+pj}) \models win(G)$ for some $j \leq N_1\}$. To compute the right hand side we solve the satisfiability problem for a finite set of formulas over a finite set of finite chains. Hence, this is computable and therefore, V is computable.

For $G \in V$, let $l_G \leq N_1$ be such that $\mathcal{M} \restriction [0, k_{l+pl_G}) \models win(G)$ and let $st_4(V, G)$ be the corresponding strategy which wins G over $\mathcal{M} \restriction [0, k_{l+pl_G})$.

Step 5 Output $Out := \{G \subseteq Char^m(X, Y, P) \mid \exists G_1 \in V$ such that $res_\tau(G) \in U$ for every $\tau \in G_1\}$.

For every $G \in Out$ we describe a finite-memory strategy $st(G)$ which wins G over $\mathcal{M}$. Assume $G \in Out$ and let $G_1 \in V$ be such that $res_\tau(G) \in U$ for every $\tau \in G_1$. Since $G_1 \in V$, there is l_{G_1} and a strategy $st_4(V, G_1)$ which wins G_1 over $\mathcal{M} \restriction [0, k_{l+pl_{G_1}})$.

Player I will play the first $l + p \times l_{G_1}$ rounds according to this winning strategy. Let ρ be a play according to this strategy, and let τ be its m-type and let $G_2 = res_\tau(G)$. The rest of the game Player I will play according to his finite-memory strategy $st_3(F, G_2)$ computed in the Step 3. Clearly, the described strategy is a finite-memory strategy.

The m-type of the whole play is in $\tau + G_2 = G$. Therefore, the described strategy is winning in $\mathcal{G}^{\mathcal{M}}_G$. This completes the description of our algorithm and the proof that if $G \in Out$, then Player I has a finite-memory winning strategy in $\mathcal{G}^{\mathcal{M}}_G$.

6. Completeness of the Algorithm

In this section we prove the completeness of our algorithm, i.e., if there is a finite-memory strategy which wins G over $\mathcal{M}$, then $G \in Out$.

Definition 6.1. *Let $\mathcal{M} := (\mathbb{N}, <, P)$ be a chain, $I = (a, b)$ be an interval, $G \subseteq Char^m(X_1, X_2, P)$, and let st be a strategy of Player I.*

1. *We say that st* can realize *G on I (in $\mathcal{M}$) if there is a play ρ consistent with st on $\mathcal{M} \restriction [0, a]$ such that*
   - *(a) for every play $\rho' := \rho\rho_1$ which is consistent with st and extends ρ to the interval $[0, b)$, one has $type_m\big(\mathcal{M}^\frown \rho_1 \restriction (a, b)\big) \in G$, and*
   - *(b) for every $\tau_1 \in G$ there is $\rho' := \rho\rho_1$ which is consistent with st and extends ρ to the interval $[0, b)$, such that $type_m\big(\mathcal{M}^\frown \rho_1 \restriction (a, b)\big) = \tau_1$.*
2. *We say that st* can win *G on I (in $\mathcal{M}$) if st can realize some $G' \subseteq G$ (in $\mathcal{M}$).*

Let $\mathcal{M} := (\mathbb{N}, <, P)$ be a chain, $\bar{s} := s_0 < s_1 < \cdots < s_i < \ldots$ be an ω-sequence, st a strategy of Player I, and $m \in \mathbb{N}$.

Define $\mathcal{H} := \mathcal{H}(\bar{s}, st, m) : \mathbb{N} \to \mathrm{Gtype}^m(X_1, X_2, P)$ as follows:

$$G \in \mathcal{H}(2i) \text{ iff } st \text{ can realize } G \text{ on } [s_i, s_i]$$

$$G \in \mathcal{H}(2i+1) \text{ iff } st \text{ can realize } G \text{ on } (s_i, s_{i+1})$$

Notations: (Shift) For a function $T : \mathbb{N} \to A$ and $i \in \mathbb{N}$, the i-th shift of T is the function $\lambda j.\ T(i + j)$; we denote the i-shift of T by T_{+i}.

Lemma 6.2. *For every $\mathcal{M} := (\mathbb{N}, <, P)$, an increasing ω-sequence $\bar{s} := s_0 < s_1 < \cdots < s_i < \ldots$, and $m \in \mathbb{N}$, if Player's I strategy st can win G on $[s_i, \infty)$ then Player I has a winning strategy in $Game(\mathcal{H}(\bar{s}, st, m)_{+2i}, G)$.*

Proof. Since st can win G on $[s_i, \infty)$, there is a play ρ_{-1} consistent with st on $\mathcal{M} \restriction [0, s_i)$ such that for every play $\rho' := \rho_{-1}\rho$ which is consistent with st and extends ρ_{-1} to the interval $[0, \infty)$, one has $type_m\big(\mathcal{M}^\frown \rho \restriction [s_i, \infty)\big) \in G$.

