A Curry-Howard Approach to Church's Synthesis*

Pierre Pradic[1] and Colin Riba[2]

1 **ENS de Lyon, Université de Lyon, LIP†, Lyon, France; and University of Warsaw, Faculty of Mathematics, Informatics and Mechanics, Warsaw, Poland**
`pierre.pradic@ens-lyon.fr`

2 **ENS de Lyon, Université de Lyon, LIP‡, Lyon, France**
`colin.riba@ens-lyon.fr`

Abstract

Church's synthesis problem asks whether there exists a finite-state stream transducer satisfying a given input-output specification. For specifications written in Monadic Second-Order Logic over infinite words, Church's synthesis can theoretically be solved algorithmically using automata and games. We revisit Church's synthesis *via* the Curry-Howard correspondence by introducing SMSO, a non-classical subsystem of MSO, which is shown to be sound and complete w.r.t. synthesis thanks to an automata-based realizability model.

1998 ACM Subject Classification F.4.1 Mathematical Logic.

Keywords and phrases Intuitionistic Arithmetic, Realizability, Monadic Second-Order Logic on Infinite Words

Digital Object Identifier 10.4230/LIPIcs.FSCD.2017.30

1 Introduction

Church's synthesis [5] consists in the automatic extraction of stream transducers (or *Mealy machines*) from input-output specifications, typically written in some subsystem of *Monadic Second-Order Logic* (MSO) over ω-words. MSO over ω-words is a decidable logic by Büchi's Theorem [3]. It subsumes non-trivial logics used in verification such as LTL (see e.g. [16, 10]).

Traditional approaches to synthesis (see e.g. [17]) are based, *via* McNaughton's Theorem [9], on the translation of MSO-formulae to *deterministic* automata on ω-words (such as *Muller* or *parity* automata)[1]. Such automata are then turned into game graphs, in which the *Opponent* O (∀bélard) plays input characters to which the *Proponent* P (∃loïse) replies with output characters. Solutions to Church's synthesis are then given by the Büchi-Landweber Theorem [4], which says that in such games, either P or O has finite-state winning strategy.

Fully automatic approaches to synthesis suffer from prohibitively high computational costs, essentially for the following two reasons. First, the translation of MSO-formulae to automata is non-elementary, and McNaughton Theorem involves a non-trivial powerset construction (such as *Safra construction*, see e.g. [16, 10]). Second, similarly as with other automatic verification techniques based on Model Checking, the solution of parity games ultimately relies on exhaustive state exploration. While they have had (and still have)

* This work was partially supported by the ANR-14-CE25-0007 – RAPIDO and the ANR-BLANC-SIMI-2-2011 – RECRÉ.

† UMR 5668 CNRS ENS Lyon UCBL INRIA.

‡ UMR 5668 CNRS ENS Lyon UCBL INRIA.

[1] A solution is also possible *via* tree automata [11].

© Pierre Pradic and Colin Riba;
licensed under Creative Commons License CC-BY

considerable success for verifying concurrency properties, such techniques hardly managed up to now to give practical algorithms for synthesis (even for fragments of LTL, see e.g. [1]).

In this work, we propose a Curry-Howard approach to Church's synthesis based on a proof system allowing for human intervention and compositional reasoning. In a typical usage scenario, the user interactively performs some proofs steps and delegate the generated subgoals to automatized synthesis procedures. The partial proof tree built by the user is then translated to a combinator able to compose the transducers synthesized by the automatic procedures[2]. Having in mind that interactive proof systems (such as Coq [15]) have known in the last decade an explosion of large developments, we believe that semi-automatic approaches like ours could ultimately help mitigate the algorithmic costs of synthesis, in particular in helping to combine automatic methods with human intervention.

The Curry-Howard correspondence asserts that, given a suitable proof system, any proof therein can be interpreted as a program. Actually, *via* the Curry-Howard correspondence, the soundness of many type/proof systems is proved by means of *realizability*, which tells how to read a formula from the logic as a specification for a program. Our starting point is the fact that MSO on ω-words can be completely axiomatized as a subsystem of second-order Peano arithmetic [14] (see also [12]). From the classical axiomatization of MSO, we derive an intuitionistic system SMSO equipped with an extraction procedure which is sound and complete w.r.t. Church's synthesis: proofs of existential statements can be translated to Mealy machines and such proofs exist for all solvable instances of Church's synthesis. The key point in our approach is that on the one hand, finite-state realizers[3] are constructively extracted from proofs in SMSO, while on the other hand, their correctness involves the full power of MSO. So in particular, our adaptation of the usual Adequacy Lemma of realizability does rely on the non-constructive proof of correctness of deterministic automata obtained by McNaughton's Theorem (see e.g. [16]), while these automata do not have to be concretely built during the extraction procedure.

The paper is organized as follows. We first recall in §2 some background on MSO and Church's synthesis. Our intuitionistic system SMSO is then presented in §3. Section 4 provides some technical material as well as detailed examples on the representation of Mealy machines in MSO, and §5 presents our realizability model.

2 Church's Synthesis and MSO on Infinite Words

Notations. Alphabets (denoted Σ, Γ, etc) are finite non-empty sets. Concatenation of words s, t is denoted either $s.t$ or $s \cdot t$, and ε is the empty word. We use the vectorial notation both for words and finite sequences, so that e.g. $\overline{B}$ denotes a finite sequence $B_1, \ldots, B_n$ and $\overline{\mathsf{a}}$ denotes a word $\mathsf{a}_1.\cdots.\mathsf{a}_n \in \Sigma^*$. Given an ω-word (or stream) $B \in \Sigma^\omega$ and $n \in \mathbb{N}$ we write $B{\restriction}n$ for the finite word $B(0).\cdots.B(n-1) \in \Sigma^*$. For each $k \in \mathbb{N}$, we still write k for the function from $\mathbb{N}$ to $\mathbf{2}$ which takes n to 1 iff $n = k$.

---

[2] We thank the anonymous referee who urged us to state this explicitly.

[3] We use the word *realizer* with two historically distinct meanings. In the context of Church's synthesis, a realizer of a $\forall\exists$-formula is a transducer which witnesses the $\forall\exists$ by computing an instantiation of the existential variables while reading input values for the universal variables (see e.g. [1]). In (constructive) proof theory, *realizability* is a relation between programs (the realizers) and formulae, usually defined by induction on formulae (see e.g. [8]). A realizer of a $\forall\exists$-formula consists of a function witnessing the $\forall\exists$, together with a realizer witnessing the correctness of that function.

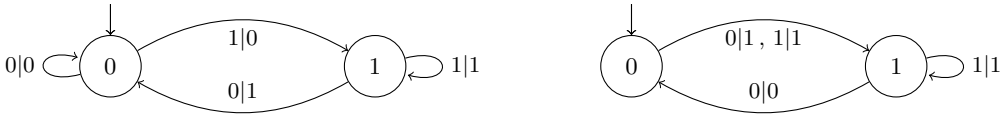

Figure 1 Examples of Mealy Machines (where a transition $\mathtt{a}|\mathtt{b}$ outputs $\mathtt{b}$ from input $\mathtt{a}$).

Church's Synthesis and Synchronous Functions. Church's synthesis consists in the automatic extraction of stream transducers (or *Mealy machines*) from input-output specifications (see e.g. [17]). As a typical specification, consider, for a machine which outputs streams $B \in \mathbf{2}^\omega$ from input streams $A \in \mathbf{2}^\omega$, the behavior (from [17]) expressed by

$$\Phi(A,B) \quad \overset{\text{def.}}{\iff} \quad \begin{cases} \forall n (A(n) = 1 \implies B(n) = 1) & \text{and} \\ \forall n (B(n) = 0 \implies B(n+1) = 1) & \text{and} \\ (\exists^\infty n\ A(n) = 0) \implies (\exists^\infty n\ B(n) = 0) \end{cases} \tag{1}$$

In words, the relation $\Phi(A,B)$ imposes $B(n) \in \mathbf{2}$ to be 1 whenever $A(n) \in \mathbf{2}$ is 1, B not to be 0 in two consecutive positions, and moreover B to be infinitely often 0 whenever A is infinitely often 0. We are interested in the realization of such specifications by finite-state stream transducers or *Mealy machines*.

▶ **Definition 2.1** (Mealy Machine)**.** A *Mealy machine* $\mathcal{M}$ with input alphabet Σ and output alphabet Γ (notation $\mathcal{M} : \Sigma \to \Gamma$) is given by a finite set of states Q with a distinguished initial state $q^\iota \in Q$, and a transition function $\partial : Q \times \Sigma \to Q \times \Gamma$.

We often write ∂^o for $\pi_2 \circ \partial : Q \times \Sigma \to \Gamma$ and ∂^* for the map $\Sigma^* \to Q$ obtained by iterating ∂ from the initial state: $\partial^*(\varepsilon) := q^\iota$ and $\partial^*(\overline{\mathtt{a}}.\mathtt{a}) := \pi_1(\partial(\partial^*(\overline{\mathtt{a}}), \mathtt{a}))$

A Mealy machine $\mathcal{M} : \Sigma \to \Gamma$ induces a function $F : \Sigma^\omega \to \Gamma^\omega$ obtained by iterating ∂^o along the input: $F(B)(n) = \partial^o(\partial^*(B\restriction n), B(n))$. Hence F can produce a length-n prefix of its output from a length-n prefix of its input. These functions are called *synchronous*.

