THE CHURCH SYNTHESIS PROBLEM WITH PARAMETERS

ALEXANDER RABINOVICH

Sackler Faculty of Exact Sciences, Tel Aviv University, Israel 69978.
e-mail address: rabinoa@post.tau.ac.il

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

Büchi and Landweber proved that the Church synthesis problem is decidable; moreover, they showed that if there is an operator F that solves the Church Synthesis Problem, then it can also be solved by an operator defined by a finite state automaton or equivalently by an MLO formula. We investigate a parameterized version of the Church synthesis problem. In this version ψ might contain as a parameter a unary predicate P. We show that the Church synthesis problem for P is computable if and only if the monadic theory of $\langle Nat, <, P \rangle$ is decidable. We prove that the Büchi-Landweber theorem can be extended only to ultimately periodic parameters. However, the MLO-definability part of the Büchi-Landweber theorem holds for the parameterized version of the Church synthesis problem.

1. INTRODUCTION

Two fundamental results of classical automata theory are decidability of the monadic second-order logic of order (MLO) over $\omega = (Nat, <)$ 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.

Büchi [Bu60] proved that the monadic theory of $\omega = \langle Nat, < \rangle$ 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 $\langle Nat, <, + \rangle$ and the monadic theory of $\langle Nat, <, \lambda x.2 \times x \rangle$ are undecidable [Rob58, Trak61]. Therefore, most efforts to find decidable expansions of ω deal with expansions of ω by monadic predicates.

Elgot and Rabin [ER66] found many interesting predicates $\mathbf{P}$ for which MLO over $\langle Nat, <, \mathbf{P} \rangle$ is decidable. Among these predicates are the set of factorial numbers $\{n! : n \in Nat\}$, the sets of k-th powers $\{n^k : n \in Nat\}$ and the sets $\{k^n : n \in Nat\}$ (for $k \in Nat$).

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., [Ch69, Th75, Sem84, CT02]). In [Rab05, Rab07, RT06] 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.

Let Spec be a specification language and Pr be an implementation language. The synthesis problem for these languages is stated as follows: find whether for a given specification $S(I, O) \in$ SPEC there is a program $\mathcal{P}$ which implements it, i.e., $\forall I(S(I, \mathcal{P}(I))$.

The specification language for the Church Synthesis problem is the Monadic second-order Logic of Order. An MLO formula $\varphi(X, Y)$ specifies a binary relation on subsets of Nat. Note that every subset P of Nat is associated with its characteristic ω-string u_P (where $u_P(i) = 1$ if $i \in P$ and otherwise $u_P(i) = 0$). Hence, $\varphi(X, Y)$ can be considered as a specification of a binary relation on ω-strings.

As implementations, Church considers functions from the set $\{0, 1\}^\omega$ of ω-strings over $\{0, 1\}$ to $\{0, 1\}^\omega$. Such functions are called operators. A machine that computes an operator at every moment $t \in Nat$ reads an input symbol $X(t) \in \{0, 1\}$ and produces an output symbol $Y(t) \in \{0, 1\}$. Hence, the output $Y(t)$ produced at t depends only on inputs symbols $X(0), X(1), \ldots, X(t)$. Such operators are called *causal* operators (C-operators); if the output $Y(t)$ produced at t depends only on inputs symbols $X(0), X(1), \ldots, X(t-1)$, the corresponding operator is called *strongly causal* (SC-operator). The sets of recursive causal and strongly causal operators are defined naturally; a C-or a SC-operator is a *finite state* operator if it is computable by a finite state automaton (for precise definitions, see Subsection 4.4).

The following problem is known as the Church Synthesis problem.

Church Synthesis problem

Input: an MLO formula $\psi(X, Y)$.

Task: Check whether there is a C-operator F such that
$Nat \models \forall X \psi(X, F(X))$ and if so, construct this operator.

The Church Synthesis problem is much more difficult than the decidability problem for MLO over ω. Büchi and Landweber [BL69] proved that the Church synthesis problem is computable. Their main theorem is stated as follows:

Theorem 1.1. *For every MLO formula $\psi(X, Y)$ either there is a finite state C-operator F such that $Nat \models \forall X \psi(X, F(X))$ or there is a finite state SC-operator G such that $Nat \models \forall Y \neg \psi(G(Y), Y)$. Moreover, it is decidable which of these cases holds and a corresponding operator is computable from ψ.*

In this paper we consider natural generalizations of the Church Synthesis Problem over expansions of ω by monadic predicates, i.e., over the structures $\langle Nat, <, \mathbf{P} \rangle$.

For example, let $\mathbf{Fac} = \{n! : n \in Nat\}$ be the set of factorial numbers, and let $\varphi(X, Y, \mathbf{Fac})$ be a formula which specifies that $t \in Y$ iff $t \in \mathbf{Fac}$ and $(t' \in X) \leftrightarrow (t' \in \mathbf{Fac})$ for all $t' \leq t$. It is easy to observe that there is no finite state C-operator F such that $\forall X \varphi(X, F(X), \mathbf{Fac})$. However, there is a recursive C-operator H such that $\forall X \varphi(X, H(X), \mathbf{Fac})$. It is also easy to construct a finite state C-operator $G(X, Z)$ such that $\forall X \varphi(X, G(X, \mathbf{Fac}), \mathbf{Fac})$. It was surprising for us to discover that it is decidable whether for a formula $\psi(X, Y, \mathbf{Fac})$ there is a C-operator F such that $\forall X \varphi(X, F(X), \mathbf{Fac})$ and if such an operator exists, then it is recursive and computable from ψ.

Here is the summary of our results. We investigate a parameterized version of the Church synthesis problem. In this version ψ might contain as a parameter a unary predicate P. Below five synthesis problems with a parameter $\mathbf{P} \subseteq Nat$ are stated. We use capital italic letters for unary predicate names and set variables and the corresponding bold letters for their interpretation.

Synthesis Problems for $\mathbf{P} \subseteq Nat$

Input: an *MLO* formula $\psi(X, Y, P)$.

Problem 1: Check whether there is a C-operator $Y = F(X, P)$ such that
$Nat \models \forall X \psi(X, F(X, \mathbf{P}), \mathbf{P})$ and if there is such a recursive operator
- construct it.

Problem 2: Check whether there is a recursive C-operator $Y = F(X, P)$ such
that $Nat \models \forall X \psi(X, F(X, \mathbf{P}), \mathbf{P})$ and if so - construct this operator.

Problem 3: Check whether there is a recursive C-operator $Y = F(X)$ such
that $Nat \models \forall X \psi(X, F(X), \mathbf{P})$ and if so - construct this operator.

The next two problems are obtained from problems 2 an 3 when "recursive" is replaced by "finite state".

Synthesis Problems for $\mathbf{P} \subseteq Nat$

Input: an *MLO* formula $\psi(X, Y, P)$.

Problem 4: Check whether there is a finite state C-operator $Y = F(X, P)$ such
that $Nat \models \forall X \psi(X, F(X, \mathbf{P}), \mathbf{P})$ and if so - construct this operator.

Problem 5: Check whether there is a finite state C-operator $Y = F(X)$ such
that $Nat \models \forall X \psi(X, F(X), \mathbf{P})$ and if so - construct this operator.

We show

Theorem 1.2. *Let $\mathbf{P}$ be a subset of Nat. The following conditions are equivalent:*

(1) *Problem 1 for $\mathbf{P}$ is computable.*
(2) *Problem 2 for $\mathbf{P}$ is computable.*
(3) *Problem 3 for $\mathbf{P}$ is computable.*
(4) *The monadic theory of $\langle Nat, <, \mathbf{P} \rangle$ is decidable.*
(5) *For every MLO formula $\psi(X, Y, P)$ either there is a recursive C-operator F such that $Nat \models \forall X \psi(X, F(X), \mathbf{P})$ or there is a recursive SC-operator G such that $Nat \models \forall Y \neg \psi(G(Y), Y, \mathbf{P})$. Moreover, it is decidable which of these cases holds and the (description of the) corresponding operator is computable from ψ.*

The more difficult part of this theorem is the implication (4)$\Rightarrow$(5).

The trivial examples of predicates with decidable monadic theory are ultimately periodic predicates. Recall that a predicate $\mathbf{P}$ is ultimately periodic if there are $0 < p, d \in Nat$ such that $(n \in \mathbf{P} \leftrightarrow n + p \in \mathbf{P})$ for all $n > d$. Ultimately periodic predicates are *MLO*-definable. Hence, for these predicates computability of Problems 1-5 can be derived from Theorem 1.1.

We prove that the Büchi-Landweber theorem can be extended only to ultimately periodic parameters.

Theorem 1.3. *Let $\mathbf{P}$ be a subset of Nat. The following conditions are equivalent and imply computability of Problem 4:*

(1) $\mathbf{P}$ *is ultimately periodic.*
(2) *For every MLO formula $\psi(X, Y, P)$ either there is a finite state C-operator F such that $Nat \models \forall X \psi(X, F(X, \mathbf{P}), \mathbf{P})$ or there is a finite state SC-operator G such that $Nat \models \forall Y \neg \psi(G(Y, \mathbf{P}), Y, \mathbf{P})$.*

In Problems 1-5 we restrict the computational complexity of the C-operators (implementations) which meet specifications. Another approach is to restrict their descriptive complexity. The finite state operators are MLO-definable in $\omega = \langle Nat, < \rangle$. An operator F is defined by a formula $\psi(X, Y, P)$ in an expansion $M = \langle Nat, <, \mathbf{P} \rangle$ of ω, if for all ω-strings $\mathbf{X}$ and $\mathbf{Y}$:
$$\mathbf{Y} = F(\mathbf{X}) \text{ iff } \omega \models \psi(\mathbf{X}, \mathbf{Y}, \mathbf{P})$$
An operator F is MLO-definable in $M = \langle Nat, <, \mathbf{P} \rangle$, if it is defined by an MLO formula in M

Our main theorem which is stated in the next section implies

Theorem 1.4. *For every MLO formula $\psi(X, Y, P)$ and every expansion $M = \langle Nat, <, \mathbf{P} \rangle$ of ω either there is an MLO-definable (in M) C-operator F such that $M \models \forall X \psi(X, F(X), P)$ or there is an MLO-definable (in M) SC-operator G such that $M \models \forall Y \neg \psi(G(Y), Y, P)$. Moreover, formulas which define these operators are computable from ψ*

The paper is organized as follows. In the next section games and their connections to the Church synthesis problem are discussed, the Büchi and Landweber theorem is rephrased in the game theoretical language, and our main definability result - Theorem 2.3 - which implies Theorem 1.4 is stated.

In Section 3, Theorem 1.2 is derived as a consequence of Theorem 2.3. In Section 4, standard definitions and facts about automata and logic are recalled. In Section 5, finite state synthesis problems with parameters are considered and Theorem 1.3 is proved.

In Section 6, parity games on graphs and their connection to the synthesis problems are discussed and definability results needed in the proof of Theorem 2.3 are proved. The proof of Theorem 2.3 is given in Section 7.

Finally, in Section 8, some open problems are stated and further results are discussed.

2. Game Version of the Church Problem and Main Definability Result

Let $\mathbf{W}$ be a set of pairs of ω strings over $\{0, 1\}$. A game $G(\mathbf{W})$ is defined as follows.
(1) The game is played by two players, called Player I (or Mr. X) and Player II (or Mr. Y).
(2) A *play* of the game has ω rounds.
(3) At round n: first, Player I chooses $x(n) \in \{0, 1\}$; then, Player II chooses $y(n) \in \{0, 1\}$.
(4) By the end of the play the ω strings $\mathbf{Y} = y(0)y(1)\ldots$ and $\mathbf{X} = x(0)x(1)\ldots$ over $\{0, 1\}$ have been constructed.

Winning conditions: Player II wins the play if the pair of ω-strings $\langle \mathbf{X}, \mathbf{Y} \rangle$ is in $\mathbf{W}$; otherwise, Player I wins the play.