Let $\mathcal{H}_{+2i}$ be the $2i$-shift of $\mathcal{H}(\bar{s}, st, m)$. Define the following strategy $st_{\mathcal{H}}$ for Player I in $Game(\mathcal{H}_{+2i}, G)$. Roughly speaking round $2j$ of this strategy corresponds to the play according to st on the subchain of $\mathcal{M}$ over $[s_{i+j}, s_{i+j}]$, and round $2j + 1$ corresponds to the play according to st on the subchain of $\mathcal{M}$ over (s_{i+j}, s_{i+j+1}).

Round $2j$**.** Set $R_{2j} := \{\rho \mid \rho_{2j-1}\rho$ is a play consistent with st on the interval $[0, s_{i+j}]\}$. Then play $G_{2j} := \{type_m(\rho) \mid \rho \in R_{2j}\}$. Note that this is a legal move, since $G_{2j} \in \mathcal{H}_{+2i}(2j)$.

Let $\tau_{2j} \in G_{2j}$ be a response of Player II. Let $\rho \in R_{2j}$ be a play of m-type τ_{2j}.

Set $\rho_{2j} := \rho_{2j-1}\rho$. Note that ρ_{2j} is play consistent with st on the interval $[0, s_{i+j+1}]$.

Round $2j + 1$**.** Set $R_{2j+1} := \{\rho \mid \rho_{2j}\rho$ is a play consistent with st on the interval $[0, s_{i+j+1})\}$. Then play $G_{2j+1} := \{type_m(\rho) \mid \rho \in R_{2j+1}\}$. Note that this is a legal move, since $G_{2j+1} \in \mathcal{H}_{+2i}(2j + 1)$.

Let $\tau_{2j+1} \in G_{2j+1}$ be a response of Player II. let $\rho \in R_{2j+1}$ be a play of m-type τ_{2j+1}.

Set $\rho_{2j+1} := \rho_{2j}\rho$. Note that ρ_{2j+1} is play consistent with st on the interval $[0, s_{i+j+1})$.

Since ρ_{j+1} extends ρ_j for each j and all of them are consistent with st there is an ω-play $\rho_{-1}\rho_\omega$ which extends all of them and is consistent with st. The m-type of ρ_ω is in G, because the m-type of every extension of ρ_{-1} consistent with st is in G. However, $type_m(\rho_\omega) = \sum \tau_i$. Therefore, $\sum \tau_i \in G$ and the described strategy $st_{\mathcal{H}}$ wins in $Game(\mathcal{H}(\bar{s}, st, m)_{+2i}, G)$. □

Define $\mathcal{F} := \mathcal{F}(\bar{s}, st, m) : \mathbb{N} \to \mathrm{Gtype}^m(X_1, X_2, P)$ as follows:

$$G \in \mathcal{F}(2i) \text{ iff } st \text{ can win } G \text{ on } [s_i, s_i]$$

$$G \in \mathcal{F}(2i+1) \text{ iff } st \text{ can win } G \text{ on } (s_i, s_{i+1})$$

Let $\mathcal{F}_{+2i}$ be the $2i$-shift of $\mathcal{F}$. Note that $\forall j(\mathcal{H}_{+2i}(j) \subseteq \mathcal{F}_{+2i}(j))$. Therefore, by Lemma 3.3 we obtain

Corollary 6.3. *If st can win G on $[s_i, \infty)$ then Player I has a winning strategy in the $Game(\mathcal{F}_{+2i}, G)$.*

Let $\mathcal{M} := (\mathbb{N}, <, P)$ be a chain, $\bar{k} := k_0 < k_1 < \cdots < k_i < \ldots$ be the enumeration of P in the increasing order, let st be a strategy of Player I, and $m \in \mathbb{N}$.

Let $\mathcal{H} := \mathcal{H}(\bar{k}, st, m)$. Let N_0, l, p and $d_0, \ldots, d_{p-1}$ be defined as in Step 2 of the algorithm, and let F be defined as in Step 3. Then we have the following Lemma:

Lemma 6.4. *If st is a finite memory strategy, then there is N such that for every i: $\mathcal{F}(\bar{k}, st, m)_{+2(l+Np)}(i) \subseteq F(i)$.*

Proof. For even i the lemma deals with games over one element chain $(\{0\}, <, \{0\})$, hence its conclusion holds for every even i and every N.

For $j < p$ define the class $\mathcal{C}_j$ of finite chains as $\mathcal{C}_j := \{k_{l+j+1+pi} - k_{l+j+pi} - 1 \mid i \in \mathbb{N}\}$. Assume that st is a finite state strategy with r states $q_1 \ldots q_r$. For $i = 1, \ldots, r$ let us denote by st_{q_i} the strategies which has the same update and output functions as st, and its initial state is q_i.