▶ **Definition 2.2** (Synchronous Function)**.** A function $F : \Sigma^\omega \to \Gamma^\omega$ is *synchronous* if for all $n \in \mathbb{N}$ and all $A, B \in \Sigma^\omega$ we have $F(A)\restriction n = F(B)\restriction n$ whenever $A\restriction n = B\restriction n$. We say that a synchronous function F is *finite-state* if it is induced by a Mealy machine.

▶ **Example 2.3.**

(a) The identity function $\Sigma^\omega \to \Sigma^\omega$ is induced by the Mealy machine with state set $\mathbf{1} = \{\bullet\}$ and identity transition function $\partial : (\bullet, \mathtt{a}) \longmapsto (\bullet, \mathtt{a})$.

(b) The Mealy machine depicted on Fig. 1 (left) induces a synchronous function $F : \mathbf{2}^\omega \to \mathbf{2}^\omega$ such that $F(B)(n+1) = 1$ iff $B(n) = 1$.

(c) The Mealy machine depicted on Fig. 1 (right), taken from [17], induces a synchronous function which realizes the specification (1).

(d) Synchronous functions are obviously continuous (taking the product topology on Σ^ω and Γ^ω, with Σ, Γ discrete), but there are continuous functions which are not synchronous, for instance the function $P : \mathbf{2}^\omega \to \mathbf{2}^\omega$ such that $P(A)(n) = 1$ iff $A(n+1) = 1$.

For the definition and adequacy of our realizability interpretation, it turns out to be convenient to work with a category of finite-state synchronous functions.

▶ **Definition 2.4.** Let $\mathbf{M}$ be the category whose objects are alphabets and whose maps from Σ to Γ are finite-state synchronous functions $F : \Sigma^\omega \to \Gamma^\omega$.

$$\begin{array}{rrcl} \textbf{Atoms:} & \alpha & ::= & x \doteq y \mid x \dot{\leq} y \mid \mathsf{S}(x,y) \mid \mathsf{Z}(x) \mid x \mathrel{\dot{\in}} X \mid \top \mid \bot \\ \textbf{Deterministic formulae:} & \delta, \delta' & ::= & \alpha \mid \delta \wedge \delta' \mid \neg\varphi \\ \textbf{MSO formulae:} & \varphi, \psi & ::= & \delta \mid \varphi \wedge \psi \mid \exists x\, \varphi \mid \exists X\, \varphi \end{array}$$

Figure 2 The Formulae of MSO.

Note that functions $\mathtt{f} : \Sigma \to \Gamma$ induce $\mathbf{M}$-maps $[\mathtt{f}] : \Sigma \to_{\mathbf{M}} \Gamma$. Also, $\mathbf{M}$ has finite products.

▶ **Proposition 2.5.** *The category* $\mathbf{M}$ *has finite products. The product of* $\Sigma_1, \ldots, \Sigma_n$ *(for* $n \geq 0$*) is given by the* $\mathbf{Set}$*-product* $\Sigma_1 \times \cdots \times \Sigma_n$ *(so that* $\mathbf{1}$ *is terminal in* $\mathbf{M}$*).*

Monadic Second-Order Logic (MSO) on Infinite Words. We consider a formulation of MSO based on a purely relational two-sorted language, with a specific choice of atomic formulae. There is a sort of *individuals*, with variables x, y, z, etc, and a sort of *(monadic) predicates*, with variables X, Y, Z, etc. Our formulae for MSO, denoted φ, ψ, etc are given on Fig. 2. They are defined by mutual induction with the *deterministic formulae* (denoted δ, δ', etc) from atomic formulae ranged over by α.

MSO formulae are interpreted in the standard model $\mathfrak{N}$ of ω-words as usual. Individual variables range over natural numbers $n, m, \ldots \in \mathbb{N}$ and predicate variables range over sets of natural numbers $A, B, \ldots \in \mathcal{P}(\mathbb{N}) \simeq \mathbf{2}^{\omega}$. The atomic predicates are interpreted as expected: $\doteq$ is equality, $\dot{\in}$ is membership, $\dot{\leq}$ is the relation $\leq$ on $\mathbb{N}$, S is the successor relation, and Z holds on n iff $n = 0$. We often write $X(x)$ or even Xx for $x \mathrel{\dot{\in}} X$. As usual we let:

$$\varphi \to \psi \;:=\; \neg(\varphi \wedge \neg\psi) \qquad \varphi \vee \psi \;:=\; \neg(\neg\varphi \wedge \neg\psi) \qquad \forall(-)\, \varphi \;:=\; \neg\exists(-)\, \neg\varphi$$

MSO on ω-words is known to be decidable by Büchi's Theorem [3].

▶ **Theorem 2.6** (Büchi [3])**.** MSO *over* $\mathfrak{N}$ *is decidable.*

Following [3] (but see also e.g. [10]), the (non-deterministic) automata method for deciding MSO proceeds by a recursive translation of MSO-formulae to *Büchi automata*. A *Büchi automaton* is a non-deterministic finite state automaton running on ω-words. Büchi automata are equipped with a set of final states, and a run on an ω-word is accepting if it has infinitely many occurrences of final states.

The crux of Büchi's Theorem is the effective closure of Büchi automata under complement. Let us recall a few known facts (see e.g. [16, 7]). First, the translation of MSO-formulae to automata is non-elementary. Second, its is known that *deterministic* Büchi automata are strictly less expressive than non-deterministic ones. Finally, it is known that complementation of Büchi automata is algorithmically hard: there is a family of languages $(\mathcal{L}_n)_{n>0}$ such that each $\mathcal{L}_n$ can be recognized by a Büchi automaton with $n+2$ states, but such that the complement of $\mathcal{L}_n$ can not be recognized by a Büchi automaton with less than $n!$ states.

Church's Synthesis for MSO. Church's synthesis problem for MSO is the following. Given as input an MSO formula $\varphi(\overline{X}; \overline{Y})$ (where $\overline{X} = X_1, \ldots, X_q$ and $\overline{Y} = Y_1, \ldots, Y_p$), (1) decide whether there exist finite-state synchronous functions $\overline{F} = F_1, \ldots, F_p : \mathbf{2}^q \to_{\mathbf{M}} \mathbf{2}$ such that $\mathfrak{N} \models \varphi(\overline{A}; \overline{F}(\overline{A}))$ for all $\overline{A} \in (\mathbf{2}^{\omega})^q \simeq (\mathbf{2}^q)^{\omega}$, and (2), construct such $\overline{F}$ whenever they exist.

▶ **Example 2.7.** The specification Φ displayed in (1) can be officially written in the language of MSO as the following formula $\phi(X;Y)$ (where $\exists^\infty t\ \varphi(t)$ stands for $\forall x \exists t(t \dot{\geq} x \wedge \varphi(t))$):

$$\phi(X;Y) \ := \ \forall t\,(Xt \to Yt) \wedge \forall t,t'\,(\mathsf{S}(t,t') \to \neg Yt \to Yt') \wedge [(\exists^\infty t\,\neg Xt) \to (\exists^\infty t\,\neg Yt)]$$

Church's synthesis has been solved by Büchi & Landweber [4], using automata on ω-words and infinite two-player games (a solution is also possible *via* tree automata [11]): there is an algorithm which, on input $\varphi(\overline{X};\overline{Y})$, (1) decides when a synchronous realizer of $\varphi(\overline{X};\overline{Y})$ exists, (2) provides a finite-state Mealy machine implementing it[4], and (3) moreover provides a synchronous finite-state counter realizer (*i.e.* a realizer of $\psi(\overline{Y};\overline{X}) := \neg\varphi(\overline{X};\overline{Y})$) when no synchronous realizer of $\varphi(\overline{X};\overline{Y})$ exists.

The standard algorithm solving Church's synthesis for MSO (see e.g. [17]) proceeds *via* McNaughton's Theorem ([9], see also e.g. [10, 16]), which states that Büchi automata can be translated to equivalent *deterministic* finite state automata, but equipped with stronger acceptance conditions than Büchi automata. There are different variants of such conditions (*Muller*, *Rabin*, *Streett* or *parity* conditions, see e.g. [16, 7]). All of them allow to specify which states an infinite run *must not* see infinitely often. For the purpose of this paper, we only need to consider the simplest of them, the Muller conditions. A *Muller condition* is given by a family of set of states $\mathcal{T}$, and a run is accepting when the set of states occurring infinitely often in it belongs to the family $\mathcal{T}$.

▶ **Theorem 2.8** (McNaughton [9])**.** *Each Büchi automaton is equivalent to a deterministic Muller automaton.*

There is a lower bound in $2^{O(n)}$ for the number of states of a Muller automaton equivalent to a Büchi automaton with n states. The best known constructions for McNaughton's Theorem (such as *Safra's construction* or its variants) give deterministic Muller automata with $2^{O(n\log(n))}$ states from non-deterministic Büchi automata with n states.

The standard solution to Church's synthesis for MSO starts by translating $\varphi(\overline{X};\overline{Y})$ to a deterministic Muller automaton, and then turns this deterministic automaton into a two-player sequential game, in which the Opponent ∀bélard plays inputs bit sequences in $\mathbf{2}^p$ while the Proponent ∃loïse replies with outputs bit sequences in $\mathbf{2}^q$. The game is equipped with an ω-regular winning condition (induced by the acceptance condition of the Muller automaton). The solution is then provided by Büchi-Landweber's Theorem, which states that ω-regular games on finite graphs are effectively determined, and moreover that the winner always has a finite state winning strategy.

