# A CURRY-HOWARD APPROACH TO CHURCH'S SYNTHESIS *

PIERRE PRADIC AND COLIN RIBA

ENS de Lyon, Université de Lyon, LIP and University of Warsaw, Faculty of Mathematics, Informatics and Mechanics
*e-mail address*: `pierre.pradic@ens-lyon.fr`

ENS de Lyon, Université de Lyon, LIP
*e-mail address*: `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.

## 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 $\omega$-words. MSO over $\omega$-words is a decidable logic by Büchi's Theorem [3]. It subsumes non-trivial logics used in verification such as LTL (see e.g. [22, 15]).

Traditional approaches to synthesis (see e.g. [23, 24]) are based, via McNaughton's Theorem [14], on the translation of MSO-formulae to *deterministic* automata on $\omega$-words (such as *Muller* or *parity* automata)[1]. Such automata are then turned into game graphs, in which the *Opponent* O ($\forall$bélard) plays input characters to which the *Proponent* P ($\exists$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. [22, 15]). 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. This paper is an extended version of the conference article [16].

[1] A solution is also possible via tree automata [17] (see also [11, 24]).

LOGICAL METHODS
IN COMPUTER SCIENCE
DOI:10.2168/LMCS-???

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 [21]) 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 $\omega$-words can be completely axiomatized as a subsystem of second-order Peano arithmetic [20] (see also [18]). 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. [22]), 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. Finally, in §6 we present our realizability model in terms of *indexed categories* (see e.g. [9]), an essential step for further generalizations.

We also have included three appendices. They give detailed arguments and constructions that we did not wanted to put in the body of the paper, either because they are necessary but unsurprising technicalities (Appendices A and C), or because they concern important but side results, proved with different techniques than those emphasized in this paper (Appendix B).

---

[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. [10]). 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.

## 2. Church's Synthesis and MSO on Infinite Words

2.1. **Notations.** Alphabets (denoted $\Sigma, \Gamma$, etc) are finite non-empty sets. Concatenation of words $s, t$ is denoted either $s.t$ or $s \cdot t$, and $\varepsilon$ 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 $\overline{\mathtt{a}}$ denotes a word $\mathtt{a}_1. \cdots .\mathtt{a}_n \in \Sigma^*$. Given an $\omega$-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 $n = k$.

2.2. **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. [23]). 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 [23]) expressed by

$$\Phi(A,B) \quad \overset{\text{def.}}{\Longleftrightarrow} \quad \left\{ \begin{array}{lcll} \forall n(A(n) = 1 & \Longrightarrow & B(n) = 1) & \text{and} \\ \forall n(B(n) = 0 & \Longrightarrow & B(n+1) = 1) & \text{and} \\ (\exists^\infty n\ A(n) = 0) & \Longrightarrow & (\exists^\infty n\ B(n) = 0) & \end{array} \right. \tag{2.1}$$

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 $\Sigma$ and output alphabet $\Gamma$ (notation $\mathcal{M} : \Sigma \to \Gamma$) is given by a finite set of states $Q_\mathcal{M}$ with a distinguished initial state $q^\imath_\mathcal{M} \in Q_\mathcal{M}$, and a transition function $\partial_\mathcal{M} : Q_\mathcal{M} \times \Sigma \to Q_\mathcal{M} \times \Gamma$.

We write $\partial^o_\mathcal{M}$ for $\pi_2 \circ \partial_\mathcal{M} : Q_\mathcal{M} \times \Sigma \to \Gamma$ and $\partial^*_\mathcal{M}$ for the map $\Sigma^* \to Q_\mathcal{M}$ obtained by iterating $\partial_\mathcal{M}$ from the initial state: $\partial^*_\mathcal{M}(\varepsilon) := q^\imath_\mathcal{M}$ and $\partial^*_\mathcal{M}(\overline{\mathtt{a}}.\mathtt{a}) := \pi_1(\partial_\mathcal{M}(\partial^*_\mathcal{M}(\overline{\mathtt{a}}), \mathtt{a}))$.