Let us write $Char^m$ for $Char^m(X_1, X_2)$. For $j < p$ define $Lim(st, j)$ as

> $Lim(st, j) := \{\, G \subseteq Char^m \mid$ there are infinitely many i such that st can win G on $(k_{l+j+1+pi} - k_{l+j+pi})$ in $\mathcal{M}\}$

Since $Char^m$ is finite there is N such that for every $j < p$ and $i' > N$

$$Lim(st, j) \supseteq \{G \subseteq Char^m \mid \ st \text{ can win } G \text{ on } (k_{l+j+1+pi'} - k_{l+j+pi'}) \text{ in } \mathcal{M}\}$$

The definition of $\mathcal{F}$, the above inclusion and (1) below immediately imply that this N works.

$$\forall s\big(s \bmod p = j\big) \to \big(Lim(st, j) \subseteq F(2s+1)\big) \tag{1}$$

Below a proof of (1) is given.

$Lim(st, j)$ is a subset of

$$\{\, G \subseteq Char^m \mid \text{ there are infinitely many } i \text{ such that one of } st_{q_1} \ldots st_{q_r} \text{ can win } G \text{ on } (k_{l+j+1+pi} - k_{l+j+pi}) \text{ in } \mathcal{M}\}$$

is equal to

$$\bigcup_{s=1}^{r} \{\, G \subseteq Char^m \mid st_{q_s} \text{ win } G \text{ on an infinite subclass of } \mathcal{C}_j\}$$

is a subset of

$$\{\ G \subseteq Char^m \mid \text{ Player I has a finite state strategy which wins } G \text{ on infinite subclass of } \mathcal{C}_j\}$$

(by Lemma 4.7 and the definition of l, p and d_j) this is equal to

$$\{\ G \subseteq Char^m \mid \text{ Player I has a finite state strategy which wins } G \text{ on the class } \{t > N_0 \mid t \ mod\ N_0 = d_j\}\}$$

(by the definition of $CWIN^d$ in Step 2) it is equal to $CWIN^{d_j}$ and (by definition of F in Step 3) it is equal to $F(2s+1)$, for every s such that $s\ mod\ p = j$.

Now we are ready to prove the completeness of our algorithm.

Let $G \subseteq Char^m(X_1, X_2, P)$ and assume that st is a finite-memory strategy of Player I which wins in $\mathcal{G}_G^{\mathcal{M}}$. Let $N := N(st)$ be as in Lemma 6.4.

Let $Pinit(st) := \{\rho \mid \rho \text{ is a play according to } st \text{ in } \mathcal{M} \restriction [0, l+pN)\}$.

Let type-init$(st) := \{type_m\big((\mathcal{M} \restriction [0, l+pN))^\frown \rho\big) \mid \rho \in Pinit(st)\}$.

Note that type-init$(st) \in V$, where V is defined in Step 4 of the algorithm.

We will show that for every $\tau \in$ type-init(st) the set $res_\tau(G)$ is in U, where U is defined in Step 3 of the algorithm. Therefore, by Step 5, we obtain that $G \in Out$.

For $\tau \in$ type-init(st), choose $\rho_\tau \in Pinit(st)$ such that

$$\tau = type_m\big((\mathcal{M} \restriction [0, l+pN))^\frown \rho_\tau\big).$$

Let $Pafter(st, \rho_\tau) := \{\rho \mid \rho_\tau\rho \text{ is consistent with } st \text{ in } \mathcal{M}\}$.

Let $G_\tau := \{type_m\big((\mathcal{M} \restriction [l+pN, \infty))^\frown \rho\big) \mid \rho \in Pafter(st, \rho_\tau)\}$.

st wins $\mathcal{G}_G^{\mathcal{M}}$, therefore for every $\rho \in Pafter(st, \rho_\tau)$ we have $type_m(\mathcal{M}^\frown \rho_\tau\rho) \in G$. However, $type_m(\mathcal{M}^\frown \rho_\tau\rho) = \tau + type_m\big((\mathcal{M} \restriction [l+pN, \infty))^\frown \rho\big)$, and therefore $G_\tau \subseteq res_\tau(G)$. On the other hand, st can win G_τ on $[l+pN, \infty)$ in $\mathcal{M}$, Therefore, by Corollary 6.3, Lemma 6.4, Lemma 3.3 and the definition of U in Step 3, we obtain that $G_\tau \in U$. Note that if $G' \in U$ and $G' \subseteq G''$ then $G'' \in U$. Since $G_\tau \in U$ and $G_\tau \subseteq res_\tau(G)$ we obtain that $res_\tau(G) \in U$.

7. Games with a bounded look-ahead

Let $\mathcal{M} = (\mathbb{N}, <, P)$ be the expansion of ω by a unary predicate P. Let h_1 and h_2 be natural numbers - look-aheads of the players. Let $\varphi(X_1, X_2, Z)$ be a formula. The game $\mathcal{G}_\varphi^{\mathcal{M}}(h_1, h_2)$ with look-ahead h_1 for Player I and look-ahead h_2 for Player II is defined as follows. The game is played by two players in ω rounds.