▶ **Example 2.9.** Consider the last conjunct $\phi_2[X,Y] := (\exists^\infty t\ \neg Xt) \to (\exists^\infty t\ \neg Yt)$ of the formula $\phi(X;Y)$ of Ex. 2.7. When translating ϕ_2 to a finite state automaton, the positive occurrence of $(\exists^\infty t\ \neg Yt)$ can be translated to a deterministic Büchi automaton. However, the negative occurrence of $(\exists^\infty t\ \neg Xt)$ corresponds to $(\forall^\infty t\ Xt) = (\exists n\,\forall t \dot{\geq} n\,Xt)$ and can not be translated to a *deterministic* Büchi automaton. Even if a very simple two-state Muller automaton exists for $(\forall^\infty t\ Xt)$, McNaughton's Theorem 2.8 is in general required for positive occurrences of the form $\forall^\infty t\ (-)$.

An Axiomatization of MSO. Our approach to Church's synthesis relies on the fact that the MSO-theory of $\mathfrak{N}$ can be completely axiomatized as a subsystem of second-order Peano

[4] It follows from the finite-state determinacy of ω-regular games that a finite-state synchronous realizer exists whenever a synchronous realizer exists (see e.g. [17]).

$$\frac{}{\overline{\varphi} \vdash t \doteq t} \qquad \frac{\overline{\varphi} \vdash \varphi[t/z] \qquad \overline{\varphi} \vdash t \doteq u}{\overline{\varphi} \vdash \varphi[u/z]} \qquad \frac{\overline{\varphi} \vdash x \dot{\leq} y \qquad \overline{\varphi} \vdash y \dot{\leq} x}{\overline{\varphi} \vdash x \doteq y}$$

$$\frac{}{\overline{\varphi} \vdash x \dot{\leq} x} \qquad \frac{\overline{\varphi} \vdash \mathsf{S}(x,y)}{\overline{\varphi} \vdash x \dot{\leq} y} \qquad \frac{\overline{\varphi} \vdash x \dot{\leq} y \qquad \overline{\varphi} \vdash y \dot{\leq} z}{\overline{\varphi} \vdash x \dot{\leq} z} \qquad \frac{}{\overline{\varphi}, \mathsf{S}(x,y), \mathsf{Z}(y) \vdash \bot}$$

$$\frac{}{\overline{\varphi} \vdash \exists y\, \mathsf{Z}(y)} \qquad \frac{}{\overline{\varphi} \vdash \exists y\, \mathsf{S}(x,y)} \qquad \frac{}{\overline{\varphi}, \mathsf{S}(y,y'), x \dot{\leq} y', \neg(x \doteq y') \vdash x \dot{\leq} y}$$

$$\frac{}{\overline{\varphi}, \mathsf{S}(y,x), \mathsf{S}(z,x) \vdash y \doteq z} \qquad \frac{}{\overline{\varphi}, \mathsf{Z}(x), \mathsf{Z}(y) \vdash x \doteq y} \qquad \frac{}{\overline{\varphi}, \mathsf{S}(x,y), \mathsf{S}(x,z) \vdash y \doteq z}$$

Figure 3 Arithmetic Rules of MSO and SMSO.

arithmetic [14] (see also [12]). We consider a specific set of axioms which consists of the rules depicted on Fig. 3 together with the following *comprehension* and *induction* rules

$$\frac{\overline{\varphi} \vdash \varphi[\psi[y]/X]}{\overline{\varphi} \vdash \exists X\, \varphi} \qquad \frac{\overline{\varphi}, \mathsf{Z}(z) \vdash \varphi[z/x] \qquad \overline{\varphi}, \mathsf{S}(y,z), \varphi[y/x] \vdash \varphi[z/x]}{\overline{\varphi} \vdash \varphi} \tag{2}$$

where z and y do not occur free in $\overline{\varphi}, \varphi$, and where $\varphi[\psi[y]/X]$ is the usual formula substitution, which commutes over all connectives (avoiding the capture of free variables), and with $(x \mathrel{\dot{\in}} X)[\psi[y]/X] \;=\; \psi[x/y]$.

▶ **Theorem 2.10** ([14]). *For every (closed)* MSO*-formula φ, we have $\mathfrak{N} \models \varphi$ if and only if $\vdash \varphi$ is derivable in classical two-sorted predicate logic with the rules of Fig. 3 and (2).*

3 A Synchronous Intuitionistic Restriction of MSO

We now introduce SMSO, an intuitionistic restriction of MSO. As expected, SMSO contains MSO *via* negative translation. But thanks to its vocabulary without primitive universals, SMSO actually admits a Glivenko Theorem, so that SMSO proves $\neg\neg\varphi$ whenever MSO $\vdash \varphi$. Moreover, SMSO is equipped with an extraction procedure which is sound and complete w.r.t. Church's synthesis: proofs of existential statements can be translated to finite state synchronous realizers, and such proofs exist for all solvable instances of Church's synthesis.

As it is common with intuitionistic versions of classical systems, SMSO has the same language as MSO, and its deduction rules are based on intuitionistic predicate calculus. Moreover, since (monadic) predicate variables are computational objects in our realizability interpretation, similarly as with higher-type Heyting arithmetic (see e.g. [8]), SMSO has a comprehension scheme which corresponds to the negative translation of the full comprehension scheme of MSO[5]. On the other hand, for the extraction of *synchronous* realizers from proofs, SMSO has a restricted induction scheme corresponding to the negative translation of the induction scheme of MSO. As a consequence, and in contrast with usual versions of intuitionistic (Heyting) arithmetic, this restricted induction scheme is not able to prove the elimination of double negation on atomic formulae. Fortunately, all atomic formulae of MSO can be interpreted by *deterministic Büchi* automata, and have a trivial computational

[5] In contrast with Girard's System F [6], in which second-order variables have no computational content.

$$\frac{}{\overline{\varphi}, \varphi \vdash \varphi} \qquad \frac{\overline{\varphi} \vdash \psi \qquad \overline{\varphi}, \psi \vdash \varphi}{\overline{\varphi} \vdash \varphi} \qquad \frac{}{\overline{\varphi}, \neg\neg\delta \vdash \delta} \qquad \frac{\overline{\varphi} \vdash \varphi \qquad \overline{\varphi} \vdash \neg\varphi}{\overline{\varphi} \vdash \bot} \qquad \frac{\overline{\varphi} \vdash \bot}{\overline{\varphi} \vdash \varphi}$$

$$\frac{\overline{\varphi} \vdash \varphi \qquad \overline{\varphi} \vdash \psi}{\overline{\varphi} \vdash \varphi \wedge \psi} \qquad \frac{\overline{\varphi} \vdash \varphi \wedge \psi}{\overline{\varphi} \vdash \varphi} \qquad \frac{\overline{\varphi} \vdash \varphi \wedge \psi}{\overline{\varphi} \vdash \psi} \qquad \frac{\overline{\varphi} \vdash \varphi[y/x]}{\overline{\varphi} \vdash \exists x\, \varphi} \qquad \frac{\overline{\varphi} \vdash \varphi[Y/X]}{\overline{\varphi} \vdash \exists X\, \varphi}$$

$$\frac{\overline{\varphi}, \varphi \vdash \psi \qquad \overline{\varphi} \vdash \exists x\, \varphi}{\overline{\varphi} \vdash \psi}\ (x \text{ not free in } \overline{\varphi}, \psi) \qquad \frac{\overline{\varphi}, \varphi \vdash \psi \qquad \overline{\varphi} \vdash \exists X\, \varphi}{\overline{\varphi} \vdash \psi}\ (X \text{ not free in } \overline{\varphi}, \psi)$$

Figure 4 Logical Rules of SMSO (where δ is deterministic).

content. This more generally leads to the notion of *deterministic* formulae (see Fig. 2), which contain negative formulae and atomic formulae. Deterministic formulae will be interpreted by deterministic (not nec. Büchi) automata, and have trivial realizers. We can therefore have as axiom the elimination of double negation for deterministic formulae, which are thus the SMSO counterpart of the formulae of Heyting arithmetic admitting elimination of double negation (see e.g. [8]).

Furthermore, SMSO is equipped with a positive *synchronous* restriction of comprehension, which allows to have realizers for all solvable instances of Church's synthesis. The synchronous restriction of comprehension asks the comprehension formula to be *uniformly bounded* in the following sense.

▶ **Definition 3.1.**

(i) Given MSO-formulae φ and θ and a variable y, the *relativization of φ to $\theta[y]$* (notation $\varphi\restriction\theta[y]$), is defined by induction on φ as usual:

$$\alpha\restriction\theta[y] := \alpha \qquad (\varphi \wedge \psi)\restriction\theta[y] := \varphi\restriction\theta[y] \wedge \psi\restriction\theta[y] \qquad (\neg\varphi)\restriction\theta[y] := \neg\varphi\restriction\theta[y]$$

$$(\exists X\, \varphi)\restriction\theta[y] := \exists X\, \varphi\restriction\theta[y] \qquad (\exists x\, \varphi)\restriction\theta[y] := \exists x\, (\theta[x/y] \wedge \varphi\restriction\theta[y])$$

where, in the clauses for $\exists$, the variables x and X are assumed not to occur free in θ. Note that y does not occur free in $\varphi\restriction\theta[y]$.

(ii) An MSO-formula $\hat{\varphi}$ is *bounded by x* if it is of the form $\psi\restriction(y \dot{\leq} x)[y]$ (notation $\psi\restriction[- \dot{\leq} x]$). It is *uniformly bounded* if moreover x is the only free individual variable of $\hat{\varphi}$.

As we shall see in §4.3, bounded formulae are exactly those definable in MSO over finite words. We are now ready to define the system SMSO.