What we want to know is: Does either one of the players have a *winning strategy* in $G(\mathbf{W})$. That is, can Player I choose his moves so that in whatever way Player II responds, we have $\langle \mathbf{X}, \mathbf{Y} \rangle \notin \mathbf{W}$? Or can Player II respond to Player I's moves in a way that ensures the opposite?

Since at round n, Player I has access only to $\mathbf{Y} \cap [0, n)$ and Player II has access only to $\mathbf{X} \cap [0, n]$, a strategy of Player I (respectively, of Player II) is a strongly causal (respectively, causal) operator. So, a winning strategy for Player II is a causal operator $F : \{0,1\}^\omega \to \{0,1\}^\omega$ such that $\langle \mathbf{X}, F(\mathbf{X}) \rangle$ is in $\mathbf{W}$ for every $\mathbf{X} \in \{0,1\}^\omega$, and a winning strategy for Player I is a strongly causal operator $G : \{0,1\}^\omega \to \{0,1\}^\omega$ such that $\langle G(\mathbf{Y}), \mathbf{Y} \rangle$ is not in $\mathbf{W}$ for every $\mathbf{Y} \in \{0,1\}^\omega$.

There is a natural topology on the set of ω-strings (see e.g., [PP04]). According to this topology, subsets of ω-strings can be classified as open, closed, Borel and so forth. The following theorem of Martin is fundamental.

Theorem 2.1 (Determinacy of Borel Games)**.** *For every Borel set* $\mathbf{W}$*, one of the players has a winning strategy in* $G(\mathbf{W})$*.*

It was McNaughton (see [Mc66]) who first observed that the Church problem can be equivalently phrased in game-theoretic language. Algorithmic questions deal with finitely described objects. Hence, McNaughton considered games $G(\mathbf{W})$ only for definable sets $\mathbf{W}$.

Let $M = \langle Nat, <, \mathbf{P} \rangle$ be a structure. A set $\mathbf{W} \subseteq \{0,1\}^\omega \times \{0,1\}^\omega$ is *defined by an MLO formula* $\psi(X, Y, P)$ *in* M if $\mathbf{W} = \{\langle \mathbf{X}, \mathbf{Y} \rangle : M \models \psi(\mathbf{X}, \mathbf{Y}, P)\}$. A set is *MLO-definable* iff it is defined by an *MLO* formula. Similarly, an operator $F : \{0,1\}^\omega \to \{0,1\}^\omega$ is definable in M if its graph $\{\langle \mathbf{X}, \mathbf{Y} \rangle : \mathbf{Y} = F(\mathbf{X})\}$ is definable; a strategy is definable iff the corresponding causal operator is definable.

Let $M = \langle Nat, <, \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$ be an expansion of $\omega = \langle Nat, < \rangle$ by unary predicates and let $\psi(X, Y, P_1, \ldots, P_n)$ be an *MLO* formula. The *McNaughton game* $\mathcal{G}_\psi^M$ is the game $G(\mathbf{W})$, where $\mathbf{W}$ is the set definable by ψ in M. Hence, the winning condition of $\mathcal{G}_\psi^M$ can be stated as

Winning conditions for $\mathcal{G}_\psi^M$**:** Player II wins a play if $M \models \psi(\mathbf{X}, \mathbf{Y}, P_1, \ldots, P_n)$; otherwise, Player I wins the play.

This leads to

Game version of the Church problem: Let $M = \langle Nat, <, \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$ be an expansion of $\omega = \langle Nat, < \rangle$ by unary predicates. Given a formula $\psi(X, Y, P_1, \ldots, P_n)$ decide whether Player II has a winning strategy in $\mathcal{G}_\psi^M$.

Theorem 1.1 states the computability of the Church problem in the structure $\omega = \langle Nat, < \rangle$ (no additional unary predicates). Even more importantly, Büchi and Landweber show that in the case of ω we can restrict ourselves to *definable strategies*, i.e., to causal (or strongly causal) operators computable by finite state automata or equivalently *MLO*-definable in ω. The Büchi-Landweber theorem can be stated in the game theoretical language as follows:

Theorem 2.1 (Büchi-Landweber, 1969)**.** *Let* $\psi(X, Y)$ *be an MLO formula., Then:*

Determinacy: *One of the players has a winning strategy in the game* $\mathcal{G}_\psi^\omega$*.*
Decidability: *It is decidable* which *of the players has a winning strategy.*
Definable strategy: *The player who has a winning strategy also has an MLO-definable winning strategy.*
Synthesis algorithm: *We can compute a formula* $\varphi(X, Y)$ *that defines (in* $(\omega, <)$*) a winning strategy for the winning player in* $\mathcal{G}_\psi^\omega$*.*

The determinacy part of the Büchi-Landweber theorem follows from the determinacy of Borel games. More generally, it is well known that for each *MLO* formula $\psi(X, Y, Z_1, \ldots, Z_n)$

and $\mathbf{P}_1, \ldots, \mathbf{P}_n \subseteq Nat$ the set $\mathbf{W} = \{\langle \mathbf{X}, \mathbf{Y} \rangle : \langle Nat, < \rangle \models \psi(\mathbf{X}, \mathbf{Y}, \mathbf{P}_1, \ldots, \mathbf{P}_n)\}$ is a Borel set (it is even inside the boolean closure of the second level of the Borel hierarchy, see e.g., [PP04]). Hence, by the determinacy of Borel games, we obtain determinacy of the McNaughton games. In other words, we have the following corollary:

Corollary 2.2 (Determinacy). *Let $M = \langle Nat, <, \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$ be an expansion of ω by unary predicates. Then, for every MLO formula $\psi(X, Y, P_1, \ldots, P_n)$*

(1) *One of the players has a winning strategy in $\mathcal{G}_\psi^M$.*
(2) *Equivalently, either there is a C-operator F such that $M \models \forall X \psi(X, F(X))$, or there is a SC-operator G such that $M \models \forall Y \neg\psi(G(Y), Y)$.*

In order to simplify notations, from now on, we will state our results only for the expansions of ω by one unary predicate. The generalization to the expansions by any number of unary predicates is straightforward.

Now, we are ready to state our main result which generalizes the definability and synthesis parts of the Büchi-Landweber theorem in a uniform way to the expansions of $\langle Nat, < \rangle$ by unary predicates.

Theorem 2.3 (Main). *There is an algorithm that given a formula $\varphi(X, Y, P)$ constructs a sentence $WIN_\varphi^{II}(P)$ and formulas $St_\varphi^I(X, Y, P)$, $St_\varphi^{II}(X, Y, P)$ such that for every structure $M = \langle Nat, <, \mathbf{P} \rangle$ Player II wins the games $\mathcal{G}_\varphi^M$ iff $M \models WIN_\varphi^{II}$. Moreover, if Player II wins $\mathcal{G}_\varphi^M$, then $St_\varphi^{II}(X, Y, P)$ defines his winning strategy; otherwise, St_φ^I defines a winning strategy of Player I.*

Theorem 1.4 is reformulated in the game language as follows.

Theorem 2.4 (Game version of Theorem 1.4). *For every MLO formula $\psi(X, Y, P)$ and every expansion $M = \langle Nat, <, \mathbf{P} \rangle$ of ω:*

Determinacy: *One of the players has a winning strategy in the game $\mathcal{G}_\psi^M$.*
Definable strategy: *The player who has a winning strategy also has an MLO-definable (in M) winning strategy.*
Synthesis algorithm: *We can compute a formula that defines (in M) a winning strategy for the winning player in $\mathcal{G}_\psi^M$.*

The determinacy part of Theorem 2.4 follows from Corollary 2.2. Its definability and synthesis parts are immediate consequences of Theorem 2.3. In the next section, Theorem 1.2 is derived as another consequence of Theorem 2.3 and of Corollary 2.2.

The proof of Theorem 2.3 will be given in Sect. 7. Section 4 provides an additional background on logic and automata, and Section 6 prepares definability results needed in the proof of Theorem 2.3.

3. Proof of Theorem 1.2

In this section we prove Theorem 1.2. Its more difficult part is the implication (4)⇒(5); the proof of this implication is easily obtained from Theorem 2.3. The proof of the other equivalences of Theorem 1.2 uses only determinacy and some simple facts.

We start with the following simple Lemma:

Lemma 3.1. *Assume that the monadic theory of $M = \langle Nat, <, \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$ is decidable*

(1) *Every set $\mathbf{Q} \subseteq Nat$ definable in M is recursive.*

(2) *Every C-operator F definable in M is recursive.*

(3) *There is an algorithm that computes a program for $\mathbf{Q}$ (respectively, for F) from a formula which defines $\mathbf{Q}$ (respectively, F).*

Proof. (1) Assume that $\mathbf{Q}$ is defined in M by a formula $\psi(X)$. Each $j \in Nat$ is defined by an MLO formula $\varphi_j(t)$. Hence, one can check whether $j \in \mathbf{Q}$, by testing whether the sentence $\exists X \exists t \varphi_j(t) \wedge \psi(X) \wedge t \in X$ holds in M.

(2) Assume that a C-operator F is defined in M by a formula $\psi(X, Y)$. We have to show that there is an algorithm which computes the n-th letter of $F(\mathbf{X})$ from the first n letters of $\mathbf{X}$. Let $a = a_0 \ldots a_n \in \{0, 1\}^*$. Define $\zeta_i^a(t, X)$ as $\varphi_i(t) \wedge t \in X$ if a_i is 1, and as $\varphi_i(t) \wedge t \notin X$ otherwise, where $\varphi_i(t)$ is a formula which defines a number i. Note that $M \models \exists t_0 \exists t_1 \ldots \exists t_n \bigwedge_{i=0}^n \zeta_i^a(t_i, \mathbf{X}))$ iff the first $n + 1$ letters of $\mathbf{X}$ are $a_0 \ldots a_n$.

Now, if $a_0 \ldots a_n \in \{0, 1\}^*$ are the first $n + 1$ letters of $\mathbf{X}$, then the $n + 1$-th letter of $F(\mathbf{X})$ is 1 iff

$$\exists X \exists Y \exists t_0 \exists t_1 \ldots \exists t_n \big(\psi(X, Y) \wedge \varphi_n(t_n) \wedge t_n \in Y \wedge \bigwedge_{i=0}^n \zeta_i^a(t_i, X) \big)$$

holds in M.

(3) follows from the proofs of (1) and (2). $\qquad \square$

Lemma 3.2 (Implication (4)$\Rightarrow$(5) of Theorem 1.2)**.** *Assume that the monadic theory of $M = \langle Nat, <, \mathbf{P} \rangle$ is decidable. Then for every MLO formula $\varphi(X, Y, P)$ either there is a recursive C-operator F such that $Nat \models \forall X \varphi(X, F(X), \mathbf{P})$ or there is a recursive SC-operator G such that $Nat \models \forall Y \neg \varphi(G(Y), Y, \mathbf{P})$. Moreover, it is decidable which of these cases holds and the (description of the) corresponding operator is computable from φ.*

Proof. For a formula φ, construct $WIN_\varphi^{II}(P)$ and formulas $St_\varphi^I(X, Y, P)$, $St_\varphi^{II}(X, Y, P)$, as in Theorem 2.3.

By the assumption that the monadic theory of M is decidable, we can check whether $WIN_\varphi^{II}(P)$ holds in M.

If $WIN_\varphi^{II}(P)$ holds in M then, by Theorem 2.3, $St_\varphi^{II}(X, Y, P)$ defines a winning strategy F for Player II. Hence, F is C-operator and $Nat \models \forall X \varphi(X, F(X), \mathbf{P})$. By Lemma 3.1, F is recursive.