1. At round $i \in \mathbb{N}$: first, Player I chooses $\rho_{X_1}(i) \in \{0,1\}$; then, Player II chooses $\rho_{X_2}(i) \in \{0,1\}$. Player I can observe whether $i + h_1 \in P$ and Player II can observe whether $i + h_2 \in P$
2. By the end of the play two predicates $\rho_{X_1}, \rho_{X_2} \subseteq \mathbb{N}$ have been constructed.
3. Then, Player I wins the play if $\mathcal{M} \models \varphi(\rho_{X_1}, \rho_{X_2}, P)$; otherwise, Player II wins the play.

Hence, at round i, Player I has access only to $\rho_{X_1}(0), \dots, \rho_{X_1}(i-1)$, $\rho_{X_2}(0), \dots, \rho_{X_2}(i-1)$ and $P(h_1), \dots, P(h_1+i)$; Player II has access only to $\rho_{X_1}(0), \dots, \rho_{X_1}(i)$, $\rho_{X_2}(0), \dots, \rho_{X_2}(i-1)$ and $P(h_2), \dots, P(h_2+i)$.

If a player has a winning strategy in $\mathcal{G}_\varphi^{\mathcal{M}}(h_1, h_2)$ then she has a winning strategy in $\mathcal{G}_\varphi^{\mathcal{M}}(h_1', h_2')$ for every h_1' and h_2' (when there is no restriction on a strategy, on its i-th move it needs to know only the moves of the other player so far and all information about past and future bits of P can be kept in its memory). If

Player I has a finite-memory winning strategy in $\mathcal{G}_{\varphi}^{\mathcal{M}}(h_1, h_2)$ then she has a finite-memory winning strategy in $\mathcal{G}_{\varphi}^{\mathcal{M}}(h_1', h_2')$ for every $h_1' \geq h_1$ and h_2'.

The proof of the next proposition is based on a reduction to Theorem 1.4.

Proposition 7.1. *Let P be an ER predicate, $h_1,\ h_2 \in \mathbb{N}$ and let $\mathcal{M} := (\mathbb{N}, <, P)$. There is an algorithm that for every MLO formula $\varphi(X_1, X_2, Z)$ decides whether Player I has a finite-memory winning strategy in $\mathcal{G}_{\varphi}^{\mathcal{M}}(h_1, h_2)$, and if so, constructs such a strategy.*

Proof. Let $P_{-h} \subseteq \mathbb{N}$ be defined as $i \in P_{-h}$ iff $i + h \in P$.

Let $\mathcal{M}_{-h_1} := (\mathbb{N}, <, P_{-h_1})$. We are going to construct a formula $\varphi_{-h_1}(X_1, X_2, Z)$ such that Player I has a finite-memory winning strategy in $\mathcal{G}_{\varphi}^{\mathcal{M}}(h_1, h_2)$ iff Player I has a finite-memory winning strategy in $\mathcal{G}_{\varphi_{-h_1}}^{\mathcal{M}_{-h_1}}$.

For $i \leq h_1$ define a formula a_i as follows:

$$a_i := \begin{cases} \mathit{True} & \text{if } i \in P \\ \mathit{False} & \text{otherwise} \end{cases}$$

Let $\alpha(t, Z)$ be defined as

$$\alpha(t, Z) := \big(\bigwedge_{i<h_1} (t = i) \rightarrow a_i \big) \wedge \big((t \geq h_1) \rightarrow Z(t + h_1) \big)$$

Define $\varphi_{-h_1}(X_1, X_2, Z)$ as

$$\varphi_{-h_1}(X_1, X_2, Z) := \exists W (\forall t W(t) \leftrightarrow \alpha(t)) \wedge \varphi(X_1, X_2, W)$$

For $\pi_1, \pi_2 \subseteq \mathbb{N}$ we have

$$\omega \models \varphi(\pi_1, \pi_2, P) \text{ iff } \omega \models \varphi_{-h_1}(\pi_1, \pi_2, P_{-h_1})$$

Moreover, Player I has a finite-memory winning strategy in $\mathcal{G}_{\varphi}^{\mathcal{M}}(h_1, h_2)$ iff Player I has a finite-memory winning strategy in $\mathcal{G}_{\varphi_{-h_1}}^{\mathcal{M}_{-h_1}}$. Note that P is an ER predicate iff P_{-h} is an ER predicate. Hence, by Theorem 1.4. we obtain that it is decidable whether Player I has a finite-memory in strategy in $\mathcal{G}_{\varphi}^{\mathcal{M}}(h_1, h_2)$. □

Section 1 (page 3) gives an example of the game $\mathcal{G}_{\varphi}^{\mathcal{M}_{fac}}$ where Player I has a winning strategy, yet she has no finite-memory winning strategy. Note that for this particular game, Player I has a finite-memory one-look-ahead winning strategy, i.e., she has a finite-memory winning strategy in $\mathcal{G}_{\varphi}^{\mathcal{M}_{fac}}(1, h_2)$ for every h_2.