▶ **Definition 3.2** (SMSO)**.** The logic SMSO has the same language as MSO. Its deduction rules are those given in Fig. 4 together with the rules of Fig. 3 and with the following rules of resp. *negative comprehension*, *deterministic induction* (where x and y do not occur free in $\overline{\varphi}, \delta$) and *synchronous comprehension* in which $\hat{\varphi}$ is uniformly bounded by y:

$$\frac{\overline{\varphi} \vdash \psi[\varphi[y]/X]}{\overline{\varphi} \vdash \neg\neg\exists X\, \psi} \qquad \frac{\overline{\varphi}, \mathsf{Z}(z) \vdash \delta[z/x] \qquad \overline{\varphi}, \mathsf{S}(y, z), \delta[y/x] \vdash \delta[z/x]}{\overline{\varphi} \vdash \delta} \qquad \frac{\overline{\varphi} \vdash \psi[\hat{\varphi}[y]/X]}{\overline{\varphi} \vdash \exists X\, \psi}$$

▶ Remark. The axiom $\overline{\varphi}, \neg\neg\delta \vdash \delta$ of double negation elimination for deterministic formulae would already be derivable in a version of SMSO where this axiom is weakened to double negation elimination for atomic formulae. We take $\overline{\varphi}, \neg\neg\delta \vdash \delta$ as an axiom because it admits trivial realizers. Similarly, the cut rule is admissible, but we include it since we have a direct composition of realizers.

A Glivenko Theorem for SMSO. Thanks to its limited vocabulary, SMSO satisfies a Glivenko theorem, and thus a very simple negative translation from MSO. Glivenko's theorem is usually stated only for propositional logic, but can be extended to formulae containing existentials; the impossible case is the universal quantification. In particular, should one extend the logical constructs with universal quantification by freely adjoining them to SMSO, this would no longer hold. This would actually not be such a severe consequence since our results would also hold with a usual recursive negative translation instead of $\neg\neg(-)$.

▶ **Theorem 3.3.** *If* $\mathsf{MSO} \vdash \varphi$*, then* $\mathsf{SMSO} \vdash \neg\neg\varphi$*.*

The Main Result. We are now ready to state the main result of this paper, which says that SMSO is correct and complete (w.r.t. its provable existentials) for Church's synthesis.

▶ **Theorem 3.4** (Main Theorem)**.** *Consider an* MSO*-formula* $\varphi(\overline{X};\overline{Y})$*.*

(i) *From a proof of* $\exists\overline{Y}\,\neg\neg\varphi(\overline{X};\overline{Y})$ *in* SMSO*, one can extract a finite-state synchronous realizer of* $\varphi(\overline{X};\overline{Y})$*.*

(ii) *If* $\varphi(\overline{X};\overline{Y})$ *admits a (finite-state) synchronous realizer, then* $\mathsf{SMSO} \vdash \exists\overline{Y}\,\neg\neg\varphi(\overline{X};\overline{Y})$*.*

The correctness part (i) of Thm. 3.4 will be proved in §5 using a notion of realizability for SMSO based on automata and synchronous finite-state functions. The completeness part (ii) will be proved in §4.1, relying the completeness of the axiomatization of MSO (Thm. 2.10) together with the correctness of the negative translation $\neg\neg(-)$ (Thm. 3.3).

4 On the Representation of Mealy Machines in MSO

This section gathers several (possibly known) results related to the representation of Mealy machines in MSO. We begin in §4.1 with the completeness part of Thm. 3.4, which follows usual representations of automata in MSO (see e.g. [16, §5.3]). We then recall from [14, 12] the *Recursion Theorem*, which is a convenient tool to reason on runs of deterministic automata in MSO (§4.2). In §4.3 we state a Lemma for the correctness part of Thm. 3.4, which relies on the usual translation of MSO-formulae over *finite words* to DFA's (see e.g. [16, §3.1]). Finally, in §4.4 we give a possible strengthening of the synchronous comprehension rule of SMSO (but which is based on Büchi's Theorem 2.6).

We work with the following notion of representation.

▶ **Definition 4.1.** Let φ be a formula with free variables among $z, x_1, \dots, x_p, X_1, \dots, X_q$. We say that φ *z-represents* $F : \mathbf{2}^p \times \mathbf{2}^q \longrightarrow_{\mathbf{M}} \mathbf{2}$ if for all $n \in \mathbb{N}$, all $\overline{A} \in (\mathbf{2}^\omega)^q$, and all $\overline{k} \in (\mathbf{2}^\omega)^p$ such that $k_i \leq n$ for all $i \leq p$, we have

$$F(\overline{k},\overline{A})(n) = 1 \qquad \text{iff} \qquad \mathfrak{N} \models \varphi[n/z, \overline{k}/\overline{x}, \overline{A}/\overline{X}] \tag{3}$$

4.1 Internalizing Mealy Machines in MSO

The completeness part (ii) of Thm. 3.4 relies on the following simple fact.

▶ **Proposition 4.2.** *For every finite-state synchronous* $F : \mathbf{2}^p \longrightarrow_{\mathbf{M}} \mathbf{2}$*, one can build a deterministic uniformly bounded formula* $\delta[\overline{X}, x]$ *which* x*-represents* F*.*

Proof. The proof is a simple adaptation of the usual pattern (see e.g. [16, §5.3]). Let $F : \mathbf{2}^p \to_{\mathbf{M}} \mathbf{2}$ be induced by a Mealy machine $\mathcal{M}$. W.l.o.g. we can assume the state set of $\mathcal{M}$ to be of the form $\mathbf{2}^q$. Then F is represented by a formula of the form

$$\delta[\overline{X},x] \;:=\; \forall\overline{Q},Y\left(\left[\begin{array}{l}\forall t \leq \dot{x}\,(\mathsf{Z}(t) \to \mathsf{I}[\overline{Q}(t)]) \quad \wedge \\ \forall t,t' \leq \dot{x}\,(\mathsf{S}(t,t') \to \mathsf{H}[\overline{Q}(t),\overline{X}(t),Y(t),\overline{Q}(t')])\end{array}\right] \longrightarrow\; Y(x)\right) \tag{4}$$

where $\overline{X} = X_1, \dots, X_p$ codes sequences of inputs, Y codes sequences of outputs, and where $\overline{Q} = Q_1, \dots, Q_q$ codes runs. ◀

▶ Remark. In the proof of Prop. 4.2, since $\mathcal{M}$ is deterministic, we can assume the formula $\mathsf{I}[\overline{Q}(t)]$ to be of the form $\bigwedge_{1\leq i\leq q}[Q_i(t) \leftrightarrow \mathsf{B}_i]$ with $\mathsf{B}_i \in \{\top, \bot\}$, and, for some propositional formulae $\mathsf{O}[-,-], \overline{\mathsf{D}}[-,-]$, the formula $\mathsf{H}[\overline{Q}(t), \overline{X}(t), \overline{Y}(t), \overline{Q}(t')]$ to be of the form

$$\big(Y(t) \longleftrightarrow \mathsf{O}[\overline{Q}(t), \overline{X}(t)]\big) \quad \wedge \quad \bigwedge_{1\leq i\leq q} \big(Q_i(t') \longleftrightarrow \mathsf{D}_i[\overline{Q}(t), \overline{X}(t)]\big) \tag{5}$$

▶ **Example 4.3.** The function induced by the Mealy machine of Ex. 2.3.(c) (depicted on Fig. 1, right), is represented by a formula of the form (4), where $\overline{Q} = Q$ (since the machine has state set $\mathbf{2}$), $\overline{X} = X$, where $\mathsf{I}[-] := [(-) \leftrightarrow \bot]$ (since state 0 is initial) and

$$\mathsf{O}[Q(t), X(t)] \quad = \quad \mathsf{D}[Q(t), X(t)] \quad = \quad (\neg Q(t) \ \vee \ [Q(t) \wedge X(t)]) \tag{6}$$

The completeness of our approach to Church's synthesis is obtained as follows.

Proof of Thm. 3.4.(ii). Assume that $\varphi(\overline{X}; \overline{Y})$ admits a realizer $C : \mathbf{2}^q \longrightarrow_{\mathbf{M}} \mathbf{2}^p$. Using the Cartesian structure of $\mathbf{M}$ (Prop. 2.5), we assume $C = \overline{C} = C_1, \dots, C_p$ with $C_i : \mathbf{2}^q \to_{\mathbf{M}} \mathbf{2}$. We thus have $\mathfrak{N} \models \varphi[\overline{B}/\overline{X}, \overline{C}(\overline{B})/\overline{Y}]$ for all $\overline{B} \in (\mathbf{2}^\omega)^q \simeq (\mathbf{2}^q)^\omega$. Now, by Prop. 4.2 there are uniformly bounded (deterministic) formulae $\overline{\delta} = \delta_1, \dots, \delta_p$, with free variables among $\overline{X}, x$, and such that (3) holds for all $i = 1, \dots, p$. It thus follows that $\mathfrak{N} \models \forall \overline{X} \varphi[\overline{\delta[x]}/\overline{Y}]$. Then, by completeness (Thm. 2.10) we know that $\vdash \varphi[\overline{\delta}[x]/\overline{Y}]$ is provable in MSO, and by negative translation (Thm. 3.3) we get $\mathsf{SMSO} \vdash \neg\neg\varphi[\overline{\delta}[x]/\overline{Y}]$. We can then apply ($p$ times) the synchronous comprehension scheme of SMSO and obtain $\mathsf{SMSO} \vdash \exists \overline{Y} \neg\neg\varphi(\overline{X}; \overline{Y})$. ◀