If $WIN_\varphi^{II}(P)$ does not hold in M, then by Theorem 2.3, $St_\varphi^I(X, Y, P)$ defines a winning strategy G for Player I. Hence, G is SC-operator and $Nat \models \forall Y \neg \varphi(G(Y), Y, \mathbf{P})$. Moreover, by Lemma 3.1, G is recursive. $\qquad \square$

Lemma 3.3. *If one of the Problems 1-5 is computable for $\mathbf{P}$, then the monadic theory of $\langle Nat, <, \mathbf{P} \rangle$ is decidable.*

Proof. Let $\beta(P)$ be a sentence in MLO and let $\psi_\beta(X, Y, P)$ be defined as

$$\big(\beta \rightarrow (Y = \{0\}) \big) \wedge \big(\neg \beta \rightarrow (X = \emptyset) \big).$$

Observe that $Nat \models \beta(\mathbf{P})$ iff there is a C-operator F such that:

$$Nat \models \forall X \psi_\beta(X, F(X, \mathbf{P}), \mathbf{P}) \text{ iff } Nat \models \forall X \psi_\beta(X, H(X, \mathbf{P}), \mathbf{P})$$

where H is a constant C-operator defined as $H = \lambda \langle X, P \rangle.10^\omega$.

Hence, if one of the Problems 1-5 is computable for $\mathbf{P}$, then we can decide whether $Nat \models \beta(\mathbf{P})$. $\qquad \square$

The proof of Lemma 3.3 also implies that if the following Problem $1'$ is decidable for **P**, then the monadic theory of $\langle Nat, <, \mathbf{P} \rangle$ is decidable.

Decision Problem $1'$ for $\mathbf{P} \subseteq Nat$

Input: an *MLO* formulas $\psi(X, Y, P)$.
Question: Check whether there is a C-operator $Y = F(X, P)$ such that
$$Nat \models \forall X \psi(X, F(X, \mathbf{P}), \mathbf{P}).$$

Problem $1'$ is actually Problem 1 without construction part.

Finally, we have

Lemma 3.4. *The implications (5)$\Rightarrow$(1), (5)$\Rightarrow$(2) and (5)$\Rightarrow$(3) hold.*

Proof. Let $\psi(X, Y, P)$ be a formula. By (5) either there is a recursive C-operator F such that $Nat \models \forall X \psi(X, F(X), \mathbf{P})$ or there is a recursive SC-operator G such that $Nat \models \forall Y \neg \psi(G(Y), Y, \mathbf{P})$. Moreover, it is decidable which of these cases holds and the corresponding operator is computable from ψ.

In the first case, the answer to Problems 1-3 is positive and F is a corresponding operator.

In the second case, the answer to Problems 1-3 is negative.

Indeed, for the sake of contradiction, assume that there is a C-operator (even non-recursive) F such that $Nat \models \forall X \psi(X, F(X, \mathbf{P}), \mathbf{P})$. Observe that F is a C-operator and G is a SC-operator. Hence, $H = \lambda X. G(F(X, \mathbf{P}))$ is a SC-operator. Every SC-operator has a fixed point. Let $\mathbf{X}_0$ be a fixed point of H and let $\mathbf{Y}_0 = F(\mathbf{X}_0, \mathbf{P})$. Then we have: $\mathbf{X}_0 = G(\mathbf{Y}_0)$. Therefore, we obtain

$$Nat \models \psi(\mathbf{X}_0, \mathbf{Y}_0, \mathbf{P}) \quad \text{because} \quad Nat \models \forall X \psi(X, F(X, \mathbf{P}), \mathbf{P}) \;, \quad \text{and}$$

$$Nat \models \neg \psi(\mathbf{X}_0, \mathbf{Y}_0, \mathbf{P}) \quad \text{because} \quad Nat \models \forall Y \neg \psi(G(Y), Y, \mathbf{P}) \;.$$

Contradiction. $\qquad\qquad\square$

4. BACKGROUND ON LOGIC AND AUTOMATA

4.1. **Notations and Terminology.** We use k, l, m, n, i for natural numbers; *Nat* for the set of natural numbers and capital bold letters **P**, **S**, **R** for subsets of *Nat*. We identify subsets of a set A and the corresponding unary (monadic) predicates on A.

The set of all (respectively, non-empty) finite strings over an alphabet Σ is denoted by Σ^* (respectively, by Σ^+). The set of ω-strings over Σ is denoted by Σ^ω.

Let $a_0 \ldots a_k \ldots$ and $b_0 \ldots b_k \ldots$ be ω-strings. We say that these ω-strings coincide on an interval $[i, j]$ if $a_k = b_k$ for $i \le k \le j$. A function F from Σ_1^ω to Σ_2^ω will be called an operator of type $\Sigma_1 \rightarrow \Sigma_2$. An operator F is called *causal* (respectively, *strongly causal*) operator, if $F(X)$ and $F(Y)$ coincide on an interval $[0, t]$, whenever X and Y coincide on $[0, t]$ (respectively, on $[0, t)$). We will refer to causal (respectively, strongly causal) operators as C-operators (respectively, SC-operators).

Let Σ_1 and Σ_2 be finite alphabet and let $F : \Sigma_1^\omega \rightarrow \Sigma_2^\omega$ be a C-operator. Note that there is a unique function $h_F : \Sigma_1^* \rightarrow \Sigma_2$ such that $F(a_1 \ldots a_n) = b_n$ if for some (equivalently for all) ω-string y: b_n is the n-th letter of $F(a_1 \ldots a_n y)$. F is said to be *recursive* if h_F is recursive.

Every SC-operator F of type $\Sigma \rightarrow \Sigma$ has a unique fixed point, i.e., there is a unique $X \in \Sigma^\omega$ such that $X = F(X)$.

Let $G : \Sigma^\omega \to \Delta^\omega$ be an operator. In the case Σ is the Cartesian product $\Sigma_1 \times \Sigma_2$ we will identify G with the corresponding operator $F : \Sigma_1^\omega \times \Sigma_2^\omega \to \Delta^\omega$. An operator $F : \Sigma_1^\omega \times \Sigma_2^\omega \to \Delta^\omega$ is said to be SC-operator (C-operator) if G is SC-operator (respectively, C-operator).

There exists a one-one correspondence between the set of all ω-strings over the alphabet $\{0,1\}^n$ and the set of all n-tuples $\langle \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$ of unary predicates over the set of natural numbers. With an n-tuple $\langle \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$ of unary predicates over Nat, we associate the ω-string $a_0 a_1 \ldots a_k \ldots$ over alphabet $\{0,1\}^n$ defined by $a_k =_{def} \langle b_1^k, \ldots b_n^k \rangle$ where b_i^k is 1 if $\mathbf{P}_i(k)$ holds and b_i^k is 0 otherwise. Let $Q = \{q_1, \ldots, q_m\}$ be a finite set of state. There is a natural one-one correspondence between the subsets of $Q \times Nat$ and the set of m-tuples of unary predicates over Nat: with $U \subseteq Q \times Nat$ we associate the m-tuple $\langle \mathbf{P}_1, \ldots, \mathbf{P}_m \rangle$ defined as $i \in \mathbf{P}_j$ iff $U(q_j, i)$ (for $i \in Nat$ and $j \leq m$).

Similarly, there is a one-one correspondence between the set of all strings of length m over the alphabet $\{0,1\}^n$ and the set of all n-tuples $\langle \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$ of unary predicates over the set $\{0, \ldots, m-1\}$.

A linearly ordered set will be called a chain. A chain with n monadic predicates over its domain will be called an n-labelled chain; whenever n is clear from the context, n-labelled chains will be called labelled chains.

We will sometimes identify an n-labelled chain $M = \langle Nat, <, \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$ with the ω-string over the alphabet $\{0,1\}^n$ which corresponds to the n-tuple $\langle \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$; this ω-string will be called the *characteristic* ω-string (or ω-word) of M. Similarly, we will identify finite n-labelled chains with corresponding strings over $\{0,1\}^n$.

4.2. **Monadic Second-Order Logic and Monadic Logic of Order.** Let σ be a relational signature. Atomic formulas of the monadic second-order logic over σ are $R(t_1, ..., t_n)$, $t_1 = t_2$, and $t_1 \in X$ where $t_1, \ldots, t_n$ are individual variables, $R \in \sigma$ is an n-are relational symbol, and X is a set variable. Formulas are obtained from atomic formulas by conjunction, negation, and quantification $\exists t$ and $\exists X$ for t an individual and X a set variable. The satisfaction relation $M, \tau_1, \ldots \tau_k; \mathbf{S}_1, \ldots, \mathbf{S}_m \models \varphi(t_1, \ldots, t_k; X_1, \ldots, X_m)$ is defined as usual with the understanding that set variables range over subsets of M.

We use standard abbreviations, e.g., we write $X \subseteq X'$ for $\forall t.\ X(t) \to X'(t)$; we write $X = X'$ for $\forall t.\ X(t) \leftrightarrow X'(t)$; symbols "$\exists^{\leq 1}$" and "$\exists!$" stands for "there is at most one" and "there is a unique".

If a signature σ contains one binary predicate $<$ which is interpreted as a linear order, and all other predicates are unary, the monadic second-order logic for this signature is called Monadic Logic of Order (MLO). The formulas of MLO are interpreted over labelled chains.

The *monadic theory* of a labelled chain M is the set of all MLO sentences which hold in M.

We will deal with the expansions of ω by monadic predicates, i.e., with the structures of the form $M = \langle Nat, <, \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$. We say that a chain $M = \langle Nat, <, \mathbf{P}_1, \ldots, \mathbf{P}_n \rangle$ is *recursive* if all $\mathbf{P}_i$ are recursive subsets of Nat.

An ω-language L is said to be *defined* by an MLO formula $\psi(X_1, \ldots, X_n)$ if the following condition holds: an ω string is in L iff the corresponding n-tuple of unary predicates satisfies ψ.

4.3. **The First-Order Version of the Monadic Second-Order Logic.** Sometimes it
will be convenient for us to consider the first-order version of the monadic second order
logic.

Let σ be a relational signature and M be a structure for σ.

Let $\bar{\sigma} = \sigma \cup \{Sing, \subseteq\}$ where $Sing$ is a new unary relational symbol and $\subseteq$ a new binary
relational symbol. Let $\bar{M}$ be the structure for $\bar{\sigma}$ defined as follows: The domain of $\bar{M}$ is
the set of all subsets of the domain of M. $Sing(A)$ holds in $\bar{M}$ if A is one element subset;
$A \subseteq B$ holds if A is a subset of B; for k-ary relational symbol $R \in \sigma$:

$$R(A_1, \dots A_k) \text{ holds in } \bar{M} \text{ iff}$$

$$A_1 = \{a_1\}, \dots, A_k = \{a_k\} \text{ and } R(a_1, \dots a_k) \text{ holds in } M$$

The following lemma is well-known and is easily proved by the structural induction.

Lemma 4.1 (Equivalence of two Versions of Monadic Logic). *The two versions of Monadic
logic are expressive equivalent, that is*

(1) *Let $\psi(X_1, \dots, X_k)$ be a formula of the monadic second-order logic for a signature σ.
There is a first-order formula $\varphi(X_1, \dots, X_k)$ in the signature $\bar{\sigma}$ such that for every
structure M for the signature σ and for subsets $A_1, \dots, A_k$ of the domain of M*

$$M, A_1, \dots, A_k \models \psi(X_1, \dots, X_k) \text{ iff } \bar{M}, A_1, \dots, A_k \models \varphi(X_1, \dots, X_k).$$

Moreover there is an algorithm that computes φ from ψ.
(2) *Let $\varphi(X_1, \dots, X_k)$ be a first-order formula in the signature $\bar{\sigma}$. There is a formula in the
monadic second-order logic for the signature σ such that for every structure M for the
signature σ and for subsets $A_1, \dots, A_k$ of the domain of M*

$$M, A_1, \dots, A_k \models \psi(X_1, \dots, X_k) \text{ iff } \bar{M}, A_1, \dots, A_k \models \varphi(X_1, \dots, X_k).$$

Moreover there is an algorithm that computes ψ from φ. $\square$

4.4. **Automata.** A *deterministic transition system* D is a tuple $\langle \mathcal{Q}, \Sigma, \delta, q_{init} \rangle$, consisting
of a set $\mathcal{Q}$ of *states*, an *alphabet* Σ, a *transition function* $\delta : \mathcal{Q} \times \Sigma \to \mathcal{Q}$ and *initial state*
$q_{init} \in \mathcal{Q}$. The transition function is extended as usual to a function from $\mathcal{Q} \times \Sigma^*$ to $\mathcal{Q}$
which will be also denoted by δ. The function $\delta_{init} : \Sigma^* \to \mathcal{Q}$ is defined as $\delta_{init}(\pi) = \delta(q_{init}, \pi)$.
A transition systems is finite if $\mathcal{Q}$ and Σ are finite.