In [12] we proved determinacy of McNaughton games with parameters by *MLO*-definable strategies. We will prove that over every ER chain, the *MLO*-definable strategies coincide with the finite-memory with look-ahead strategies. Consequently, we obtain the following Theorem.

Theorem 7.2 (Determinacy for look-ahead finite-memory strategy). *Let P be an ER predicate, and let $\mathcal{M} = (\mathbb{N}, <, P)$. For every MLO formula $\varphi(X_1, X_2, Z)$ there is h such that one of the players has a finite-memory with look-ahead h winning strategy in $\mathcal{G}_{\varphi}^{\mathcal{M}}$. Furthermore, there is an algorithm that computes such h and a finite-memory winning strategy for the winner in $\mathcal{G}_{\varphi}^{\mathcal{M}}(h, h)$.*

In the next subsection we recall the main definability result of [12], and state a lemma which is used to derive Theorem 7.2 from the results of [12]. The proof of this lemma is postponed to subsection 7.2.

7.1. MLO-definable strategies

Recall that in a McNaughton game at round i, Player I has access only to the moves $\rho_{X_2}(0) \dots \rho_{X_2}(i-1)$ of Player II, and Player II has access only to the moves $\rho_{X_1}(0) \dots \rho_{X_1}(i)$ of player I. Therefore, the following formalizes well the notion of a strategy in this game:

Definition 7.3 (Causal operator). *Let $F : \mathbb{P}(\mathbb{N}) \rightarrow \mathbb{P}(\mathbb{N})$ maps the subsets of $\mathbb{N}$ into the subsets of $\mathbb{N}$. We call F* causal *(resp.* strongly causal*) iff for all $\rho, \rho' \subseteq \mathbb{N}$ and $i \in \mathbb{N}$:*

if $\rho \cap [0,i] = \rho' \cap [0,i]$ *(resp.* $\rho \cap [0,i) = \rho' \cap [0,i)$*), then*

$F(\rho) \cap [0,i] = F(\rho') \cap [0,i]$.

That is, if ρ *and* ρ' *agree* up to and including *(resp.* up to*)* i*, then* $F(\rho)$ *and* $F(\rho')$ *do so.*

An operator $F : \mathbb{P}(\mathbb{N}) \to \mathbb{P}(\mathbb{N})$ is *implicitly defined* by a formula $\psi(X,Y,P)$ over a structure $\mathcal{M} = (\mathbb{N}, <, P)$ if for any $\rho_1, \rho_2 \subseteq \mathbb{N}$ we have

$F(\rho_1) = \rho_2 \quad \text{iff} \quad \mathcal{M} \models \psi[\rho_1, \rho_2]$

and F is *implicitly MLO definable* over $\mathcal{M}$ iff it is defined by an *MLO* formula over $\mathcal{M}$. An operator F is *explicitly defined* by a formula $\alpha(X,P,t)$ over the structure $\mathcal{M}$ if for every $\rho_1, \rho_2 \subseteq \mathbb{N}$ the following holds:

$$\rho_2 = F(\rho_1) \text{ iff } \mathcal{M} \models \forall t(\rho_2(t) \leftrightarrow \alpha(\rho_1, t)).$$

Note that if F is implicitly defined by $\psi(X,Y,P)$ over $\mathcal{M}$ then it is explicitly defined by $\exists Y \psi \wedge Y(t)$. If F is explicitly defined by $\alpha(X,t,P)$, then it is implicitly defined by $\forall t(Y(t) \leftrightarrow \alpha)$.

Our proof of Theorem 7.2 is based on the following Theorem (cf. Theorem 2.3 in [12]) and does not rely on Theorem 1.4.

Theorem 7.4. *There is an algorithm that given a formula* $\varphi(X_1, X_2, Z)$ *constructs a sentence* $WIN^{II}_\varphi(Z)$ *and formulas* $St^I_\varphi(X_2, t, Z)$ *and* $St^{II}_\varphi(X_1, t, Z)$ *such that for every structure* $\mathcal{M} = (\mathbb{N}, <, P)$ *Player II wins the game* $\mathcal{G}^{\mathcal{M}}_\varphi$ *iff* $\mathcal{M} \models WIN^{II}_\varphi$*. Moreover, if Player II wins* $\mathcal{G}^{\mathcal{M}}_\varphi$*, then* St^{II}_φ *defines his winning strategy; otherwise,* St^I_φ *defines a winning strategy of Player I.*

Lemma 7.5. *Let* $\alpha(X_2, t, Z)$ *be a formula. Assume that* P *is an ER predicate and* α *defines a causal or strongly causal operator in* $\mathcal{M} := (\mathbb{N}, <, P)$*. Then*

1. *There is* $N_0(\alpha)$ *and a finite-memory strategy with look-ahead* N_0 *which computes the operator definable by* α*.*
2. *Furthermore,* $N_0(\alpha)$ *and a finite-state transducer for* st *are computable from* α *and* P*.*

We prove Lemma 7.5 in the next subsection.