▶ **Example 4.4.** Recall the specification (1) from [17], represented in MSO by the formula $\phi(X; Y)$ of Ex. 2.7. Write $\phi(X; Y) = \phi_0[X, Y] \wedge \phi_1[X, Y] \wedge \phi_2[X, Y]$ where

$$\begin{array}{lcl} \phi_0[X, Y] & := & \forall t\, (Xt \ \to \ Yt) \\ \phi_1[X, Y] & := & \forall t, t'\, (\mathsf{S}(t, t') \wedge \neg Yt \ \to \ Yt') \\ \phi_2[X, Y] & := & (\exists^\infty t\ \neg Xt) \ \to \ (\exists^\infty t\ \neg Yt) \end{array}$$

Note that ϕ_0 and ϕ_1 are monotonic in Y, while ϕ_2 is anti-monotonic in Y. The formula ϕ_0 is trivially realized by the identity function $\mathbf{2} \to_{\mathbf{M}} \mathbf{2}$ (see Ex. 2.3.(a)), which is itself represented by the deterministic uniformly bounded formula $\delta_0[X, x] := (x \mathrel{\dot\in} X)$. For ϕ_1 (which asks Y not to have two consecutive occurrences of 0), consider

$$\delta_1[X, x] \quad := \quad \delta_0[X, x] \ \vee \ \exists t \mathrel{\dot\leq} x[\mathsf{S}(t, x) \wedge \neg X(t)]$$

We have $\mathsf{MSO} \vdash \phi_0[X, \delta_1[x]/Y]$ since $\delta_0 \vdash_{\mathsf{MSO}} \delta_1$ and moreover $\mathsf{MSO} \vdash \phi_1[X, \delta_1[x]/Y]$ since

$$\mathsf{S}(t, t')\,,\ \neg Xt\,,\ \neg\exists u(\mathsf{S}(u, t) \wedge \neg Xu) \ \vdash_{\mathsf{MSO}} \ Xt' \ \vee \ \exists t''(\mathsf{S}(t'', t') \wedge \neg Xt'')$$ ◀

The case of ϕ_2 in Ex. 4.4 is more complex. The point is that $\phi_2[\delta_1[x]/Y]$ does not hold because if $\forall^\infty t\ \neg Xt$ (that is if X remains constantly 0 from some time on), then δ_1 will output no 1's. On the other hand, the machine of Ex. 2.3.(c) involves an internal state, and can be represented using a fixpoint formula of the form (4). Reasoning on such formulae is easier with more advanced tools on MSO, that we provide in §4.2.

4.2 The Recursion Theorem

Theorem 3.4.(ii) ensures that SMSO is able to handle all solvable instances of Church's synthesis, but it gives no hint on how to actually produce proofs. When reasoning on fixpoint formulae as those representing Mealy machines in Prop. 4.2, a crucial role is played by the *Recursion Theorem* for MSO [14] (see also [12]). The Recursion Theorem says that MSO allows to define predicates by well-founded induction w.r.t. the relation $\dot{<}$ defined as $(x \dot{<} y) := (x \dot{\leq} y \wedge \neg(x \doteq y))$. Given a formula ψ and variables X and x, we say that ψ *is x-recursive in X* when the following formula $\mathsf{Rec}^x_X(\psi)$ holds:

$$\mathsf{Rec}^x_X(\psi) \quad := \quad \forall z \forall Z, Z' \left(\forall y \dot{<} z \, [Zy \longleftrightarrow Z'y] \;\longrightarrow\; [\psi[Z/X, z/x] \longleftrightarrow \psi[Z'/X, z/x]]\right)$$

(where z, Z, Z' do not occur free in ψ). For $\psi[X, x]$ x-recursive in X, the Recursion Theorem says that, provably in MSO, the equation $\forall x(Xx \longleftrightarrow \psi[X, x])$ has a unique solution.

▶ **Theorem 4.5** (Recursion Theorem [14]). *If* $\mathsf{MSO} \vdash \mathsf{Rec}^x_X(\psi)$ *then*

$$\begin{array}{lll} & \forall z \left(Zz \longleftrightarrow \forall X \left[\forall x \dot{\leq} z(Xx \leftrightarrow \psi) \longrightarrow Xz\right]\right) & \vdash_{\mathsf{MSO}} \quad \forall x \, (Zx \longleftrightarrow \psi[Z/X]) \\ \textit{and} & \forall x(Zx \longleftrightarrow \psi[Z/X]) \, , \, \forall x(Z'x \longleftrightarrow \psi[Z'/X]) & \vdash_{\mathsf{MSO}} \quad \forall x \, (Zx \longleftrightarrow Z'x) \end{array}$$

▶ **Example 4.6.**

(a) W.r.t. the representation used in Prop. 4.2, let $\theta[\overline{X}, \overline{Q}, Y, x]$ be

$$\forall t \dot{\leq} x(\mathsf{Z}(t) \longrightarrow \mathsf{I}[\overline{Q}(t)]) \quad \wedge \quad \forall t, t' \dot{\leq} x \, (\mathsf{S}(t, t') \longrightarrow \mathsf{H}[\overline{Q}(t), \overline{X}(t), Y(t), \overline{Q}(t')])$$

so that $\delta[\overline{X}, x] = \forall \overline{Q}, Y \left(\theta[\overline{X}, \overline{Q}, Y, x] \to Y(x)\right)$. The Recursion Theorem 4.5 implies that, provably in MSO, for all $\overline{X}$ there are unique predicates $\overline{Q}, Y$ s.t. $\forall x.\theta[\overline{X}, \overline{Q}, Y, x]$. Indeed, assuming I and H are as in (5) we have that $\theta[\overline{X}, \overline{Q}, Y, x]$ is equivalent to $\theta^o[\overline{Q}, \overline{X}, Y, x] \wedge \bigwedge_{1 \leq i \leq q} \theta_i[\overline{Q}, \overline{X}, Y, x]$, where

$$\begin{array}{rrcl} & \theta^o[\overline{X}, \overline{Q}, Y, x] & := & \forall t \dot{\leq} x \, (Y_i(t) \;\longleftrightarrow\; \mathsf{O}_i[\overline{Q}(t), \overline{X}(t)]) \\ & \theta_i[\overline{X}, \overline{Q}, Y, x] & := & \forall t \dot{\leq} x \, (Q_i(t) \;\longleftrightarrow\; \tilde{\theta}_i[\overline{Q}, \overline{X}, t]) \\ \text{with} & \tilde{\theta}_i[\overline{X}, \overline{Q}, t] & := & (\mathsf{Z}(t) \wedge \mathsf{B}_i) \;\vee\; \exists u \dot{\leq} t \left(\mathsf{S}(u, t) \wedge \mathsf{D}_i[\overline{Q}(u), \overline{X}(u)]\right) \end{array}$$

Now, apply Thm. 4.5 to $\mathsf{O}[\overline{Q}(t), \overline{X}(t)]$ (resp. $\tilde{\theta}_i$), which is t-recursive in Y (resp. in Q_i).

(b) The machine of Ex. 2.3.(c) is represented as in (a) with O and D given by (6) (see Ex. 4.3, recalling that the machine as only two states). Hence MSO proves that for all X there are unique Q, Y such that $\forall x.\theta[X, Q, Y, x]$. Continuing now Ex. 4.4, let

$$\delta_2[X, x] \quad := \quad \forall Q, Y \, (\theta[X, Q, Y, x] \longrightarrow Y(x))$$

It is not difficult to derive $\mathsf{MSO} \vdash \phi_0[\delta_2[x]/Y] \wedge \phi_1[\delta_2[x]/Y]$. In order to show $\phi_2[\delta_2[y]/Y]$, one has to prove $\exists^\infty t \; (\neg Xt) \;\vdash_{\mathsf{MSO}}\; \exists^\infty t \; \exists Q, Y(\theta[X, Q, Y, t] \;\wedge\; \neg Yt)$. Thanks to Thm. 4.5, this follows from $\forall x.\theta[X, Q, Y, x] \, , \, \exists^\infty t \; (\neg Xt) \;\vdash_{\mathsf{MSO}}\; \exists^\infty t \; (\neg Yt)$ which itself can be derived using induction.

4.3 From Bounded Formulae to Mealy Machines

We now turn to a useful fact for part (i) of Thm. 3.4, namely, for synchronous comprehension, the extraction of finite-state synchronous functions from bounded formulae. This relies on the standard translation of MSO-formulae *over finite words* to DFA's (see e.g. [16, §3.1]).