A *finite deterministic automaton* $\mathcal{A}$ is a tuple $\langle \mathcal{Q}, \Sigma, \delta, q_{init}, F \rangle$, where $\langle \mathcal{Q}, \Sigma, \delta, q_{init} \rangle$ is
a finite deterministic transition system and F is a subset of $\mathcal{Q}$. A string $\pi \in \Sigma^*$ is accepted
by $\mathcal{A}$ if $\delta_{init}(\pi) \in F$. The language accepted (or defined) by $\mathcal{A}$ is the set of string accepted
by $\mathcal{A}$.

A *Mealey automaton* is a tuple $\langle \mathcal{Q}, \Sigma, \delta, q_{init}, \Delta, out \rangle$, where $\langle \mathcal{Q}, \Sigma, \delta, q_{init} \rangle$ is a deter-
ministic transition system, Δ is an alphabet and $out : \mathcal{Q} \to \Delta$ is an output function. With
a Mealey automaton $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \delta, q_{init}, \Delta, out \rangle$ we associate a function $h_{\mathcal{A}} : \Sigma^* \to \Delta$ and an
operator $F_{\mathcal{A}} : \Sigma^\omega \to \Delta^\omega$ defined as follows:

$$h_{\mathcal{A}}(a_0 \dots a_{i-1}) = out(\delta_{init}(a_0 \dots a_{i-1}))$$

$$F_{\mathcal{A}}(a_0 \dots a_i \dots) = b_0 \dots b_i \dots \text{ iff } b_i = h_{\mathcal{A}}(a_0 \dots a_{i-1})$$

It is easy to see that an operator is strongly causal (SC-operator) iff it is definable by a
Mealey automaton. We say that a SC-operator $F : \Sigma^\omega \to \Delta^\omega$ is finite state iff it is definable
by a finite state Mealey automaton.

A finite Mealey automaton $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \rightarrow, \delta, q_{init}, \Delta, col \rangle$, where the output alphabet Δ is a (finite) subset of Nat, is called a (deterministic) *parity automaton*; the output function *col* is usually refered to as *coloring function*.

With every ω-string $a_0 a_1 \cdots a_i \cdots \in \Sigma^\omega$ we associate the ω-sequence of successive states $\delta_{init}(a_0)\delta_{init}(a_0 a_1) \cdots \delta_{init}(a_0 \cdots a_i) \cdots$ and the set $\texttt{Inf}$ of all $q \in \mathcal{Q}$ that appear infinitely many times in this sequence. An ω-string is accepted by $\mathcal{A}$ if the minimal element of the set $\{col(q) : q \in \texttt{Inf}\}$ is even. The ω-language *accepted* (or defined) by $\mathcal{A}$ is the set of all ω-strings accepted by $\mathcal{A}$.

Sometimes the alphabet Σ of $\mathcal{A}$ will be the Cartesian product $\Sigma_1 \times \Sigma_2 \times \Sigma_3$ of other alphabets. In this case we say that $\mathcal{A}$ defines a relation $R_{\mathcal{A}} \subseteq \Sigma_1^\omega \times \Sigma_2^\omega \times \Sigma_3^\omega$; a triplet $\langle a, b, c \rangle$ of ω-strings is in $R_{\mathcal{A}}$ iff the ω string $(a_0, b_0, c_0)(a_1, b_1, c_1) \ldots (a_i, b_i, c_i) \ldots$ is accepted by $\mathcal{A}$.

Here is the classical theorem due to Büchi, Elgot and Trakhtenbrot.

Theorem 4.1. (1) *A language is accepted by a finite deterministic automaton iff it is definable by an MLO formula.*
(2) *An ω-language is accepted by a deterministic parity automaton iff it is definable by an MLO formula.*
(3) *Moreover, there is an algorithm which for every formula $\varphi(X_1, \ldots, X_m)$ computes an equivalent deterministic automaton $\mathcal{A}$ i.e., the language definable by φ is accepted by $\mathcal{A}$. There is an algorithm which for every deterministic automaton $\mathcal{A}$ computes an equivalent MLO formula. Similarly, there are translation algorithms between formulas and deterministic parity automata.* $\square$

A *Moore automaton* is a tuple $\langle \mathcal{Q}, \Sigma, \delta, q_{init}, \Delta, out \rangle$, where $\langle \mathcal{Q}, \Sigma, \delta, q_{init} \rangle$ is a deterministic transition system, Δ is an alphabet and $out : \mathcal{Q} \times \Sigma \rightarrow \Delta$ is an output function.

With a Moore automaton $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \delta, q_{init}, \Delta, out \rangle$ we associate a function $h_{\mathcal{A}} : \Sigma^+ \rightarrow \Delta$ and an operator $F_{\mathcal{A}} : \Sigma^\omega \rightarrow \Delta^\omega$ defined as follows:

$$h_{\mathcal{A}}(a_0 \ldots, a_i) = out(\delta_{init}(a_0 \ldots a_{i-1}), a_i)$$

$$F_{\mathcal{A}}(a_0 \ldots a_i \ldots) = b_0 \ldots b_i \ldots \text{ iff } b_i = h_{\mathcal{A}}(a_0 \ldots, a_i)$$

It is easy to see that an operator is causal (C-operator) iff it is definable by a Moore automaton.

We say that a C-operator $F : \Sigma^\omega \rightarrow \Delta^\omega$ is finite state iff it is definable by a finite state Moore automaton.

5. Finite State Synthesis Problems with Parameters

Recall that a predicate $\mathbf{P} \subseteq Nat$ is ultimately periodic if there is $p, d \in Nat$ such that $(n \in \mathbf{P} \leftrightarrow n + p \in \mathbf{P})$ for all $n > d$. Ultimately periodic predicates are *MLO*-definable. Therefore, for every ultimately periodic predicate $\mathbf{P}$ the monadic theory of $\langle Nat, <, \mathbf{P} \rangle$ is decidable.

The next theorem implies Theorem 1.3 and shows that Theorem 1.1 can be extended only to ultimately periodic predicates.

Theorem 5.1. *Let $\mathbf{P}$ be a subset of Nat. The following conditions are equivalent and imply computability of Problem 4:*
(1) $\mathbf{P}$ *is ultimately periodic.*

(2) *For every MLO formula $\psi(X, Y, P)$ either there is a finite state C-operator F such that $Nat \models \forall X \psi(X, F(X, \mathbf{P}), \mathbf{P})$ or there is a finite state C-operator G such that $Nat \models \forall Y \neg \psi(G(Y, \mathbf{P}), Y, \mathbf{P})$.*
(3) $\mathbf{P}$ *satisfies the following selection condition:*
    For every formula $\alpha(X, P)$ such that $Nat \models \exists X \alpha(X, \mathbf{P})$ there is a finite state C-operator $H : \{0,1\}^\omega \to \{0,1\}^\omega$ such that $Nat \models \alpha(H(\mathbf{P}), \mathbf{P})$.

Proof. The implication (1)$\Rightarrow$ (2) follows from Theorem 1.1 and the fact that every ultimately periodic predicate is definable by an *MLO* formula. The implication (2)$\Rightarrow$(3) is trivial.

The implication (3)$\Rightarrow$(1) is derived as follows. Let $\alpha(X, P)$ be $\forall t \big(X(t) \leftrightarrow P(t+1) \big)$. Note $Nat \models \exists X \alpha(X, \mathbf{P})$ for every $\mathbf{P} \subseteq Nat$. Therefore, if $\mathbf{P}$ satisfies selection condition, then there is C-operator $H : \{0,1\}^\omega \to \{0,1\}^\omega$ such that $Nat \models \alpha(H(\mathbf{P}), \mathbf{P})$.

Assume that a finite state Moore automaton $\mathcal{A}$ computes H and has n states. We are going to show that $\mathbf{P}$ is ultimately periodic with period at most $2n + 1$. For $i \in Nat$ let a_i be one if $i \in \mathbf{P}$ and a_i be zero otherwise. Let $q_0 q_1 \ldots q_{2n+1} \ldots$ be the sequence states passed by $\mathcal{A}$ on the input $a_0 a_1 \ldots a_{2n+1} \ldots$. There are $i < j < 2n$ such that $a_i = a_j$ and $q_i = q_j$. Observe that $q_{i+1} = \delta_{\mathcal{A}}(q_i, a_i) = \delta_{\mathcal{A}}(q_j, a_j) = q_{j+1}$ and $a_{i+1} = out_{\mathcal{A}}(q_i, a_i) = out_{\mathcal{A}}(q_j, a_j) = a_{j+1}$. And by induction we get that $q_{i+m} = q_{j+m}$ and $a_{i+m} = a_{j+m}$ for all $m \in Nat$. Therefore, $\mathbf{P}$ is an ultimately periodic with a period $j - i < 2n$. $\square$

Note that this theorem does not imply that Problem 4 is computable only for ultimately periodic predicates. The next theorem can be established by the same arguments.

Theorem 5.2. *The following conditions are equivalent and imply computability of Problem 5:*

(1) $\mathbf{P}$ *is ultimately periodic.*
(2) *For every MLO formula $\psi(X, Y, P)$ either there is a finite state C-operator F such that $Nat \models \forall X \psi(X, F(X), \mathbf{P})$ or there is a finite state SC-operator G such that $Nat \models \forall Y \neg \psi(G(Y), Y, \mathbf{P})$. Moreover, it is decidable which of these cases holds and the corresponding operator is computable from ψ.*

6. Parity Games on Graphs and the Synthesis Problem

In subsection 6.1, we provide standard definitions and facts about infinite two-player perfect information games on graphs. In [BL69], a reduction of the Church synthesis problem to infinite two-player games on finite graphs was provided. In subsection 6.2, we provide a reduction of the Church synthesis problem with parameters to infinite two-player games on infinite graphs; this reduction is "uniform" in the parameters. The main definability results needed for the proof of Theorem 2.3 are given in Sect. 6.3 and Sect. 6.4.

6.1. **Parity Games on Graphs.** We consider here two-player perfect information games, played on graphs, in which each player chooses, in turn, a vertex adjacent to a current vertex. The presentation is based on [PP04].

A (directed) bipartite graph $G = (V_1, V_2, E)$ is called a *game arena* if the outdegree of every vertex is at least one. If G is an arena, a game on G is defined by an initial node $v_{init} \in V_1$ and a set of winning ω-paths $\mathcal{F}$ from this node.

Player I plays on vertices in V_1 and Player II on vertices in V_2. A play from a node $v_1 = v_{init}$ is an infinite path $v_1 v_2 \ldots v_i \ldots$ in G formed by the two players starting from the initial position v_1. Whenever the current position v_i belongs to V_1 (respectively V_2), then Player I (respectively, Player II) chooses a successor node v_{i+1} such $E(v_i, v_{i+1})$. Since the graph is bipartite, Player I plays at the odd positions ($v_{2i+1} \in V_i$) and Player II plays at the even positions ($v_{2i} \in V_2$). Player I wins if the play belongs to $\mathcal{F}$.

A strategy f for Player I (Player II) is a function which assigns to every path of even (respectively, odd) length a node adjacent to the last node of the path. A play $v_{init} v_2 v_3 \ldots$ is played according to a strategy f_1 of Player I (strategy f_2 of Player II) if for every prefix $\pi = v_{init} v_2 \ldots v_n$ of even (respectively, odd) length $v_{n+1} = f_1(\pi)$ (respectively, $v_{n+1} = f_2(\pi)$). A strategy is winning for Player I (respectively, for Player II) if all the plays played according to this strategy are in $\mathcal{F}$ (respectively, in the complement of $\mathcal{F}$). A strategy is *memoryless* if it depends only on the last nodes in the path.