Theorem 7.2 immediately follows from Theorem 7.4 and Lemma 7.5. It is clear that h can be defined as $h := \max\big(N_0(St^I_\varphi(X_1, X_2, Z)), N_0(St^{II}_\varphi(X_1, X_2, Z))\big)$.

7.2. Proof of Lemma 7.5

We are going to prove Lemma 7.5 for the case when α defines a causal operator. It is easy to modify the proof for strongly causal operators. We also assume that P is an infinite subset of $\mathbb{N}$. The case when P is finite is simpler.

For an infinite $P \subseteq \mathbb{N}$ define a function $succ_P : \mathbb{N} \to \mathbb{N}$ as follows:

$$succ_P(i) := \min(j \in P \mid j > i)$$

Let $\mathcal{M} := (\mathbb{N}, <, P)$ be a chain and $a \in \mathbb{N}$. $\mathcal{M}$ can be represented as the sum of three chains:

$$\mathcal{M} = \mathcal{M} \big| [0,a] + \mathcal{M} \big| (a, succ_P(a)) + \mathcal{M} \big| [succ_P(a), \infty)$$

As an instance of the Composition Theorem for the case when the index structure has three elements we obtain that for every $\alpha(X,t,Z)$ there is a finite set of tuples of formulas $\big(\tau^i_1(X,t,Z), \tau^i_2(X,Z), \tau^i_3(X,Z)\big)$ $(i < k)$ such that for every $Q, P \subseteq \mathbb{N}$ and $a \in \mathbb{N}$:

$\mathcal{M} \models \alpha(Q, a, P)$ if and only if there is i such that

1. $\mathcal{M} \big| [0,a] \models \tau^i_1(Q, a, P)$ and

2. $\mathcal{M} \upharpoonright (a, succ_P(a)) \models \tau_2^i(Q, P)$ and

3. $\mathcal{M} \upharpoonright [succ_P(a), \infty) \models \tau_3^i(Q, P)$.

Note that if $\alpha(X, t, Z)$ defines a causal operator in $\mathcal{M}$ then $\mathcal{M} \models \alpha(Q, a, P)$ iff $\mathcal{M} \models \alpha(Q', a, P)$, where $Q' \upharpoonright [0, a] = Q \upharpoonright [0, a]$ and Q' is empty on (a, ∞). Note also that P is empty on the interval $(a, succ_P(a))$. Therefore, we obtain

Lemma 7.6. *Assume that $\alpha(X, t, Z)$ defines a causal operator on $\mathcal{M} := (\mathbb{N}, <, P)$. Then there is a finite set of tuples of formulas $\big(\beta_1^i(X, t, Z), \beta_2^i, \beta_3^i(Z)\big)$ (for $i < k$) such that for every $a \in \mathbb{N}$ and $Q \subseteq \mathbb{N}$: $\mathcal{M} \models \alpha(Q, a, P)$ if and only if there is i such that*

1. *$\mathcal{M} \upharpoonright [0, a] \models \beta_1^i(Q, a, P)$ and*

2. *The chain with $(succ_P(a) - a - 1)$ elements satisfies β_2^i and*

3. *$\mathcal{M} \upharpoonright [succ_P(a), \infty) \models \beta_3^i(P)$.*

We are going to show that if P is an ER predicate, then each of the conditions in the above lemma is computable by a finite-memory operator with a look-ahead.

First, by the equivalence between MLO and finite automata over the class of finite chains we obtain

Claim 1. For every $\beta(X, t, Z)$ there is a finite memory strategy st for Player II such that for every $a \in \mathbb{N}$ and $Q \subseteq [0, a]$ if Q is a sequence of the first a moves of Player I in the McNaughton game on $\mathcal{M}$ then st outputs 1 at a-th move iff $\mathcal{M} \upharpoonright [0, a] \models \beta_1^i(Q, a, P)$.

The next Claim deals with the second condition.

Claim 2. For every sentence β and every ER predicate P there is $N_0 \in \mathbb{N}$ and a finite memory strategy st for Player II with look-ahead N_0 such that for every $a \in \mathbb{N}$, st outputs 1 at a-th move in the McNaughton game on $(\mathbb{N}, <, P)$ iff β is satisfiable on a linear order with $(succ_P(a) - a - 1)$ elements.

Proof of Claim 2. Let n be an upper bound on the quantifier depth of β and let $N_0 := N_0(n)$ be as in Lemma 2.9. Then there is $R \subseteq \{0, N_0 - 1\}$ such that for every $m \geq N_0$, an m-element linear order satisfies β iff $m \ mod \ N_0 \in R$. Let $S := \{i < N_0 \mid \beta \text{ is satisfiable on the } i\text{-element chain}\}$.

We are going to describe a finite-memory strategy with look-ahead N_0 which satisfies the conclusion of Claim 2.

Let $\bar{k} := k_0 < k_1 < \dots < k_i < \dots$ be the enumeration of the elements of P in the increasing order.