A Mealy machine $\mathcal{M} : \Sigma \to \Gamma$ induces a function $F : \Sigma^\omega \to \Gamma^\omega$ obtained by iterating $\partial^o_\mathcal{M}$ along the input: $F(B)(n) = \partial^o_\mathcal{M}(\partial^*_\mathcal{M}(B{\restriction}n), B(n))$. 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.

**Examples 2.3.**

(1) The identity function $\Sigma^\omega \to \Sigma^\omega$ is induced by the Mealy machine with state set $\mathbf{1} = \{\bullet\}$ and identity transition function $\partial : (\bullet, \mathtt{a}) \longmapsto (\bullet, \mathtt{a})$.
(2) 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 $B(n) = 1$.
(3) The Mealy machine depicted on Fig. 1 (right), taken from [23], induces a synchronous function which realizes the specification (2.1).
(4) Synchronous functions are obviously continuous (taking the product topology on $\Sigma^\omega$ and $\Gamma^\omega$, with $\Sigma, \Gamma$ 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.

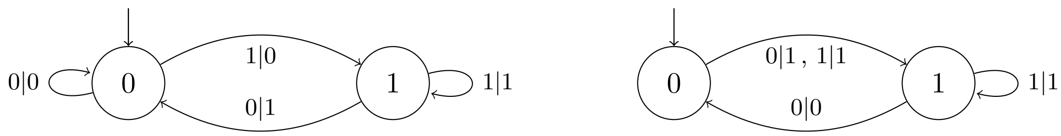



Figure 1: Examples of Mealy Machines (where a transition $\mathtt{a}|\mathtt{b}$ outputs $\mathtt{b}$ from input $\mathtt{a}$).

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

Figure 2: The Formulae of $\mathsf{MSO}$.

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

The following obvious fact will be useful for our realizability model in §5.

**Remark 2.5.** Functions $\mathtt{f} : \Sigma \to \Gamma$ induce $\mathbf{M}$-maps $[\mathtt{f}] : \Sigma \to_{\mathbf{M}} \Gamma$.

It is also worth noticing that the category $\mathbf{M}$ has finite products.

**Proposition 2.6.** *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 $\mathbf{M}$).*

**Remark 2.7.** Recall from e.g. [13] that categorical products require the unicity of tupling morphisms. On the other hand, note that a given $\mathbf{M}$-morphism can be induced by several (actually infinitely many) different Mealy machines. As a consequence, Proposition 2.6 would have been wrong if for $\mathbf{M}$-morphisms one would have taken Mealy *machines* instead of *functions*.

### 2.3. **Monadic Second-Order Logic ($\mathsf{MSO}$) on Infinite Words.**

We consider a formulation of $\mathsf{MSO}$ 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 $\mathsf{MSO}$, denoted $\varphi, \psi,$ etc are given on Fig. 2. They are defined by mutual induction with the *deterministic formulae* (denoted $\delta, \delta',$ etc) from atomic formulae ranged over by $\alpha$.

$\mathsf{MSO}$ formulae are interpreted in the standard model $\mathfrak{N}$ of $\omega$-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 $\mathbb{N}$, $\mathsf{S}$ is the successor relation, and $\mathsf{Z}$ holds on $n$ iff $n = 0$. We often write $X(x)$ or even $Xx$ for $x \mathrel{\dot{\in}} X$. As usual we let:

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

$\mathsf{MSO}$ on $\omega$-words is known to be decidable by Büchi's Theorem [3].

**Theorem 2.8** (Büchi [3])**.** $\mathsf{MSO}$ *over $\mathfrak{N}$ is decidable.*

Following [3] (but see also e.g. [15]), the (non-deterministic) automata method for deciding $\mathsf{MSO}$ proceeds by a recursive translation of $\mathsf{MSO}$-formulae to *Büchi automata*. A *Büchi automaton* is a non-deterministic finite state automaton running on $\omega$-words. Büchi automata are equipped with a set of final states, and a run on an $\omega$-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. [22, 8]) on the complementation of Büchi automata. First, the translation of $\mathsf{MSO}$-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.