Parity games are games on graphs in which the set of winning paths are defined by parity conditions. More precisely, let $G = (V_1, V_2, E)$ be a game arena and let $c : V_1 \cup V_2 \to \{0, 1, \ldots m\}$ be a coloring.

Let $\rho = v_1 v_2 \ldots$ be a play. With such a play ρ, we associate the set of colors C_ρ that appear infinitely many times in the ω-sequence $col(v_1) col(v_2) \ldots$; a play ρ is winning for Player I if the minimal element of C_ρ is odd. The following theorem due to Emerson/Julta and Mostowski (see, [EJ91, GTW02, PP04]) is fundamental:

Theorem 6.1. *In a parity game, one of the players has a memoryless winning strategy.*

6.2. **Games and the Church Synthesis Problem.** Let $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \delta_\mathcal{A}, q_{init}, col \rangle$ be a deterministic parity automaton over the alphabet $\Sigma = \{0, 1\} \times \{0, 1\} \times \{0, 1\}$, let $R_\mathcal{A} \subseteq \{0, 1\}^\omega \times \{0, 1\}^\omega \times \{0, 1\}^\omega$ be the relation defined by $\mathcal{A}$ and let $\mathbf{P}$ be a subset of *Nat*. We will define a parity game $G_{\mathcal{A}, \mathbf{P}}$ such that

(1) Player I has a winning strategy in $G_{\mathcal{A}, \mathbf{P}}$ iff there is a SC-operator $G : \{0, 1\}^\omega \to \{0, 1\}^\omega$ such that $\neg R_\mathcal{A}(G(Y), Y, \mathbf{P})$ holds for every Y.

(2) Player II has a winning strategy in $G_{\mathcal{A}, \mathbf{P}}$ iff there is a C-operator $F : \{0, 1\}^\omega \to \{0, 1\}^\omega$ such that $R_\mathcal{A}(X, F(X), \mathbf{P})$ holds for every X.

The arena $G(V_1, V_2, E)$ of $G_{\mathcal{A}, \mathbf{P}}$ is defined as follows:

Nodes: $V_1 = \mathcal{Q} \times Nat$ and $V_2 = \mathcal{Q} \times \{0, 1\} \times Nat$.

Edges from V_1: From $\langle q, n \rangle \in V_1$ two edges exit; one to $\langle q, 0, n \rangle \in V_2$, and the second to $\langle q, 1, n \rangle \in V_2$. We will assign labels to these edges. The first one will be labeled by 0 and the second one will be labeled by 1. These edge labels play no role in the game on our graph; however, it will be convenient to refer to them later.

Edges from V_2: From $\langle q, a, n \rangle \in V_2$ two edges exit defined as follows: let c be 1 if $n \in \mathbf{P}$ and 0 if $n \notin \mathbf{P}$; and for $b \in \{0, 1\}$ let q_b be $\delta_\mathcal{A}(q, \langle a, b, c \rangle)$. One edge from $\langle q, a, n \rangle$ is connected to $\langle q_0, n + 1 \rangle$, and the second one to $\langle q_1, n + 1 \rangle$. We label the first edge by 0, and the second one by 1.

The color of a node of the arena is defined by the color of its automaton's component, i.e., $c(\langle q, n \rangle) = c(\langle q, a, n \rangle = col(q)$.

The node $\langle q_{init}, 0 \rangle$ is the initial node of the game.

Every node of the game graph for $G_{\mathcal{A}, \mathbf{P}}$ has two successors. The subsets of V_1 (respectively, of V_2) can be identified with the memoryless strategies of Player I (respectively, of

Player II). For a subset $U_1 \subseteq V_1$, the corresponding memoryless strategy f_{U_1} is defined as

$$f_{U_1}(\langle q, n \rangle) = \begin{cases} \langle q, 1, n \rangle & \text{if } \langle q, n \rangle \in U_1 \\ \langle q, 0, n \rangle & \text{otherwise} \end{cases}$$

In other words, for $v \in V_1$ the strategy f_{U_1} chooses the nodes reachable from v by the edge with the label $U_1(v)$.

Observation 6.1 (bijection between the memoryless strategies and the subset of nodes)**.** The function that assigns to every subset U of V_1 the strategy f_U for Player I is a bijection between the set of memoryless strategies for Player I and the subset of V_1. Similarly, the function that assigns to every subset U of V_2 the strategy f_U is a bijection between the set of memoryless strategies for Player II and the subset of V_2.

A subset $U_1 \subseteq V_1$, induces a function $h_{U_1} : \{0,1\}^* \to V_2$ and a SC-operator $F_{U_1} : \{0,1\}^\omega \to \{0,1\}^\omega$. First, we provide the definition for h_{U_1}, and later for F_{U_1}.

Let G_{U_1} be the subgraph of $G_{\mathcal{A},\mathbf{P}}$, obtained by removing from every node $v \in V_1$ the edge labelled by $\neg U_1(v)$, and removing the label from the other edge exiting v. In this graph, every V_1 node has outdegree one, and every V_2 node has two exiting edges; one is labeled by 0 and the other is labeled by 1. For every π in $\{0,1\}^*$ there is a unique path from $\langle q_{init}, 0 \rangle$ to a state $v_2 \in V_2$ such that π is the sequence of labels on the edges of this path; this node v_2 is h_{U_1} image of π.

Now a SC-operator $F_{U_1} : \{0,1\}^\omega \to \{0,1\}^\omega$ induced by U_1 is defined as follows. Let $\pi = b_0 b_1 \ldots$ be an ω-string. There is a unique ω-path ρ from $\langle q_{init}, 0 \rangle$ in G_{U_1} such that π is the sequence labels on the edges of this path. Let $v_1 v_2 \ldots$ be the sequence of V_1 nodes on ρ and let $a_i = 1$ if $v_i \in U_1$ and 0 otherwise. The ω sequence $a_0 a_1 \ldots$ is defined as the F_{U_1} image of π.

Similarly, $U_2 \subseteq V_2$ induces a function $h_{U_2} : \{0,1\}^+ \to V_1$ and C-operator F_{U_2}.

Below we often use "a function F corresponds to a set U" as synonym "a set U induces a function F".

The properties of the above constructions are summarized as follows:

Lemma 6.2. (1) *Let U_1 be a subset of V_1. The memoryless strategy defined by U_1 is winning for Player I in $G_{\mathcal{A},\mathbf{P}}$ iff $\neg R_{\mathcal{A}}(F_{U_1}(Y), Y, \mathbf{P})$ holds for every Y.*
(2) *Let U_2 be a subset of V_2. The memoryless strategy defined by U_2 is winning for Player II in $G_{\mathcal{A},\mathbf{P}}$ iff $R_{\mathcal{A}}(X, F_{U_2}(X), \mathbf{P})$ holds for every X.*
(3) *Let $\varphi(X, Y, Z)$ be an MLO formula equivalent to $\mathcal{A}$, let ψ be $\varphi(X, Y, P)$ and let M be $\langle Nat, <, \mathbf{P} \rangle$. The memoryless strategy defined by U is winning for Player I (respectively, Player II) in $G_{\mathcal{A},\mathbf{P}}$ iff the operator induced by U is a winning strategy for Player I (respectively, for Player II) in $\mathcal{G}_\psi^M$.*

Our next objective is to show that the set of memoryless winning strategies and the operator induced by a memoryless strategy are *MLO*-definable in $\langle Nat, <, \mathbf{P} \rangle$. For this purpose, we show in 6.3 that these are definable in the monadic-second order logic for the structure appropriate for the game graph $G_{\mathcal{A},\mathbf{P}}$. Then, in Sect. 6.4, we translate these definitions to *MLO* formulas over $\langle Nat, <, \mathbf{P} \rangle$.

6.3. Definability in the game structure.
The game arena $G_{\mathcal{A},\mathbf{P}}$ can be considered as a logical structure $M = M_{\mathcal{A},\mathbf{P}}$ for the signature $\tau_{\mathcal{A}} = \{R_i : i \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\}\} \cup \{Init, P, \prec$

$, E_0, E_1\}$, where R_i, $Init$ and P are unary predicates and $\prec$, , E_0, E_1 are binary predicates with the interpretation

$$R_i^M = \begin{cases} \{\langle q, j \rangle : j \in Nat\} & \text{for } i = q \in \mathcal{Q} \\ \{\langle q, a, j \rangle : j \in Nat\} & \text{for } i = \langle q, a \rangle \in \mathcal{Q} \times \{0, 1\} \end{cases}$$

$$P^M = \{\langle q, m \rangle : m \in \mathbf{P}\} \cup \{\langle q, a, m \rangle : a \in \{0, 1\} \text{ and } m \in \mathbf{P}\}$$

$$Init^M = \{\langle q_{init}, 0 \rangle\}$$

$$\mathbf{E}_0^M(v_1, v_2) \text{ (respectively, } \mathbf{E}_1^M(v_1, v_2)) \text{ holds}$$

iff there is an edge labeled by 0 (respectively, by 1) from v_1 to v_2.

$$v_1 \prec v_2 \text{ iff } v_1 = \langle i_1, j_1 \rangle \text{ and } v_2 = \langle i_2, j_2 \rangle \text{ and } j_1 < j_2.$$

The next lemma shows that the set of memoryless winning strategies is definable in $M_{\mathcal{A}, \mathbf{P}}$.

Lemma 6.3 (The set of memoryless winning strategies is definable in $M_{\mathcal{A}, \mathbf{P}}$). *Let $\mathcal{A}$ be a parity automaton.*

(1) *There is a monadic second-order formula $winSt_{G_\mathcal{A}}^I(X)$ such that $M_{\mathcal{A}, \mathbf{P}} \models winSt_{G_\mathcal{A}}^I(U)$ iff U corresponds to a memoryless winning strategy for Player I.*
(2) *There is a monadic second-order formula $winSt_{G_\mathcal{A}}^{II}(X)$ such that $M_{\mathcal{A}, \mathbf{P}} \models winSt_{G_\mathcal{A}}^{II}(U)$ iff U corresponds to a memoryless winning strategy for Player II.*
(3) *Moreover, $winSt_{G_\mathcal{A}}^I(X)$ and $winSt_{G_\mathcal{A}}^{II}(X)$ are computable from $\mathcal{A}$.*

Proof. We will formalize that "player I wins all the plays consistent with a memoryless strategy X".

A play is an infinite path that starts from the initial node. Note that the arena $G_{\mathcal{A}, \mathbf{P}}$ is an acyclic graph. Hence, we can formalize that "Z is the set of nodes of an infinite path that starts from a node v" as a formula $Play(v, Z)$ which is the conjunction of the following formulas:

(1) For every node u of Z there is a unique node $u' \in Z$ such that an edge from u enters u'.
(2) For every node $u \neq v$ of Z there is a unique node $u' \in Z$ such that an edge from u' enters u; there is no edge that enters from a node of Z into v.
(3) v is in Z.
(4) For every partition of Z into two non-empty set Z_1 and Z_2 there is an edge between a node in Z_1 and a node in Z_2.

The assertion "u is a Player I node" is formalized by the formula $Pos_1(u)$, defined as $\bigvee_{q \in \mathcal{Q}} R_q(u)$. Next, we formalize that "Z is the set of nodes of a play consistent with a strategy U of Player I" by the formula $Consis(U, Z)$ which is the conjunction of the following formulas:

(1) "U is a subset of Player I nodes" is formalized by

$$\forall u(u \in U \rightarrow Pos_1(u))$$

(2) "Z is the set of nodes of a play from the initial node":

$$\exists v(Init(v) \land Play(v, Z))$$

(3) "Z is consistent with U"

$$\forall z z' \in Z \big(Pos_1(z) \land (E_1(z, z') \lor E_1(z.z')) \big) \rightarrow (E_1(z, z') \leftrightarrow z \in U)$$

Assume that the coloring function of $\mathcal{A}$ assigns to the states numbers in the set $\{0, 1 \ldots, m\}$, and let Q_i $(i = 0, \ldots m)$ be the set of states of $\mathcal{A}$ which are colored by i. Color i appears infinitely often in a play with the set of nodes Z if $inf_i(Z)$ defined as $\forall z \exists z' \in Z(z \prec z' \wedge (\vee_{q \in Q_i} R_q(z')))$ holds. Hence, the formula $Even(Z)$ defined as

$$\bigvee_{k \leq m/2} \left(inf_{2k}(Z) \wedge \bigwedge_{j < 2k} \neg inf_{2k}(Z) \right)$$

holds for a play Z iff the minimal color that appears infinitely often in Z is even.