▶ **Lemma 4.7.** *Let $\hat{\varphi}$ be a formula with free variables among $z, x_1, \dots, x_p, X_1, \dots, X_q$, and which is bounded by z. Then $\hat{\varphi}$ z-represents a finite-state synchronous $C : \mathbf{2}^p \times \mathbf{2}^q \to_{\mathbf{M}} \mathbf{2}$ induced by a Mealy machine computable from $\hat{\varphi}$.*

▶ Remark. Given $C : \mathbf{2}^p \times \mathbf{2}^q \to_{\mathbf{M}} \mathbf{2}$ z-represented by $\psi\restriction[- \dot{\leq} z]$ (with z not free in ψ), for all $n \in \mathbb{N}$, all $\overline{A} \in (\mathbf{2}^\omega)^q$ and all $\overline{k} \in (\mathbf{2}^\omega)^p$ with $k_i \leq n$, we have $C(\overline{k}, \overline{A})(n) = 1$ if and only if $\langle \overline{k}, \overline{A}\restriction(n+1)\rangle \models \psi$ (in the sense of MSO over finite words). It follows that if C is induced by a Mealy machine $\mathcal{M} = (Q, q^\imath, \partial)$, then with the DFA $\mathcal{A} := (Q \times \mathbf{2} + \{q^\imath\}, q^\imath, \partial_{\mathcal{A}}, Q \times \{1\})$ where $\partial_{\mathcal{A}}(q^\imath, \mathtt{a}) := \partial(q^\imath, \mathtt{a})$ and $\partial_{\mathcal{A}}((q, b), \mathtt{a}) := \partial(q, \mathtt{a})$, we have $C(\overline{k}, \overline{A})(n) = 1$ iff $\mathcal{A}$ accepts the finite word $\langle \overline{k}, \overline{A}\restriction(n+1)\rangle$. Hence $\mathcal{M}$ must pay the price of the non-elementary lower-bound for translating MSO-formulae over finite words to DFAs (see e.g. [7, Chap. 13]). ◀

▶ **Example 4.8.** Recall the continuous but not synchronous function P of Ex. 2.3.(d). The function P can be used to realize a predecessor function, and thus is represented (in the sense of (3)) by a formula $\varphi[X, Y, x]$ such that $\mathfrak{N} \models \varphi[A, B, n]$ iff $A = \{k+1\}$ and $B = \{k\}$ for some $k \leq n$. But φ is not equivalent to a bounded formula, since by Lem. 4.7 bounded formulae represent synchronous functions.

4.4 Internally Bounded Formulae

The synchronous comprehension scheme of MSO is motivated by Lem. 4.7, which tells that uniformly bounded formulae induce Mealy machines. However, being uniformly bounded may seem to be a strict syntactic requirement, and one may wish to relax synchronous comprehension to formulae which behave as bounded formulae, that is to formulae $\psi[\overline{X}, x]$ such that the following formula $\mathsf{B}^x_{\overline{X}}(\psi[\overline{X}, x])$ holds (where $z, \overline{Z}, \overline{Z'}$ do not occur free in ψ):

$$\mathsf{B}^x_{\overline{X}}(\psi[\overline{X}, x]) \quad := \quad \forall z \forall \overline{Z}\,\overline{Z'} (\forall y \dot{\leq} z [\overline{Zz \longleftrightarrow Z'z}] \longrightarrow [\psi[\overline{Z}/\overline{X}, z/x] \longleftrightarrow \psi[\overline{Z'}/\overline{X}, z/x]])$$

▶ **Theorem 4.9.** *If $\mathsf{MSO} \vdash \mathsf{B}^x_{\overline{X}}(\psi[\overline{X}, x])$ and the free variables of ψ are among $x, \overline{X}$, then there is a uniformly bounded formula $\hat{\varphi}[\overline{X}, x]$ which is effectively computable from ψ and such that $\mathsf{MSO} \vdash \forall \overline{X}\, \forall x\, (\psi[\overline{X}, x] \longleftrightarrow \hat{\varphi}[\overline{X}, x])$.*

▶ Remark. Theorem 4.9 relies on the decidability of MSO. Note that Thm. 4.9 in part. applies if $\mathsf{SMSO} \vdash \mathsf{B}^x_{\overline{X}}(\psi[\overline{X}, x])$. Moreover, if $\psi[X, x]$ is recursive (in the sense of §4.2), then $\mathsf{B}^x_X(\psi[X, x])$ holds, but not conversely.

5 The Realizability Interpretation of MSO

This Section presents our realizability model for SMSO, and uses it to prove Thm. 3.4.(i). Our approach to Church's synthesis *via* realizability uses automata in two different ways. First, from a *proof* $\mathscr{D}$ in SMSO of an existential formula $\exists \overline{Y} \varphi(\overline{X}; \overline{Y})$, one can compute a finite-state synchronous realizer $\overline{F}$ of $\varphi(\overline{X}; \overline{Y})$. Second, the adequacy of realizability (and in particular the correctness of $\overline{F}$ w.r.t. $\varphi(\overline{X}; \overline{Y})$) is *proved* using automata for $\varphi(\overline{X}; \overline{Y})$ obtained by McNaughton's Theorem, but these automata do not have to be built concretely.

5.1 Uniform Automata

The adequacy of realizability will be proved using the notion of *uniform automata* (adapted from [13]). In our context, uniform automata are essentially usual non-deterministic automata, but in which non-determinism is expressed *via* an explicitly given set of *moves*. This allows a simple inheritance of the Cartesian structure of synchronous functions (Prop. 2.5), and

thus to interpret the positive existentials of SMSO similarly as usual (weak) sums of type theory. In particular, the set of moves $M(\mathcal{A})$ of an automaton $\mathcal{A}$ interpreting a formula φ will exhibit the strictly positive existentials of φ as $M(\mathcal{A}) = M(\varphi)$ where

$$M(\alpha) \simeq M(\neg\varphi) \simeq \mathbf{1} \qquad M(\varphi \wedge \psi) \simeq M(\varphi) \times M(\psi) \qquad M(\exists(-)\,\varphi) \simeq \mathbf{2} \times M(\varphi) \tag{7}$$

▶ **Definition 5.1** ((Non-Deterministic) Uniform Automata)**.** A (non-deterministic) *uniform automaton* $\mathcal{A}$ over Σ (notation $\mathcal{A} : \Sigma$) has the form

$$\mathcal{A} \quad = \quad (Q_{\mathcal{A}}\,,\, q^{\imath}_{\mathcal{A}}\,,\, M(\mathcal{A})\,,\, \partial_{\mathcal{A}}\,,\, \Omega_{\mathcal{A}}) \tag{8}$$

where $Q_{\mathcal{A}}$ is the finite set of *states*, $q^{\imath}_{\mathcal{A}} \in Q_{\mathcal{A}}$ is the *initial state*, $M(\mathcal{A})$ is the finite non-empty set of *moves*, the *acceptance condition* $\Omega_{\mathcal{A}}$ is an ω-regular subset of $Q^{\omega}_{\mathcal{A}}$, and the *transition function* $\partial_{\mathcal{A}}$ has the form

$$\partial_{\mathcal{A}} \quad : \quad Q_{\mathcal{A}} \times \Sigma \quad \longrightarrow \quad M(\mathcal{A}) \quad \longrightarrow \quad Q_{\mathcal{A}}\,.$$

A *run* of $\mathcal{A}$ on an ω-word $B \in \Sigma^{\omega}$ is an ω-word $R \in M(\mathcal{A})^{\omega}$. We say that R is *accepting* (notation $R \Vdash \mathcal{A}(B)$) if $(q_k)_{k\in\mathbb{N}} \in \Omega_{\mathcal{A}}$ for the sequence of states $(q_k)_{k\in\mathbb{N}}$ defined as $q_0 := q^{\imath}_{\mathcal{A}}$ and $q_{k+1} := \partial_{\mathcal{A}}(q_k, B(k), R(k))$. We say that $\mathcal{A}$ *accepts* B if there exists an accepting run of $\mathcal{A}$ on B, and we let $\mathcal{L}(\mathcal{A})$ be the set of ω-words accepted by $\mathcal{A}$.

Following the usual terminology, an automaton $\mathcal{A}$ as in (8) is *deterministic* if $M(\mathcal{A}) \simeq \mathbf{1}$.

Let us now sketch how uniform automata will be used in our realizability interpretation of SMSO. First, by adapting to our context usual constructions on automata (§5.2), to each MSO-formula φ with free variables among (say) $\overline{X} = X_1, \ldots, X_q$, we associate a uniform automaton $[\![\varphi]\!]$ over $\mathbf{2}^q$ (Fig. 5). Then, from an SMSO-derivation $\mathscr{D}$ of a sequent (say) $\varphi \vdash \psi$ (with free variables among $\overline{X}$ as above), we will extract a finite-state synchronous function $F_{\mathscr{D}} : \mathbf{2}^q \times M([\![\varphi]\!]) \longrightarrow_{\mathbf{M}} M([\![\psi]\!])$, such that $F_{\mathscr{D}}(\overline{B}, R) \Vdash [\![\psi]\!](\overline{B})$ whenever $R \Vdash [\![\varphi]\!](\overline{B})$. In the case of $\vdash \exists Y\, \phi(\overline{X}; Y)$, the finite-state realizer $F_{\mathscr{D}}$ will be of the form $\langle C, G\rangle$ with C and G finite-state synchronous functions $C : \mathbf{2}^q \longrightarrow_{\mathbf{M}} \mathbf{2}$ and $G : \mathbf{2}^q \longrightarrow_{\mathbf{M}} M(\phi)$ such that $G(\overline{B}) \Vdash [\![\phi]\!](\overline{B}, C(\overline{B}))$ for all $\overline{B}$. This motivates the following notion.

▶ **Definition 5.2** (The Category Aut_{Σ})**.** For each alphabet Σ, the category Aut_{Σ}, has automata $\mathcal{A} : \Sigma$ as objects. Morphisms F from $\mathcal{A}$ to $\mathcal{B}$ (notation $\mathcal{A} \Vdash F : \mathcal{B}$) are finite-state synchronous maps $F : \Sigma \times M(\mathcal{A}) \longrightarrow_{\mathbf{M}} M(\mathcal{B})$ such that $F(B, R) \Vdash \mathcal{B}(B)$ whenever $R \Vdash \mathcal{A}(B)$.

▶ Remark.

(a) Note that if $\mathcal{B} \Vdash F : \mathcal{A}$ for some F, then $\mathcal{L}(\mathcal{B}) \subseteq \mathcal{L}(\mathcal{A})$.

(b) One could also consider the category AUT_{Σ} defined as Aut_{Σ}, but with maps not required to be finite-state. All statements of this Section hold for AUT_{Σ}, but for Cor. 5.10, which would lead to non necessarily finite-state realizers and would not give Thm. 3.4.(i).