2.4. **Church's Synthesis for $\mathsf{MSO}$.** Church's synthesis problem for $\mathsf{MSO}$ is the following. Given as input an $\mathsf{MSO}$ formula $\varphi(\overline{X};\overline{Y})$ (where $\overline{X} = X_1,\dots,X_q$ and $\overline{Y} = Y_1,\dots,Y_p$), (1) decide whether there exist finite-state synchronous functions $\overline{F} = F_1,\dots,F_p : \mathbf{2}^q \to_{\mathbf{M}} \mathbf{2}$ 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 2.9.** The specification $\Phi$ displayed in (2.1) can be officially written in the language of $\mathsf{MSO}$ as the following formula $\phi(X;Y)$ (where $\exists^\infty t\ \varphi(t)$ stands for $\forall x \exists t(t \mathrel{\dot{\geq}} x \land \varphi(t))$):

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

Church's synthesis has been solved by Büchi & Landweber [4], using automata on $\omega$-words and infinite two-player games (a solution is also possible *via* tree automata [17]): 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 $\mathsf{MSO}$ (see e.g. [23]) proceeds *via* McNaughton's Theorem ([14], see also e.g. [15, 22]), 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. [22, 8]). 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 2.10** (McNaughton [14])**.** *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

[4]It follows from the finite-state determinacy of $\omega$-regular games that a finite-state synchronous realizer exists whenever a synchronous realizer exists (see e.g. [23]).

**Equality Axioms:**

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

**Partial Order Axioms:**

$$\frac{}{\overline{\varphi} \vdash x \mathrel{\dot{\leq}} x} \qquad \frac{\overline{\varphi} \vdash x \mathrel{\dot{\leq}} y \qquad \overline{\varphi} \vdash y \mathrel{\dot{\leq}} z}{\overline{\varphi} \vdash x \mathrel{\dot{\leq}} z} \qquad \frac{\overline{\varphi} \vdash x \mathrel{\dot{\leq}} y \qquad \overline{\varphi} \vdash y \mathrel{\dot{\leq}} x}{\overline{\varphi} \vdash x \doteq y}$$

**Basic $\mathsf{Z}$ and $\mathsf{S}$ Axioms (total injective relations):**

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

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

**Arithmetic Axioms:**

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

Figure 3: Arithmetic Rules of $\mathsf{MSO}$ and $\mathsf{SMSO}$.

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 $\mathsf{MSO}$ 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 $\omega$-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 $\omega$-regular games on finite graphs are effectively determined, and moreover that the winner always has a finite state winning strategy.

**Example 2.11.** 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. 2.9. When translating $\phi_2$ 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 $(\forall^\infty t\ Xt) = (\exists n \forall t \mathrel{\dot{\geq}} n\, Xt)$ 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 2.10 is in general required for positive occurrences of the form $\forall^\infty t\ (-)$.

2.5. **An Axiomatization of $\mathsf{MSO}$.** Our approach to Church's synthesis relies on the fact that the $\mathsf{MSO}$-theory of $\mathfrak{N}$ can be completely axiomatized as a subsystem of second-order Peano arithmetic [20] (see also [18]). We consider a specific set of axioms which consists of the rules depicted on Fig. 3 together with the following *comprehension* and *induction* rules

$$\frac{\overline{\varphi} \vdash \varphi[\psi[y]/X]}{\overline{\varphi} \vdash \exists X\, \varphi} \qquad \frac{\overline{\varphi}, \mathsf{Z}(z) \vdash \varphi[z/x] \qquad \overline{\varphi}, \mathsf{S}(y, z), \varphi[y/x] \vdash \varphi[z/x]}{\overline{\varphi} \vdash \varphi} \tag{2.2}$$