Finally, $winSt^I_{G_\mathcal{A}}(X)$ can be defined as $\forall Z(Consis(X, Z) \rightarrow \neg Even(Z))$.

Note that our construction of $winSt^I_{G_\mathcal{A}}(X)$ from $\mathcal{A}$ is algorithmic.

The formula $winSt^{II}_{G_\mathcal{A}}(X)$ is defined from $\mathcal{A}$ similarly.   □

Next, we will show that the operator $F_U : \{0, 1\}^\omega \rightarrow \{0, 1\}^\omega$ induced by a memoryless strategy U of one of the players is definable in $M_{\mathcal{A}, \mathbf{P}}$. The operator F_U maps ω-strings to ω strings. Therefore, we should agree how the ω-strings are encoded by subset of $M_{\mathcal{A}, \mathbf{P}}$.

Note that for $q \in \mathcal{Q}$ the set of elements in R_q ordered by $\prec$ is isomorphic to $\langle Nat, < \rangle$. Accordingly, we can represent the ω-strings by the subsets of R_{qinit}, and in the next lemma the operators from $\{0, 1\}^\omega$ to $\{0, 1\}^\omega$ are identified with corresponding functions from the set of subset of $R_{q_{init}}$ to the set of subset of $R_{q_{init}}$.

Lemma 6.4 (Definability of the operator induced by a memoryless strategy). *Let $\mathcal{A}$ be a parity automaton.*

(1) *There is a monadic second-order formula $\psi_I(X, Y, U)$ such that $M_{\mathcal{A}, \mathbf{P}} \models \psi_I(\mathbf{X}, \mathbf{Y}, \mathbf{U})$ iff $\mathbf{U}$ is a memoryless strategy for Player I in $G_{\mathcal{A}, \mathbf{P}}$, $\mathbf{X}, \mathbf{Y} \subseteq R_{q_{init}}$ and $\mathbf{X} = F_\mathbf{U}(\mathbf{Y})$, where $F_\mathbf{U}$ is the operator induced by $\mathbf{U}$.*

(2) *There is a monadic second-order formula $\psi_{II}(X, Y, U)$ such that $M_{\mathcal{A}, \mathbf{P}} \models \psi_{II}(\mathbf{X}, \mathbf{Y}, \mathbf{U})$ iff $\mathbf{U}$ is a memoryless strategy for Player II in $G_{\mathcal{A}, \mathbf{P}}$, $\mathbf{X}, \mathbf{Y} \subseteq R_{q_{init}}$ and $\mathbf{Y} = F_\mathbf{U}(\mathbf{X})$, where $F_\mathbf{U}$ is the operator induced by $\mathbf{U}$.*

(3) *Moreover, ψ_I and ψ_{II} are computable from $\mathcal{A}$.*   □

Proof. We just formalize in the monadic-second order logic the construction of F_U given in Sect. 6.2.

Let $Consis(U, Z)$ be the formula from the proof of Lemma 6.3 which expresses "Z is the set of nodes of a play consistent with a strategy U of Player I". We need to say that X (respectively, Y) is the sequence of edges[1] chosen by Player I (respectively, by Player II) in the play Z. It can be formalized by formula $moves(X, Y, Z)$ which is the conjunction of

(1) $X \subseteq R_{q_{init}} \wedge Y \subseteq R_{q_{init}}$ - X and Y "encodes" ω strings over $\{0, 1\}$.

(2) We can formalize that X is the sequence of edges chosen by Player I as follows. Let $u \in Z$ be a node of Player II. Then, u is $\langle q, 1, j \rangle$ iff $\langle q_{init}, j \rangle$ is in X:

$$\forall u \in Z \forall v \in X \left(Pos_{II}(u) \wedge \neg(u \prec v) \wedge \neg(v \prec u) \right) \rightarrow \left(v \in X \leftrightarrow \bigvee_{q \in \mathcal{Q}} u \in R_{q,1} \right)$$

(3) Similarly, we can formalize that Y is the sequence of edges chosen by Player II as follows. Let $u = \langle q, a, j \rangle \in Z$ and $u' = \langle q', j+1 \rangle \in Z$ and let c be 1 (respectively, c be 0) if $j \in \mathbf{P}$ (respectively, $j \notin \mathbf{P}$) and let b be 1 (respectively, b be 0) if $\langle q_{init}, j \rangle \in Y$ (respectively, $\langle q_{init}, j \rangle \notin Y$). Then, $q' = \delta_\mathcal{A}(q, \langle a, b, c \rangle)$.

___
[1]Strictly speaking, the sequence of labels of the edges.

Note that for each memoryless strategy U of Player I, we have

$$\forall Y \subseteq R_{q_{init}} \exists! Z \exists! X \, Consis(U, Z) \wedge moves(X, Y, Z)$$

. Finally, $\psi_I(X, Y, U)$ can be defined as $\exists Z \, Consis(U, Z) \wedge moves(X, Y, Z)$.
   $\psi_{II}(X, Y, U)$ is defined in a similar way. $\square$

6.4. **Interpretation of $M_{\mathcal{A},\mathbf{P}}$ in the structure $\langle Nat, < \mathbf{P} \rangle$.** Every set S of nodes in $M_{\mathcal{A},\mathbf{P}}$ corresponds to the tuple $\langle \ldots, W_i, \ldots \rangle$ where $i \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\})$ of subsets of Nat, such that $\langle q, m \rangle \in S$ iff $m \in W_q$ and $\langle q, a, m \rangle \in S$ iff $m \in W_{\langle q,a \rangle}$.
   The proof of the following lemma shows that there is an interpretation of the structure $M_{\mathcal{A},\mathbf{P}}$ in the structure $\langle Nat, < \mathbf{P} \rangle$.

Lemma 6.5. *For every formula $\psi(X^1, \ldots X^k)$ in the second order monadic logic over the signature $\tau_{\mathcal{A}}$ with free monadic variables $X^1, \ldots, X^k$ there is a formula $\varphi(Y, \overrightarrow{Z^1} \ldots, \overrightarrow{Z^k})$, where $\overrightarrow{Z^j}$ is a tuple of monadic variables $\{Z_i^j : i \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\}\}$, such that for every $\mathbf{P} \subseteq Nat$ and a tuple $\langle \ldots, W_i^j, \ldots \rangle$ where $j \in \{1, \ldots, k\}$ and $i \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\})$ of subsets of Nat the following equivalence holds:*

$$\langle Nat, < \rangle \models \varphi(\mathbf{P}, \ldots, W_i^j, \ldots), \; iff \; M_{\mathcal{A},\mathbf{P}} \models \psi(S^1, \ldots, S^k)$$

where S^j is the subset of nodes in $G_{\mathcal{A},\mathbf{P}}$, which corresponds to $\langle \ldots, W_i^j, \ldots \rangle$.
Moreover, there is an algorithms that computes φ from ψ.

Proof. The proof proceeds by the structural induction. It is more convenient to consider the first-order version of the monadic second-order logic (see, Subsection 4.3).
   Basis - Atomic formulas.

- $Sing(X)$ is translated as

$$\bigvee_{i \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\}} Sing(Z_i) \wedge \bigwedge_{i \neq i' \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\}} (Sing(Z_i) \rightarrow Empty(Z_{i'})),$$

   where $Empty(W)$ is a shorthand for $\forall W'(W \subseteq W')$.
- $X^1 \subseteq X^2$ is translated as

$$\bigwedge_{i \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\}} Z_i^1 \subseteq Z_i^2$$

- $X^1 \prec X^2$ is translated as the conjunction of the following:
  (1) one of the Z_i^1 ($i \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\}$) is singleton and all the others are empty.
  (2) one of the Z_i^2 ($i \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\}$) is singleton and all the others are empty.
  (3) if $Z_{i_1}^1$ and $Z_{i_2}^2$ are singletons, then $Z_{i_1}^1 < Z_{i_2}^2$, i.e., the (unique) natural number which is in $Z_{i_1}^1$ is less than the natural number in $Z_{i_2}^2$.
- Other relations in $\tau_{\mathcal{A}}$ are translated similarly just by the straightforward formalization of our definition of the game arena, e.g., $E_0(X^1, X^2)$ is translated as the conjunction of the following:
  (1) one of the Z_i^1 ($i \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\}$) is singleton and all others are empty.
  (2) one of the Z_i^2 ($i \in \mathcal{Q} \cup \mathcal{Q} \times \{0,1\}$) is singleton and all others are empty.
  (3) if Z_q^1 (for some $q \in \mathcal{Q}$) is singleton, then $Z_{\langle q,0 \rangle}^2$ is singleton and both of them contain the same natural number.

(4) if $Z^1_{\langle q,a \rangle}$ (for some $q \in \mathcal{Q}$ and $a \in \{0, 1, \}$) is singleton, and the unique number n from $Z^1_{\langle q,a \rangle}$ is in $\mathbf{P}$ (i.e., $Z^1_{\langle q,a \rangle} \subseteq \mathbf{P}$) and q' is be $\delta_{\mathcal{A}}(q, \langle a, 0, 1 \rangle)$, then $Z^2_{q'}$ is singleton and the unique element of $Z^2_{q'}$ is the successor of the unique element of $Z^1_{\langle q,a \rangle}$.

(5) if $Z^1_{\langle q,a \rangle}$ (for some $q \in \mathcal{Q}$ and $a \in \{0, 1, \}$) is singleton, and the unique number n from $Z^1_{\langle q,a \rangle}$ is not in $\mathbf{P}$ (i.e., $\neg(Z^1_{\langle q,a \rangle} \subseteq \mathbf{P})$) and q' is $\delta_{\mathcal{A}}(q, \langle a, 0, 0 \rangle)$, then $Z^2_{q'}$ is singleton, and the unique element of $Z^2_{q'}$ is the successor of the unique element of $Z^1_{\langle q,a \rangle}$.

Inductive step. If φ_j is the translation of ψ_j for $j = 1, 2$, then $\psi_1 \wedge \psi_2$ is translated as $\varphi_1 \wedge \varphi_2$ and $\neg \psi_1$ is translated as $\neg \varphi_1$.

Finally, if $\varphi(Y, \overrightarrow{Z^1} \ldots, \overrightarrow{Z^k})$ is the translation of $\psi(X^1, \ldots X^k)$, then $\exists X^1 \psi$ is translated as $\exists \overrightarrow{Z^1} \varphi$. $\square$

The next two lemmas use the interpretation of $M_{\mathcal{A},\mathbf{P}}$ in $\langle Nat, < \mathbf{P} \rangle$ to show *MLO* definability in $\langle Nat, < \mathbf{P} \rangle$ of the set of memoryless winning strategies and the operator induced by a memoryless strategy.

Lemma 6.6 (The set of memoryless winning strategies is definable in $\langle Nat, < \mathbf{P} \rangle$).
Let $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \delta_{\mathcal{A}}, q_{init}, col \rangle$, be a deterministic parity automaton over the alphabet $\Sigma = \{0, 1\} \times \{0, 1\} \times \{0, 1\}$.

(1) *There is an MLO formula $WinSt^I_{\mathcal{A}}(Z_1, \ldots, Z_{|\mathcal{Q}|}, Z)$ such that for every $\mathbf{P} \subseteq Nat$ and $W_1, \ldots, W_{|\mathcal{Q}|} \subseteq Nat$:*

$$Nat \models WinSt^I_{\mathcal{A}}(W_1, \ldots, W_{|\mathcal{Q}|}, \mathbf{P})$$

iff the corresponding subset $U \subseteq \mathcal{Q} \times Nat$ defines a memoryless winning strategy for Player I in $G_{\mathcal{A},\mathbf{P}}$.