(c) Uniform automata are a variation of usual automata on ω-words, which is convenient for our purposes, namely the adequacy of our realizability interpretation. Hence, while it would have been possible to define uniform automata with any of the usual acceptance condition (see e.g. [16]), we lose nothing by assuming their acceptance condition to be given by arbitrary ω-regular sets.

5.2 Constructions on Automata

We gather here constructions on uniform automata that we will need to interpret MSO formulae. First, automata are closed under the following operation of *finite substitution*.

▶ **Proposition 5.3.** *Given $\mathcal{A} : \Sigma$ and a function $\mathtt{f} : \Gamma \to \Sigma$, let $\mathcal{A}[\mathtt{f}] : \Gamma$ be the automaton identical to $\mathcal{A}$, but with $\partial_{\mathcal{A}[\mathtt{f}]}(q, \mathtt{b}, u) := \partial_{\mathcal{A}}(q, \mathtt{f}(\mathtt{b}), u)$. Then $B \in \mathcal{L}(\mathcal{A}[\mathtt{f}])$ iff $\mathtt{f} \circ B \in \mathcal{L}(\mathcal{A})$.*

▶ **Example 5.4.** Assume $\mathcal{A}$ interprets a formula φ with free variables among $\overline{X}$, so that $\overline{B} \in \mathcal{L}(\mathcal{A})$ iff $\mathfrak{N} \models \varphi[\overline{B}/\overline{X}]$. Then φ is also a formula with free variables among $\overline{X}, \overline{Y}$, and we have $\overline{B}\overline{B'} \in \mathcal{L}(\mathcal{A}[\pi])$ iff $\mathfrak{N} \models \varphi[\overline{B}/\overline{X}/\overline{B'}/\overline{Y}]$, where $\pi : \overline{X} \times \overline{Y} \to \overline{X}$ is a projection.

The Cartesian structure of $\mathbf{M}$ lifts to Aut_Σ. This gives the interpretation of conjunctions.

▶ **Proposition 5.5.** *For each Σ, the category Aut_Σ has finite products. Its terminal object is the automaton $\mathbf{I} = (\mathbf{1}, \bullet, \mathbf{1}, \partial_{\mathbf{I}}, \mathbf{1}^\omega)$, where $\partial_{\mathbf{I}}(-,-,-) = \bullet$. Binary products are given by*

$$\begin{array}{rrcl} & \mathcal{A} \times \mathcal{B} & := & (Q_{\mathcal{A}} \times Q_{\mathcal{B}}\,,\, (q^\imath_{\mathcal{A}}, q^\imath_{\mathcal{B}})\,,\, M(\mathcal{A}) \times M(\mathcal{B})\,,\, \partial\,,\, \Omega) \\ \textit{where} & \partial((q_{\mathcal{A}}, q_{\mathcal{B}}), \mathtt{a}, (u, v)) & := & (\partial_{\mathcal{A}}(q_{\mathcal{A}}, \mathtt{a}, u)\ ,\ \partial_{\mathcal{B}}(q_{\mathcal{B}}, \mathtt{a}, v)) \end{array}$$

and where $(q_n, q'_n)_n \in \Omega$ iff $((q_n)_n \in \Omega_{\mathcal{A}}$ and $(q'_n)_n \in \Omega_{\mathcal{B}})$. Note that Ω is ω-regular since $\Omega_{\mathcal{A}}$ and $\Omega_{\mathcal{B}}$ are ω-regular (see e.g. [10, Ex. I.11.3.7]). Moreover, $\mathcal{L}(\mathbf{I}) = \Sigma^\omega$ and $\mathcal{L}(\mathcal{A} \times \mathcal{B}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B})$.

Uniform automata are equipped with the obvious adaptation of the usual projection on non-deterministic automata, which interprets existentials. Given a uniform automaton $\mathcal{A} : \Sigma \times \Gamma$, its *projection on* Σ is the automaton

$$(\exists_\Gamma \mathcal{A} : \Sigma) \ := \ (Q_{\mathcal{A}}\,,\, q^\imath_{\mathcal{A}}\,,\, \Gamma \times M(\mathcal{A})\,,\, \partial\,,\, \Omega_{\mathcal{A}}) \quad \text{where} \quad \partial(q, \mathtt{a}, (\mathtt{b}, u)) \ := \ \partial_{\mathcal{A}}(q, (\mathtt{a}, \mathtt{b}), u)$$

▶ **Proposition 5.6.** *Given $\mathcal{A} : \Sigma \times \Gamma$ and $\mathcal{B} : \Sigma$, the realizers $\mathcal{B} \Vdash F : \exists_\Gamma \mathcal{A}$ are exactly the $\mathbf{M}$-pairs $\langle C, G\rangle$ of synchronous functions $C : \Sigma \times M(\mathcal{B}) \to_{\mathbf{M}} \Gamma$ and $G : \Sigma \times M(\mathcal{B}) \to_{\mathbf{M}} M(\mathcal{A})$ such that $G(B, R) \Vdash \mathcal{A}\langle B, C(B, R)\rangle$ for all $B \in A^\omega$ and all $R \Vdash \mathcal{B}(B)$.*

The negation $\neg(-)$ of SMSO is interpreted by an operation $\sim(-)$ on uniform automata which involves McNaughton's Theorem 2.8.

▶ **Proposition 5.7.** *Given a uniform automaton $\mathcal{A} : \Sigma$, there is a uniform* deterministic *$\sim\mathcal{A} : \Sigma$ such that $B \in \mathcal{L}(\sim\mathcal{A})$ iff $B \notin \mathcal{L}(\mathcal{A})$.*

5.3 The Realizability Interpretation

Consider a formula φ with free variables among $\overline{x} = x_1, \ldots, x_p$ and $\overline{X} = X_1, \ldots, X_q$. Its interpretation $[\![\varphi]\!]_{\overline{x},\overline{X}}$ is the uniform automaton over $\mathbf{2}^p \times \mathbf{2}^q$ defined by induction over φ in Fig. 5, where $\mathcal{A}_\alpha$ is a deterministic uniform automaton for the atomic formula α, $\mathsf{Sing} : \mathbf{2}$ is a deterministic uniform automaton accepting the $B \in \mathbf{2}^\omega \simeq \mathcal{P}(\mathbb{N})$ such that B is a singleton, and π, π' are suitable projections. We write $[\![\varphi]\!]$ when $\overline{x}, \overline{X}$ are irrelevant or understood from the context. Note that the set of moves $M(\varphi)$ of $[\![\varphi]\!]$ indeed satisfies (7), so in particular $[\![\delta]\!]$ is deterministic for a deterministic δ. Moreover, as expected we get:

▶ **Proposition 5.8.** *Given an MSO-formula φ with free variables among $\overline{x} = x_1, \ldots, x_p$ and $\overline{X} = X_1, \ldots, X_q$, for all $\overline{k} \in (\mathbf{2}^\omega)^p \simeq (\mathbf{2}^p)^\omega$ and all $\overline{B} \in (\mathbf{2}^\omega)^q \simeq (\mathbf{2}^q)^\omega$, we have $(\overline{k}, \overline{B}) \in \mathcal{L}([\![\varphi]\!]_{\overline{x},\overline{X}})$ iff $\mathfrak{N} \models \varphi[\overline{k}/\overline{x}, \overline{B}/\overline{X}]$.*

Let $\varphi_1, \ldots, \varphi_n, \varphi$ be MSO-formulae with free variables among $\overline{x} = x_1, \ldots, x_p$ and $\overline{X} = X_1, \ldots, X_q$. Then we say that a synchronous function

$$F \quad : \quad \mathbf{2}^p \times \mathbf{2}^q \times M(\varphi_1) \times \cdots \times M(\varphi_n) \quad \longrightarrow_{\mathbf{M}} \quad M(\varphi)$$

realizes the sequent $\varphi_1, \ldots, \varphi_n \vdash \varphi$ (notation $\varphi_1, \ldots, \varphi_n \Vdash F : \varphi$ or $\overline{\varphi} \Vdash F : \varphi$) if

$$[\![\varphi_1]\!]_{\overline{x},\overline{X}} \times \cdots \times [\![\varphi_n]\!]_{\overline{x},\overline{X}} \quad \Vdash \quad F \quad : \quad [\![\varphi]\!]_{\overline{x},\overline{X}}$$

$$[\![\alpha]\!]_{\overline{x},\overline{X}} := \mathcal{A}_\alpha[\pi] \qquad [\![\neg\psi]\!]_{\overline{x},\overline{X}} := \sim[\![\psi]\!]_{\overline{x},\overline{X}} \qquad [\![\exists X\,\psi]\!]_{\overline{x},\overline{X}} := \exists_{\mathbf{2}}([\![\psi]\!]_{\overline{x},\overline{X},X})$$
$$[\![\psi_1 \wedge \psi_2]\!]_{\overline{x},\overline{X}} := [\![\psi_1]\!]_{\overline{x},\overline{X}} \times [\![\psi_2]\!]_{\overline{x},\overline{X}} \qquad [\![\exists x\,\psi]\!]_{\overline{x},\overline{X}} := \exists_{\mathbf{2}}(\mathsf{Sing}[\pi] \times [\![\psi]\!]_{\overline{x},x,\overline{X}}[\pi'])$$

Figure 5 Interpretation of MSO-Formulae as Uniform Automata.