$$
\begin{array}{rl}
(1) & \vdash \neg(x \mathrel{\dot{<}} x) \\
(2) & x \mathrel{\dot{<}} y, y \mathrel{\dot{<}} z \vdash x \mathrel{\dot{<}} z \\
(3) & \mathsf{S}(x,y), x \doteq y \vdash \bot \\
(4) & \vdash \forall x \exists y\ x \mathrel{\dot{<}} y \\
(5) & \mathsf{S}(y,y'), x \mathrel{\dot{\leq}} y, x \doteq y' \vdash \bot \\
(6) & \mathsf{Z}(x) \vdash x \mathrel{\dot{\leq}} y \\
(7) & x \mathrel{\dot{\leq}} y, \mathsf{Z}(y) \vdash \mathsf{Z}(x) \\
(8) & \forall y (x \mathrel{\dot{\leq}} y) \vdash \mathsf{Z}(x) \\
(9) & x \mathrel{\dot{<}} y, \mathsf{S}(x,x') \vdash x' \mathrel{\dot{\leq}} y \\
(10) & x \mathrel{\dot{\leq}} y, \mathsf{S}(x,x'), \mathsf{S}(y,y') \vdash x' \mathrel{\dot{\leq}} y' \\
(11) & \vdash \forall x \forall y \left[ y \mathrel{\dot{<}} x \quad \longleftrightarrow \quad \exists z (y \mathrel{\dot{\leq}} z \ \land\ \mathsf{S}(z,x)) \right] \\
(12) & \vdash x \mathrel{\dot{<}} y \lor x \doteq y \lor y \mathrel{\dot{<}} x \\
(13) & \vdash \forall x, y \left[ \mathsf{S}(x,y) \quad \longleftrightarrow \quad (x \mathrel{\dot{<}} y \ \land\ \neg \exists z (x \mathrel{\dot{<}} z \mathrel{\dot{<}} y)) \right]
\end{array}
$$

Figure 4: Arithmetic Lemmas of $\mathsf{MSO}$.

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 2.12** ([20])**.** *For every (closed)* $\mathsf{MSO}$*-formula* $\varphi$*, 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 (2.2).*

Actually obtaining Theorem 2.12 from [18] requires some work. The difference between [18] and the present system is that the axiomatization of [18] is expressed in terms of the strict part of $\dot{\leq}$ (written $\dot{<}$) and that comprehension is formulated with the following usual axiom scheme (where $X$ is not free in $\varphi$):

$$\exists X \forall x \, (X(x) \quad \longleftrightarrow \quad \varphi[x/y]) \tag{2.3}$$

We state here the properties required to bridge the gap between [18] and the present axiomatization of $\mathsf{MSO}$. Missing details are provided in Appendix A. First, the comprehension scheme of the present version of $\mathsf{MSO}$ directly implies (2.3), since using

$$\forall x \, (\varphi[x/y] \quad \longleftrightarrow \quad \varphi[x/y]) \qquad = \qquad \forall x \, (X(x) \quad \longleftrightarrow \quad \varphi[x/y]) \, [\varphi[y]/X]$$

we have

$$\dfrac{\dfrac{}{\vdash \forall x \, (\varphi[x/y] \quad \longleftrightarrow \quad \varphi[x/y])}}{\vdash \exists X \forall x \, (X(x) \quad \longleftrightarrow \quad \varphi[x/y])}$$

In order to deal with the $\dot{<}$-axioms of [18], we rely on a series of arithmetical lemmas of $\mathsf{MSO}$. Let

$$(x \mathrel{\dot{<}} y) \quad := \quad \left[ x \mathrel{\dot{\leq}} y \ \land\ \neg(x \doteq y) \right]$$

**Lemma 2.13.** $\mathsf{MSO}$ *proves all the sequents of Figure 4.*

Finally, the induction axiom of [18] is the usual *strong induction* axiom:

$$\vdash \forall X\,[\forall x(\forall y(y \mathrel{\dot{<}} x \to X(y)) \to X(x)) \to \forall x\ X(x)] \tag{2.4}$$

**Lemma 2.14.** $\mathsf{MSO}$ *proves the strong induction axiom (2.4).*

The detailed proofs of Lemmas 2.13 and 2.14 are deferred to Appendix A.