(2) *There is an MLO formula $WinSt^{II}_{\mathcal{A}}(Z_1, \ldots, Z_{|\mathcal{Q}|}, Z'_1, \ldots, Z'_{|\mathcal{Q}|}, Z)$ such that for every $\mathbf{P} \subseteq Nat$ and $W_1, \ldots, W_{|\mathcal{Q}|}, W'_1, \ldots, W'_{|\mathcal{Q}|} \subseteq Nat$:*

$$Nat \models WinSt^{II}_{\mathcal{A}}(W_1, \ldots, W_{|\mathcal{Q}|}, W'_1, \ldots, W'_{|\mathcal{Q}|}, \mathbf{P})$$

iff the corresponding subset $U \subseteq \mathcal{Q} \times \{0, 1\} \times Nat$ defines a memoryless winning strategy for Player II in $G_{\mathcal{A},\mathbf{P}}$.

(3) *Moreover, there is an algorithm that computes formulas $WinSt^I_{\mathcal{A}}$ and $WinSt^{II}_{\mathcal{A}}$ from $\mathcal{A}$.*

Proof. Follows from Lemma 6.3 and Lemma 6.5. Let $winSt^I_{G_{\mathcal{A}}}(X)$ be the formula constructed in Lemma 6.3 which defines (in $M_{\mathcal{A},\mathbf{P}}$) the set of memoryless winning strategies of the first player. Let $\psi(Y, \vec{Z}^1)$ be its translation, as in Lemma 6.5, where $\overrightarrow{Z^1}$ is the tuple of variables indexed by $i \in \mathcal{Q} \cup \mathcal{Q} \times \{0, 1\}$. Note that since $\overrightarrow{Z^1}$ defines (in $M_{\mathcal{A},\mathbf{P}}$) a strategy of Player I, $Z_{\langle q,a \rangle}$ for $q \in \mathcal{Q}$ and $a \in \{0, 1\}$ should be interpreted as the empty set. The formula $WinSt^I_{\mathcal{A}}$ is obtained from ψ by replacing $Z_{\langle q,a \rangle}$ for $q \in \mathcal{Q}$ and $a \in \{0, 1\}$ by the empty set and replacing Y by Z.

$WinSt^{II}_{\mathcal{A}}$ is defined in a similar way. $\square$

From Lemma 6.4 and Lemma 6.5, by the same arguments, we can derive the following Lemma:

Lemma 6.7 (The operator induced by a memoryless strategy is definable in $\langle Nat, < \mathbf{P} \rangle$).
Let $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \delta_{\mathcal{A}}, q_{init}, col \rangle$, be a deterministic parity automaton over the alphabet $\Sigma = \{0, 1\} \times \{0, 1\} \times \{0, 1\}$.

(1) *There is an MLO formula $op_{\mathcal{A}}^I(X, Y, Z_1, \ldots, Z_{|\mathcal{Q}|}, Z)$ which has the following property:*
   Let $\mathbf{P} \subseteq Nat$ and $W_1, \ldots, W_{|\mathcal{Q}|} \subseteq Nat$, and let F_U be the C-operator, induced by Player I memoryless strategy $U \subseteq \mathcal{Q} \times Nat$ which corresponds to $\langle W_1, \ldots, W_{|\mathcal{Q}|}\rangle$. Then, $Nat \models op_{\mathcal{A}}^I(X, Y, W_1, \ldots, W_{|\mathcal{Q}|}, \mathbf{P})$ iff $X = F_U(Y)$.

(2) *There is an MLO formula $op_{\mathcal{A}}^{II}(X, Y, Z_1, \ldots, Z_{|\mathcal{Q}|}, Z_1', \ldots, Z_{|\mathcal{Q}|}', Z)$ which has the following property:*
   Let $\mathbf{P} \subseteq Nat$ and $W_1, \ldots, W_{|\mathcal{Q}|}, W_1', \ldots, W_{|\mathcal{Q}|}' \subseteq Nat$ and let F_U be the C-operator, induced by Player II memoryless strategy $U \subseteq \mathcal{Q} \times \{0, 1\} \times Nat$ which corresponds to $\langle W_1, \ldots, W_{|\mathcal{Q}|}, W_1', \ldots, W_{|\mathcal{Q}|}'\rangle$. Then,

$$Nat \models op_{\mathcal{A}}^{II}(X, Y, W_1, \ldots, W_{|\mathcal{Q}|}, W_1', \ldots, W_{|\mathcal{Q}|}', \mathbf{P}) \text{ iff } Y = F_U(X).$$

(3) *Moreover, there is an algorithm that computes formulas $op_{\mathcal{A}}^I$ and $op_{\mathcal{A}}^{II}$ from $\mathcal{A}$.*

7. Proof of Theorem 2.3

We are now almost ready to prove Theorem 2.3. We will show an algorithm that, given a formula $\varphi(X, Y, P)$, constructs

(1) a sentence $WIN_\varphi^{II}(P)$ and
(2) formulas $St_\varphi^I(X, Y, P)$ and $St_\varphi^{II}(X, Y, P)$

such that for every structure $M = \langle Nat, <, \mathbf{P}\rangle$, Player II has a winning strategy in the games $\mathcal{G}_\varphi^M$ iff $M \models WIN_\varphi^{II}$. If Player II (respectively, Player I) has a winning strategy, then $St_\varphi^{II}(X, Y, P)$ (respectively, $St_\varphi^I(X, Y, P)$) defines his winning strategy.

Let $\varphi(X, Y, P)$ be a formula. We are going to construct $WIN_\varphi^{II}(P)$, as follows. First, let $\varphi'(X, Y, Z)$ be a formula obtained from φ by replacing all the occurrences of P by a fresh variable Z.

Let $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \delta_{\mathcal{A}}, q_{init}, col\rangle$ be a deterministic parity automaton over the alphabet $\Sigma = \{0, 1\} \times \{0, 1\} \times \{0, 1\}$, which is equivalent to φ'. Let $WinSt_{\mathcal{A}}^I(Z_1, \ldots, Z_{|\mathcal{Q}|}, Z)$ be constructed from $\mathcal{A}$, as in Lemma 6.6. Player I has a winning strategy iff

$$M \models \exists Z_1 \ldots \exists Z_{|\mathcal{Q}|} \, WinSt_{\mathcal{A}}^I(Z_1, \ldots, Z_{|\mathcal{Q}|}, P) \ .$$

Finally, $WIN_\varphi^{II}(P)$ can be defined as $\neg \exists Z_1 \ldots \exists Z_{|\mathcal{Q}|} \, WinSt_{\mathcal{A}}^I(Z_1, \ldots, Z_{|\mathcal{Q}|}, P)$.

The correctness of the construction follows from Lemma 6.6 and Lemma 6.2.

In order to construct $St_\varphi^I(X, Y, P)$, $St_\varphi^{II}(X, Y, P)$, we need the following definition and Theorem.

Definition 7.1 (Selection). Let $\varphi(\bar{Y})$, $\psi(\bar{Y})$ be formulas and C a class of structures. We say that ψ *selects* (or, is a *selector* for) φ over C iff for every $M \in C$:

(1) $M \models \exists^{\leq 1} \bar{Y} \psi(\bar{Y})$,
(2) $M \models \forall \bar{Y}(\psi(\bar{Y}) \rightarrow \varphi(\bar{Y}))$, and
(3) $M \models \exists \bar{Y} \varphi(\bar{Y}) \rightarrow \exists \bar{Y} \psi(\bar{Y})$.

Here, $\bar{Y}$ is a tuple of distinct variables and "$\exists^{\leq 1} \bar{Y} \ldots$" stands for "there exists at most one...". The definition can be rephrased as ψ is a selector of φ over C iff for each $M \in C$, if φ is satisfiable in M, then it is satisfiable by the (unique) tuple defined by ψ.

We say that C has the *selection property* iff every formula φ has a selector ψ over C.

Theorem 7.2. *The class of labelled ω-chains has the selection property. Moreover, there is an algorithm which constructs for every φ a formula ψ which selects φ over the class of labelled ω-chains.*

Theorem 7.2 was proved in [Rab07]. Its version without the "Moreover" clause was stated without proof in [LS98].

Now, we are ready to define $St_\varphi^I(X, Y, P)$ and $St_\varphi^{II}(X, Y, P)$.

Let $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \delta_{\mathcal{A}}, q_{init}, col \rangle$ and $WinSt_{\mathcal{A}}^I(Z_1, \ldots, Z_{|\mathcal{Q}|}, Z)$ be as in the construction of $WIN_\varphi^{II}(P)$ above.

Construct $\alpha(Z_1, \ldots, Z_{|\mathcal{Q}|}, P)$ as a selector for $WinSt_{\mathcal{A}}^I(Z_1, \ldots, Z_{|\mathcal{Q}|}, P)$ over the class $\{\langle Nat, <, \mathbf{P} \rangle : \mathbf{P} \subseteq Nat\}$ of structures. If $WinSt_{\mathcal{A}}^I(Z_1, \ldots, Z_{|\mathcal{Q}|}, P)$ is satisfiable in M, then every tuple which satisfies it corresponds to a memoryless winning strategy of Player I in the parity game $G_{\mathcal{A}, \mathbf{P}}$. In particular, the tuple defined by α describes a memoryless winning strategy of Player I in the parity game $G_{\mathcal{A}, \mathbf{P}}$. Let $op_{\mathcal{A}}^I$ be as in Lemma 6.6. Then, $\exists Z_1 \ldots \exists Z_{|\mathcal{Q}|} \big(\alpha(Z_1, \ldots, Z_{|\mathcal{Q}|}, P) \wedge op_{\mathcal{A}}^I(X, Y, Z_1, \ldots, Z_{|\mathcal{Q}|}) \big)$ defines in M a SC-operator F, induced by a memoryless winning strategy of Player I in the parity game $G_{\mathcal{A}, \mathbf{P}}$.

Hence, by Lemma 6.2, $\neg R_{\mathcal{A}}(F(Y), Y, \mathbf{P})$ holds for every Y. Therefore, by the definitions of $\mathcal{A}$ and $R_{\mathcal{A}}$, we have

$$\langle Nat, < \rangle \models \forall Y \neg \psi((F(Y), Y, \mathbf{P}).$$

Hence F, defined by $\exists Z_1 \ldots \exists Z_{|\mathcal{Q}|} \big(\alpha \wedge op_{\mathcal{A}}^I(X, Y, Z_1, \ldots, Z_{|\mathcal{Q}|}) \big)$, is a winning strategy for Player I in $\mathcal{G}_\varphi^M$.

The formula $St_\varphi^{II}(X, Y, P)$ which defines a winning strategy for Player II is constructed similarly.

8. Conclusion and Related Work

We investigated the Church synthesis problem with parameters. We provided the necessary and sufficient conditions for the computability of Synthesis problems 1-3.

The conditions of Theorem 5.1 and Theorem 5.2 are sufficient, but are not necessary for the computability of Synthesis problems 4-5. For example, let $\mathbf{Fac} = \{n! : n \in Nat\}$ be the set of factorial numbers. We can show that Problems 4-5 are computable for this predicate $\mathbf{Fac}$ [Rab06].

It is an open question whether the decidability of $\langle Nat, <, \mathbf{P} \rangle$ is a sufficient condition for the computability of Synthesis problems 4-5.

We proved that the definability and synthesis parts of the Büchi and Landweber theorem hold for all expansions of ω by unary predicates.

Büchi proved that the *MLO*-theory of any countable ordinal is decidable. After stating their main theorem, Büchi and Landweber write:

> "We hope to present elsewhere a corresponding extension of [our main theorem] from ω to any countable ordinal."

However, despite the fundamental role of the Church problem, no such extension is even mentioned in a later book by Büchi and Siefkes [BS73], which summarizes the theory of finite automata and *MLO* over words of countable length.