Let l' be such that $k_{i+1} - k_i > N_0$ for every $m > l'$. The sequence, $k_{i+1} - k_i \ mod \ N_0$ is ultimately periodic with a lag $l > l'$ and period p. For $j < p$, set $d_j := \big(k_{l+j+1} - k_{l+j} - 1\big) \ mod \ N_0$.

The desirable strategy st behaves as follows. For each $a < k_l$ it computes whether β is satisfiable on a chain with $(succ_P(a) - a - 1)$ elements, and outputs 1 on the round a if so.

For $a \geq k_l$ it uses its finite memory to calculate $j < p$ such that the current round a is in interval $[k_m, k_{m+1})$ for $j = m - l \ mod \ p$. When we are inside an interval $[k_m, k_{m+1})$, on every round a we compute $r_a := a - k_m - d_j \ mod \ N_0$ until $k_{m+1} - a < N_0$. We need a finite memory to compute r_a and N_0-look-ahead to check whether $k_{m+1} - a < N_0$. If $k_{m+1} - a \geq N_0$ then we output 1 if $(d_j - r_a) \ mod \ N_0 \in R$ and 0 otherwise. When $k_{m+1} - a < N_0$ we output 1 if $k_{m+1} - a \in S$ and 0 otherwise. □

The next claim asserts that the third condition of Lemma 7.6 can be computed by a finite memory strategy without look-ahead.

Claim 3. For every formula $\beta(Z)$ and every ER predicate P there is a finite memory strategy st for Player II such that for every $a \in \mathbb{N}$, st outputs 1 at a-th move in the McNaughton game on $(\mathbb{N}, <, P)$ iff $\mathcal{M} \upharpoonright [succ_P(a), \infty) \models \beta(P)$.

Proof of Claim 3. Let n be an upper bound on the quantifier depth of β and let $N_0 := N_0(n)$ be as in Lemma 2.9. Let $\bar{k} := k_0 < k_1 < \dots < k_i < \dots$ be the enumeration of the elements of P in the increasing order.

Let l' be such that $k_{i+1} - k_i > N_0$ for every $m > l'$ The sequence, $k_{i+1} - k_i \ mod \ N_0$ is ultimately periodic with a lag $l > l'$ and period p. For $j < p$, set $d_j := \big(k_{l+j+1} - k_{l+j} - 1\big) \ mod \ N_0$. Then for $m \geq l$, $\mathcal{M} \upharpoonright [k_m, k_{m+1})$ is $\equiv^n$ -equivalent to a chain $\mathcal{L}_j := (\{0, 1, \dots N_0 + d_j\}, <, \{0\})$, where $j = m - l \ mod \ p$. By

Proposition 2.5, $\equiv^n$ is a congruence with respect to the sum of chains; hence, if $m > l$ and $j = m - l \ mod \ p$, then $\mathcal{M} \bigm| [k_m, \infty) = \sum_{i \in \omega} \mathcal{M} \bigm| [k_{m+i}, k_{m+i+1})$ is $\equiv^n$-equivalent to the periodic chain $\mathcal{M}'_j := (\mathcal{L}_j + \mathcal{L}_{j+1} + \cdots + \mathcal{L}_{p-1} + \mathcal{L}_0 + \cdots + \mathcal{L}_{j-1}) \times \omega$. For $j < p$ define s_j as 1 if $\mathcal{M}'_j \models \beta$, and as 0 otherwise.

The desirable strategy st behaves as follows. For each $a < k_l$ it outputs 1 on the round a if β is satisfiable on $\mathcal{M} \bigm| [succ_P(a), \infty)$ and outputs 0 otherwise.

For $a \geq k_l$ it uses its finite memory to calculate $j < p$ such that a current move a is in interval $[k_{m-1}, k_m)$ for $j = m - l \ mod \ p$ and outputs s_j. □

Now, we are ready to prove Lemma 7.5. Assume that $\alpha(X_2, t, Z)$ defines a causal operator in $\mathcal{M} := (\mathbb{N}, <, P)$. We can compute $\big(\beta_1^i(X, t, Z), \beta_2^i, \beta_3^i(Z)\big)$ $(i < k)$ as in Lemma 7.6.

By Claims 1-3, for each β_j^i (for $i < k$ and $j \in \{1, 2, 3\}$) we can compute the corresponding finite-memory with a look-ahead strategy st_j^i. Let N_0 be an upper bound on the look-aheads of all these strategies. On each round we can calculate the output of all strategies st_j^i. If there is i such that the output of st_1^i, st_2^1 and st_3^i is 1, then we output 1; otherwise, we output 0. It is clear that this strategy st computes the operator definable by α. We need only finite memory to implement st, and st uses look-ahead N_0.