## 3. SMSO: A Synchronous Intuitionistic Restriction of MSO

We now introduce $\mathsf{SMSO}$, an intuitionistic restriction of $\mathsf{MSO}$. As expected, $\mathsf{SMSO}$ contains $\mathsf{MSO}$ *via* negative translation. But thanks to its vocabulary without primitive universals, $\mathsf{SMSO}$ actually admits a Glivenko Theorem, so that $\mathsf{SMSO}$ proves $\neg\neg\varphi$ whenever $\mathsf{MSO} \vdash \varphi$. Moreover, $\mathsf{SMSO}$ 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, $\mathsf{SMSO}$ has the same language as $\mathsf{MSO}$, 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. [10]), $\mathsf{SMSO}$ has a comprehension scheme which corresponds to the negative translation of the full comprehension scheme of $\mathsf{MSO}$[5]. On the other hand, for the extraction of *synchronous* realizers from proofs, $\mathsf{SMSO}$ has a restricted induction scheme corresponding to the negative translation of the induction scheme of $\mathsf{MSO}$. 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 $\mathsf{MSO}$ can be interpreted by *deterministic Büchi* automata, and have a trivial computational 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 $\mathsf{SMSO}$ counterpart of the formulae of Heyting arithmetic admitting elimination of double negation (see e.g. [10]).

Furthermore, $\mathsf{SMSO}$ 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.**

(1) Given $\mathsf{MSO}$-formulae $\varphi$ and $\theta$ and a variable $y$, the *relativization of* $\varphi$ *to* $\theta[y]$ (notation $\varphi{\restriction}\theta[y]$), is defined by induction on $\varphi$ 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 $\theta$. Note that $y$ does not occur free in $\varphi{\restriction}\theta[y]$.

[5] In contrast with Girard's System F [7], 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 5: Logical Rules of $\mathsf{SMSO}$ (where $\delta$ is deterministic).

(2) An $\mathsf{MSO}$-formula $\hat{\varphi}$ is *bounded by* $x$ if it is of the form $\psi{\upharpoonright}(y \mathrel{\dot{\leq}} x)[y]$ (notation $\psi{\upharpoonright}[- \mathrel{\dot{\leq}} x]$). 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 $\mathsf{MSO}$ over finite words. We are now ready to define the system $\mathsf{SMSO}$.

**Definition 3.2** ($\mathsf{SMSO}$)**.** The logic $\mathsf{SMSO}$ has the same language as $\mathsf{MSO}$. Its deduction rules are those given in Fig. 5 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 3.3.** The axiom $\overline{\varphi}, \neg\neg\delta \vdash \delta$ of double negation elimination for deterministic formulae would already be derivable in a version of $\mathsf{SMSO}$ 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.

3.1. **A Glivenko Theorem for $\mathsf{SMSO}$.** Thanks to its limited vocabulary, $\mathsf{SMSO}$ satisfies a Glivenko theorem, and thus a very simple negative translation from $\mathsf{MSO}$. 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 $\mathsf{SMSO}$, 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 3.4.** *If* $\mathsf{MSO} \vdash \varphi$*, then* $\mathsf{SMSO} \vdash \neg\neg\varphi$*.*

*Proof.* By induction on $\mathsf{MSO}$ derivations, we show that if $\overline{\varphi} \vdash \varphi$ is derivable in $\mathsf{MSO}$, then $\overline{\varphi} \vdash \neg\neg\varphi$ is derivable in $\mathsf{SMSO}$. This amounts to showing that for every $\mathsf{MSO}$ rule of the form

$$\frac{(\overline{\varphi}_i \vdash \varphi_i)_{i \in I}}{\overline{\psi} \vdash \psi}$$

the following rule is admissible in $\mathsf{SMSO}$:

$$\frac{(\overline{\varphi}_i \vdash \neg\neg\varphi_i)_{i\in I}}{\overline{\psi} \vdash \neg\neg\psi}$$

We implicitly use the admissibility of *weakenings* in $\mathsf{SMSO}$, *i.e.* the admissibility of the rule

$$\frac{\overline{\varphi} \vdash \varphi}{\overline{\varphi}, \psi \vdash \varphi}$$

The propositional rules may be treated exactly as in the usual proof of Glivenko's theorem for propositional logic, and it is folklore that Glivenko's theorem extends to existential quantifications (see e.g. [10, Prop. 10.3]). It remains to deal with the axioms of $\mathsf{MSO}$.