▶ **Theorem 5.9** (Adequacy). *Let $\overline{\varphi}, \varphi$ be* MSO*-formulae with variables among $\overline{x}, \overline{X}$. From an* SMSO*-derivation $\mathscr{D}$ of $\overline{\varphi} \vdash \varphi$, one can compute an* **M***-morphism $F_{\mathscr{D}}$ s.t. $\overline{\varphi} \Vdash_{\overline{x},\overline{X}} F_{\mathscr{D}} : \varphi$.*

Proof. The proof is by induction on derivations. Note that if $\overline{\varphi} \vdash_{\mathsf{SMSO}} \varphi$, then $\overline{\varphi} \models_{\mathfrak{N}} \varphi$. In part., for all rules whose conclusion is of the form $\overline{\varphi} \vdash \delta$ with δ deterministic, it follows from Prop. 5.8 and (7) that the unique **M**-map with codomain $M(\delta) \simeq \mathbf{1}$ (and with appropriate domain) is a realizer. A similar argument applies to the *Ex Falso* rule (elimination of $\bot$), but in this case the realizer of $\overline{\varphi} \vdash \varphi$ is not canonical, and elimination of equality is direct from Prop. 5.8. Adequacy for synchronous comprehension is deferred to §5.3.1. As for the rules of Fig. 4, the first two rules follow from the fact that **M** is a category with finite limits (Prop. 2.5), and the rules for conjunction (resp. existentials) follow from Prop. 5.5 (resp. Prop. 5.6). It remains the rules $\overline{\overline{\varphi} \vdash \exists y\,\mathsf{Z}(y)}$ and $\overline{\overline{\varphi} \vdash \exists y\,\mathsf{S}(x,y)}$ of Fig. 3. For the latter, we use the Mealy machine depicted on Fig. 1 (left) (Ex. 2.3.(b)) together with the fact that $\mathsf{S}(-,-)$ is deterministic. The case of the former is similar and simpler. ◀

Adequacy of realizability, together with Prop. 5.6, directly gives Thm. 3.4.(i).

▶ **Corollary 5.10** (Thm. 3.4.(i)). *Given a derivation $\mathscr{D}$ in* SMSO *of $\vdash \exists\overline{Y}\,\varphi(\overline{X};\overline{Y})$ with $\overline{X} = X_1,\dots,X_q$ and $\overline{Y} = Y_1,\dots,Y_p$, we have $F_{\mathscr{D}} = \langle \overline{C}, G\rangle$ where $\overline{C} = C_1,\dots,C_p$ with $C_i : \mathbf{2}^q \longrightarrow_{\mathbf{M}} \mathbf{2}$ and $\mathfrak{N} \models \varphi(\overline{B}, \overline{C}(\overline{B}))$ for all $\overline{B} \in (\mathbf{2}^\omega)^q \simeq (\mathbf{2}^q)^\omega$.*

5.3.1 Realization of Synchronous Comprehension

We now turn to the adequacy of the synchronous comprehension rule. It directly follows from the existence of finite-state characteristic functions for bounded formulae (Lem. 4.7) and from the following semantic substitution lemma, which allows, given a synchronous function $C_{\hat{\varphi}}$ y-represented by $\hat{\varphi}$, to lift a realizer of $\psi[\hat{\varphi}[y]/Y]$ into a realizer of $\exists Y\,\psi$.

▶ **Lemma 5.11.** *Let $\overline{x} = x_1,\dots,x_p$ and $\overline{X} = X_1,\dots,X_q$. Let $\hat{\varphi}$ be a formula with free variables among $y, \overline{X}$, and assume that $\hat{\varphi}$ y-represents $C_{\hat{\varphi}} : \mathbf{2}^q \longrightarrow_{\mathbf{M}} \mathbf{2}$. Then for every* MSO*-formula ψ with free variables among $\overline{x}, \overline{X}$, there is a finite-state synchronous function*

$$H_\psi \quad : \quad M(\psi[\hat{\varphi}[y]/Y]) \quad \longrightarrow_{\mathbf{M}} \quad M(\psi)$$

such that for all $\overline{k} \in (\mathbf{2}^\omega)^p$, all $\overline{A} \in (\mathbf{2}^\omega)^q$ and all $R \in M(\psi)^\omega$, we have

$$R \Vdash [\![\psi[\hat{\varphi}[y]/Y]]\!]_{\overline{x},\overline{X}}(\overline{k},\overline{A}) \qquad \Longrightarrow \qquad H_\psi(R) \Vdash [\![\psi]\!]_{\overline{x},\overline{X},Y}(\overline{k},\overline{A}, C_{\hat{\varphi}}(\overline{A})) \tag{9}$$

Adequacy of synchronous comprehension then directly follows.

▶ **Lemma 5.12.** *Let ψ with free variables among $\overline{x}, \overline{X}, Y$ and let $\hat{\varphi}$ be a formula with free variables among $y, \overline{X}$ and which is uniformly bounded by y. Then there is a finite-state realizer $\psi[\hat{\varphi}[y]/Y] \Vdash_{\overline{x},\overline{X}} F : \exists Y\,\psi$, effectively computable from ψ and φ.*

Proof. Let $C_{\hat{\varphi}}$ satisfying (3) be given by Lem. 4.7, and let H_ψ satisfying (9) be given by Lem. 5.11. It then directly follows from Prop. 5.6 and Len. 5.11 that $\psi[\hat{\varphi}[y]/Y] \Vdash_{\overline{x},\overline{X}} \langle C_{\hat{\varphi}} \circ [\pi], H_\psi \circ [\pi']\rangle : \exists Y\,\psi$, where π, π' are suitable projections. ◀

6 Conclusion

In this paper, we revisited Church's synthesis *via* an automata-based realizability interpretation of an intuitionistic proof system SMSO for MSO on ω-words, and we demonstrated that our approach is sound and complete, in the sense of Thm. 3.4. As it stands, this approach must still pay the price of the non-elementary lower-bound for the translation of MSO formulae over finite words to DFA's (see the Remark after Lem. 4.7, §4.3) and the system SMSO is limited by its set of connectives and its restricted induction scheme.

Further Works. First, following the approach of [13], SMSO could be extended with primitive universal quantifications and implications as soon as one goes to a *linear* deduction system. In particular, primitive universals and implications would allow to extend the logic with atomic formulae for Mealy machines with defining axioms of the form (4). We expect this to give better lower bounds w.r.t. completeness (for each solvable instance of Church's synthesis, to provide proofs with realizers of a reasonable complexity). Among other outcomes of going to a linear deduction system, following [13] we expect similar proof-theoretical properties as with the usual *Dialectica* interpretation (see e.g. [8]), such as realizers of linear Markov rules and choices schemes. On the other hand, we do not know yet if effective computations of modulus of uniform continuity could be pertinent for Church's synthesis (e.g. for a non-linear Markov rule). Moreover, we expect that a linear variant of MSO on finite words could help (for some classes of formulae) to mitigate the Remark of §4.3.

The case of induction is more complex. One possibility to have finite-state realizers for a more general induction rule would be to rely on saturation techniques for regular languages. Another possibility, which may be of practical interest, is to follow the usual Curry-Howard approach and allow possibly infinite-state realizers.

Another direction of future work is to incorporate specific reasoning principles on Mealy machines. For instance, a translation from (a subsystem of) [2] could be interesting.

Acknowledgements. We thank the anonymous referees for helpful comments.

References

1 R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Sa'ar. Synthesis of reactive (1) designs. *Journal of Computer and System Sciences*, 78(3):911–938, 2012.

2 M. Bonsangue, J. Rutten, and A. Silva. Coalgebraic logic and synthesis of Mealy machines. In *Proceedings of FOSSACS'08*, pages 231–245. Springer, 2008.

3 J. R. Büchi. On a Decision Methond in Restricted Second-Order Arithmetic. In E. Nagel et al., editor, *Logic, Methodology and Philosophy of Science (Proc. 1960 Intern. Congr.)*, pages 1–11. Stanford Univ. Press, 1962.

4 J. R. Büchi and L. H. Landweber. Solving Sequential Conditions by Finite-State Strategies. *Transation of the American Mathematical Society*, 138:367–378, 1969.

5 A. Church. Applications of recursive arithmetic to the problem of circuit synthesis. In *Summaries of the SISL*, volume 1, pages 3–50. Cornell Univ., 1957.

6 J.-Y. Girard. *Interprétation Fonctionnelle et Élimination des Coupures de l'Arithmétique d'Ordre Supérieur*. PhD thesis, Université Paris 7, 1972.

7 E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002.

8 U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Springer Monographs in Mathematics. Springer, 2008.

9 R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521–530, 1966.

10 D. Perrin and J.-É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Pure and Applied Mathematics. Elsevier, 2004.

11 M. O. Rabin. Automata on infinite objects and Church's Problem. *Amer. Math. Soc.*, 1972.

12 C. Riba. A model theoretic proof of completeness of an axiomatization of monadic second-order logic on infinite words. In *Proceedings of IFIP-TCS'12*, 2012.

13 C. Riba. A Dialectica-Like Approach to Tree Automata. Available on HAL (hal-01261183), `https://hal.archives-ouvertes.fr/hal-01261183`, 2016.

14 D. Siefkes. *Decidable Theories I: Büchi's Monadic Second Order Successor Arithmetic*, volume 120 of *LNM*. Springer, 1970.

15 The Coq Developement Team. *The Coq Proof Assistant Reference Manual*, 2016. `http://coq.inria.fr/`.

16 W. Thomas. Languages, Automata, and Logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume III, pages 389–455. Springer, 1997.

17 W. Thomas. Solution of Church's Problem: A tutorial. *New Perspectives on Games and Interaction*, 5:23, 2008.