8. Conclusion

We proved that the finite-memory synthesis problem is decidable for the expansions of ω by predicates from *ER*. Let $k \in \mathbb{N}$ and P_k be the union of $\{n! \mid n \in \mathbb{N}\}$ and $\{n! + k \mid n \in \mathbb{N}\}$. For every $k > 0$, the predicate P_k is not sparse and hence it is not an ER predicate. However, a slight modification of our proof shows that the finite-memory synthesis problem is decidable for $\mathcal{M}_k := (\mathbb{N}, <, P_k)$. It is more difficult to prove that the finite-memory synthesis problem is decidable for $\mathcal{M} := (\mathbb{N}, <, P)$, where the characteristic function of P is the concatenation of $U_n := (0^n 1)^n$ (for $n \in \mathbb{N}$). The predicate P is sparse, but it is not residually ultimately periodic.

In [12] it was proved that the decidability of the monadic theory of $\mathcal{M}$ is equivalent to the decidability of the recursive strategy synthesis problem for $\mathcal{M}$.

The question whether the decidability of the monadic theory of $\mathcal{M}$ implies the decidability of the finite-memory synthesis problem for $\mathcal{M}$ remains open.

A natural question to consider is the synthesis problem for strategies between finite-memory and recursive ones, e.g., the strategies computable by push-down automata [21].

The use of the composition method in our proof can be hidden and a presentation can be given based on automata theoretic concepts. The logical n-types can be replaced by "n-types", using semigroups or automata rather than formulas to describe properties of words. However, such a proof would be unnatural.

Acknowledgments

I would like to thank anonymous referees for their helpful suggestions.

References

[1] J. R. Büchi. On a decision method in restricted second order arithmetic In *Proc. International Congress on Logic, Methodology and Philosophy of Science*, E. Nagel at al. eds, Stanford University Press, pp 1-11, 1960.
[2] J. R. Büchi and L. H. Landweber. Solving sequential conditions by finitestate strategies. Transactions of the AMS, 138(27):295–311, 1969.
[3] O. Carton and W.Thomas. The Monadic Theory of Morphic Infinite Words and Generalizations. Inf. Comput. 176(1), pp. 51-65, 2002.
[4] A. Church. Logic, Arithmetic and Automata, Proc. Intrnat. Cong. Math. 1963, Almquist and Wilksells, Uppsala, 1963.
[5] C. Elgot and M. O. Rabin. Decidability and Undecidability of Extensions of Second (First) Order Theory of (Generalized) Successor. J. Symb. Log., 31(2), pp. 169-181, 1966.
[6] E. Grädel, W. Thomas and T. Wilke. Automata, Logics, and Infinite Games, LNCS 2500, 2002.
[7] Y. Gurevich. Monadic second-order theories, in: J. Barwise, S. Feferman (eds.), *Model-Theoretic Logics*, Springer-Verlag, pp. 479-506, 1985.
[8] S. Feferman and R.L. Vaught. The first-order properties of products of algebraic systems. *Fundamenta Mathematica* **47**:57–103, 1959.

[9] R. McNaughton. Finite-state infinite games, Project MAC Rep., MIT, 1965.
[10] D. Perrin and J. E. Pin. Infinite Words Automata, Semigroups, Logic and Games. Pure and Applied Mathematics Vol 141 Elsevier, 2004.
[11] A. Rabinovich. On decidability of Monadic logic of order over the naturals extended by monadic predicates. Information and Computation, 205(6):870-889, 2007.
[12] A. Rabinovich. Church Synthesis Problem with Parameters. Logical Methods in Computer Science, Vol. 3 (4:9):1-24, 2007, DOI: 10.2168/LMCS-3(4:9)2007.
[13] A. Rabinovich. Decidable Extensions of Church's Problem. In CSL 2009, LNCS 5771, pp. 424-439, 2009.
[14] A. Rabinovich and W. Thomas. Decidable Theories of the Ordering of Natural Numbers with Unary Predicates. In CSL 2006, Springer LNCS 4207, 562-574, 2006.
[15] R. M. Robinson. Restricted Set-Theoretical Definitions in Arithmetic. In Proceedings of the AMS Vol. 9, No. 2. pp. 238-242, 1958.
[16] A. Semenov. Logical theories of one-place functions on the set of natural numbers. Mathematics of the USSR - Izvestia, vol. 22, pp 587-618, 1984.
[17] S. Shelah. The monadic theory of order. *Ann. of Math.* **102**:379–419, 1975.
[18] D. Siefkes. The recursive sets in certain monadic second order fragments of arithmetic. Arch. Math. Logik,17, pp. 71-80, 1975.
[19] W. Thomas, Ehrenfeucht games, the composition method, and the monadic theory of ordinal words. In A Selection of Essays in Honor of A. Ehrenfeucht, Springer, LNCS 1261 pp. 118-143, 1997.
[20] B. A. Trakhtenbrot. Finite automata and the logic of one-place predicates. (Russian version 1961). In AMS Transl. 59, pp. 23-55, 1966.
[21] I. Walukiewicz. Pushdown processes: games and model checking. Information and Computation 164 pp. 234-263, 2001.