In [Rab06a], we proved that the determinacy and decidability parts of the Büchi and Landweber theorem hold for all countable ordinals; however, its definability and synthesis parts hold for an ordinal α iff $\alpha < \omega^\omega$ [RS06].

In [RT07], the Church Problem for fragments of MLO was considered. First-order MLO formulas are MLO formulas without the second-order quantifiers. In [RT06], it was proved that if $\psi(X, Y)$ is a first-order MLO formula, then one of the players has a first-order definable winning strategy in $\mathcal{G}_\psi^\omega$. Similar results were obtained for several interesting fragments of MLO. However, it is an open question whether these results hold when parameters are added.

Kupferman and Vardi [KV97] considered the synthesis problem with incomplete information for the specifications described by temporal logics LTL and CTL*. Their main results deal with the complexity of this synthesis problem. The decidability of the synthesis problem with incomplete information for LTL (respectively, for CTL*) can be easily derived from the Büchi-Landweber (respectively, Rabin) theorem. It seems that there are no interesting connections between the synthesis problems with incomplete information and the synthesis problems with parameters considered here.

In [RT98] a program for the relativization of finite automata theory was proposed. Our results can be seen as the first step in this direction. This step corresponds to the case where oracles are C-operators without inputs.

In the rest of this section we comment on Rabin's proof of the Church synthesis Problem, discuss possibilities of extending it to the Church synthesis Problem with parameters, and state some open questions.

Rabin [Rab72] provided an alternative proof for computability of the Church synthesis problem. This proof uses an automata on infinite trees as a natural tool for treating the synthesis problem. A C-operator $F:\{0,1\}^\omega \to \{0,1\}^\omega$ can be represented by a labelled infinite full binary tree $\langle T_2, <, \mathbf{S} \rangle$, where $\mathbf{S}$ is a subset of the tree nodes. Namely, the branches of the tree represent $X \in \{0,1\}^\omega$ and the sequence of values assigned by $\mathbf{S}$ to the nodes along the branch X represents $F(X) = Y \in \{0,1\}^\omega$. Also, the fact that $\mathbf{S}$ represents a C-operator F which uniformizes $\varphi(X, Y)$ can be expressed by an MLO formula $\psi(Z)$ (computable from $\varphi(X, Y)$): $T_2 \models \psi(\mathbf{S})$ iff $Nat \models \forall \varphi(X, F_\mathbf{S}(X))$, where $F_\mathbf{S}$ is the C-operator that corresponds to $\mathbf{S}$. Hence, the question whether there exists a C-operator which uniformizes φ is reduced to the problem whether $T_2 \models \exists Z \psi(Z)$. Now, the Rabin basis theorem states that if $T_2 \models \exists Z \psi(Z)$ then there is a regular subset $\mathbf{S} \subseteq T_2$ such that $T_2 \models \psi(\mathbf{S})$. The C-operator which corresponds to a regular set $\mathbf{S}$ is computable by a finite state automaton. Hence, the Büchi and Landweber theorem is obtained as a consequence of the decidability of the monadic logic of order of the full binary tree and the basis theorem.

One could try to apply the Rabin method to the Church synthesis problem with parameters. The reduction which is similar to Rabin's reduction shows that for every $\varphi(X, Y, P)$ there is a sentence $WIN_\varphi^{II}(Q)$ such that for every structure $M = \langle Nat, <, \mathbf{P} \rangle$ Player II wins the games $\mathcal{G}_\varphi^M$ iff

$$\langle T_2, <, \mathbf{Q} \rangle \models WIN_\varphi^{II}(Q),$$

where a node is in $\mathbf{Q}$ if its distance from the root is in $\mathbf{P}$. Moreover, $WIN_\varphi^{II}(Q)$ is computable from φ (cf. Theorem 2.3).

Hence, the decidability of Problem 1$'$ for $\mathbf{P} \subseteq Nat$ (recall that this is Problem 1 without its constructive part, see Sect. 3) is reduced to the decidability of the monadic theory of the labelled full binary tree $\langle T_2, <, \mathbf{Q} \rangle$, where a node is in $\mathbf{Q}$, if its distance from the root is in $\mathbf{P}$. The decidability of the latter problem can be reduced by Shelah-Stupp Muchnick Theorem [Shel75, Wal02, Th03] to the decidability of $\langle Nat, <, \mathbf{P} \rangle$. Now, in order to establish computability of problems 1-3, one can try to prove the basis theorem for $\langle T_2, <, \mathbf{Q} \rangle$.

Unfortunately, arguments similar to the proof of Theorem 5.1 show that for $\mathbf{P}, \mathbf{Q}$, as above, the following are equivalent:

(1) $\mathbf{P}$ is ultimately periodic.
(2) For every $\psi(Z, Q)$ such that $T_2 \models \exists Z \psi(Z, \mathbf{Q})$ there is a finite state operator $F(Y, U)$ such that the set which corresponds to the C-operator $\lambda X F(X, \mathbf{P})$ satisfies $\psi(Z, \mathbf{Q})$.

In case selector over $\langle T_2, <, \mathbf{Q} \rangle$ is computable, it would be easy to derive the computability of Problems 1-3. However, even the following is open

Open Question - Selectoion property for an expansion of T_2 by unary predicate.

> Is it true that for every $\mathbf{Q} \subseteq T_2$ every formula $\varphi(Y, Q)$ has a selector over $\langle T_2, <, \mathbf{Q} \rangle$, i.e., there is $\psi(Y, Q)$ such that (1) $\langle T_2, < .\mathbf{Q} \rangle \models \exists^{\leq 1} Y \psi(Y)$, (2) $\langle T_2, <, \mathbf{Q} \rangle \models \forall Y (\psi(Y) \to \varphi(Y))$, and (3) $\langle T_2, < .\mathbf{Q} \rangle \models \exists Y \varphi(Y) \to \exists \bar{Y} \psi(Y)$.

Note that this question asks whether every expansion of T_2 has the selection property and is different from the Rabin's uniformization problem over T_2, which asks whether every formula has a selector over the class of all the expansions of T_2 by unary predicates. The negative answer to the Rabin uniformization problem was obtained by Gurevich and Shelah [GS83] who proved that a formula $\varphi(Y, Q)$ which express that "if Q is not empty than Y is a singleton set which is a subset of Q" has no selector over the class of all the expansions of T_2.

We believe that if the answer to the above mentioned question is positive, then its proof is non-trivial.

Our proof of Theorem 1.2 implies that for $\mathbf{P} \subseteq Nat$ and $\mathbf{Q} \subseteq T_2$, where a node is in $\mathbf{Q}$ if its distance from the root is in $\mathbf{P}$, the following are equivalent:

(1) The monadic theory of $\langle Nat, <, \mathbf{P} \rangle$ is decidable,
(2) For every $\psi(Z, Q)$ such that $T_2 \models \exists Z \psi(Z, \mathbf{Q})$ there is a recursive $\mathbf{S} \subset T_2$ such that $T_2 \models \psi(\mathbf{S}, \mathbf{Q})$.

However, we do not know the answer to the following question:

Open Question: Are the following assertions equivalent?

(1) The monadic theory of $\langle T_2, < \mathbf{S} \rangle$ is decidable.
(2) For every $\psi(Z, U)$ such that $T_2 \models \exists Z \psi(Z, \mathbf{S})$ there is a recursive set $\mathbf{Q} \subset T_2$ such that $T_2 \models \psi(\mathbf{Q}, \mathbf{S})$.

Acknowledgments

References

[Bu60] 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.

[BL69] J. R. Büchi and L. H. Landweber. Solving sequential conditions by finitestate strategies. Transactions of the AMS, 138(27):295–311, 1969.

[BS73] J. R. Büchi, D. Siefkes, The Monadic Second-order Theory of all Countable Ordinals, Springer Lecture Notes 328 (1973), pp. 1-126.

[CT02] O. Carton and W.Thomas. The Monadic Theory of Morphic Infinite Words and Generalizations. Inf. Comput. 176(1), pp. 51-65, 2002.

[Ch69] Y. Choueka. Finite Automata on Infinite Structure. Ph.D Thesis, Hebrew University, 1970.

[ER66]   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.

[EJ91]   E. A. Emerson, C. S. Jutla: Tree Automata, Mu-Calculus and Determinacy (Extended Abstract) FOCS91: 368-377, 1991.

[GTW02] E. Grädel, W. Thomas and T. Wilke. Automata, Logics, and Infinite Games, LNCS 2500, 2002.

[GS83]   Y. Gurevich and S. Shelah. Rabin's uniformization problem. The Journal of Symbolic Logic, **48**:1105-1119, 1983.

[KV97]   O. Kupferman and M.Y. Vardi, Synthesis with incomplete information, In 2nd International Conference on Temporal Logic, pp 91–106, 1997.

[Mc66]   R. McNaughton, Testing and generating infinite sequences by a finite automaton, Information and Control 9 (1966), pp. 521-530.

[LS98]   S. Lifsches, S. Shelah, Uniformization and skolem functions in the class of trees, Jou. of Symolic Logic, Vol. 63(1) (Mar. 1998), pp. 103-127.

[PP04]   D. Perrin and J. E. Pin. Infinite Words Automata, Semigroups, Logic and Games. Pure and Applied Mathematics Vol 141 Elsevier, 2004.

[Rab72]   M. O. Rabin. Automata on Infinite Objects and Church's Problem Amer. Math. Soc. Providence, RI, 1972.

[Rab05]   A. Rabinovich. On decidability of monadic logic of order over the naturals extended by monadic predicates. 2005 Summer Meeting of the Association for Symbolic Logic, Logic Colloquium 05. The Bulletin of Symbolic Logic **12**:343-344, 2006.

[Rab06]   A. Rabinovich. The Church problem over ω expanded by factorial numbers. In preparation, 2006.

[Rab06a]   A. Rabinovich. The Church Problem for Countable Ordinals. Submitted, 2006.

[Rab07]   A. Rabinovich. On decidability of Monadic logic of order over the naturals extended by monadic predicates. Information and Computation, 2007.

[RT98]   A. Rabinovich and B.A. Trakhtenbrot. From Finite Automata toward Hybrid Systems Procceddings of Fundamentals of Computation Theory. Lecture Notes in Computer Science 1450, pp. 411-422, Springer, 1998.

[RS06]   A. Rabinovich and A. Shomrat. Selection in the Monadic Theory of Countable Ordinals. Submitted 2006.

[RT06]   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.

[RT07]   A. Rabinovich and W. Thomas. Logical Refinements of Church's Problem. In CSL 2007, LNCS 4646, 69-83, 2007.

[Rob58]   R. M. Robinson. Restricted Set-Theoretical Definitions in Arithmetic. In Proceedings of the AMS Vol. 9, No. 2. pp. 238-242, 1958.

[Sem84]   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.

[Se04]   O. Serre. Games With Winning Conditions of High Borel Complexity. In ICALP 2004, LNCS volume 3142, pp. 1150-1162, 2004.

[Shel75]   S. Shelah. The monadic theory of order. *Ann. of Math.* **102**:379–419, 1975.

[Sie75]   D. Siefkes. The recursive sets in certain monadic second order fragments of arithmetic. Arch. Math. Logik, pp71-80, 17(1975).

[Th75]   W. Thomas. Das Entscheidungsproblem für einige Erweiterungen der Nachfalger-Arithmetic. Ph. D. Thesis Albert-Ludwigs Universität, 1975.

[Th95]   W. Thomas. On the synthesis of strategies in infinite games. In STACS '95, LNCS vo. 900, pp. 1-13. 1995.

[Th03]   W. Thomas. Constructing infinite graphs with a decidable MSO-theory. In MFCS03, LNCS 2747, 2003.

[Trak61]   B. A. Trakhtenbrot. Finite automata and the logic of one-place predicates. (Russian version 1961). In AMS Transl. 59, 1966, pp. 23-55.

[Wal02]   I. Walukiewicz. Monadic second order logic on tree-like structures.TCS 1:275, pp 311-346, 2002.