**Comprehension:** The comprehension rule of $\mathsf{MSO}$ (which behaves as an extended rule of $\exists$-introduction) requires to exploit the introduction rule for existentials and the double-negated version of comprehension of $\mathsf{SMSO}$. We have to prove that the following is admissible in $\mathsf{SMSO}$:

$$\frac{\overline{\varphi} \vdash \neg\neg\psi[\varphi[y]/X]}{\overline{\varphi} \vdash \neg\neg\exists X\, \psi}$$

Note that we are done as soon as we show that $\mathsf{SMSO}$ proves

$$\overline{\varphi}, \neg\exists X\, \psi \vdash \neg\psi[\varphi[y]/X] \tag{3.1}$$

Indeed, from $\overline{\varphi} \vdash \neg\neg\psi[\varphi[y]/X]$ we get $\overline{\varphi}, \neg\exists X\, \psi \vdash \neg\neg\psi[\varphi[y]/X]$, which together with (3.1) allows to derive $\overline{\varphi}, \neg\exists X\, \psi \vdash \bot$. We then trivially get $\overline{\varphi} \vdash \neg\neg\exists X\, \psi$.

A derivation of (3.1) is given on Figure 6.

**Induction:** We need to show that

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

Since $\neg\neg\varphi$ is deterministic, $\mathsf{SMSO}$ derives

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

so we may conclude by noticing that

$$\frac{\overline{\varphi}, \mathsf{S}(y,z), \varphi[y/x] \vdash \neg\neg\varphi[z/x]}{\overline{\varphi}, \mathsf{S}(y,z), \neg\neg\varphi[y/x] \vdash \neg\neg\varphi[z/x]}$$

is derivable in intuitionistic propositional logic.

**Elimination of Equality:** We have to show that the following rule is admissible in $\mathsf{SMSO}$:

$$\frac{\overline{\varphi} \vdash \neg\neg\varphi[t/z] \qquad \overline{\varphi} \vdash \neg\neg(t \doteq u)}{\overline{\varphi} \vdash \neg\neg\varphi[u/z]}$$

Since $t \doteq u$ is deterministic, by cutting the right premise with the deterministic double-negation elimination rule of $\mathsf{SMSO}$, we are left with deriving the following in $\mathsf{SMSO}$:

$$\frac{\overline{\varphi} \vdash \neg\neg\varphi[t/z] \qquad \overline{\varphi} \vdash t \doteq u}{\overline{\varphi} \vdash \neg\neg\varphi[u/z]}$$

But this is an instance of the rule of elimination of equality.

$$\dfrac{\dfrac{\dfrac{}{\overline{\varphi}\,,\,\neg\exists X\,\psi\,,\,\psi[\varphi[y]/X]\vdash\psi[\varphi[y]/X]}}{\overline{\varphi}\,,\,\neg\exists X\,\psi\,,\,\psi[\varphi[y]/X]\vdash\neg\neg\exists X\,\psi}\qquad\dfrac{}{\overline{\varphi}\,,\,\neg\exists X\,\psi\,,\,\psi[\varphi[y]/X]\vdash\neg\exists X\,\psi}}{\dfrac{\overline{\varphi}\,,\,\neg\exists X\,\psi\,,\,\psi[\varphi[y]/X]\vdash\bot}{\overline{\varphi}\,,\,\neg\exists X\,\psi\vdash\neg\psi[\varphi[y]/X]}}$$

Figure 6: Derivation of (3.1) in $\mathsf{SMSO}$ (Proof of Thm. 3.4).

**Other Arithmetic Axioms of $\mathsf{MSO}$ (Fig. 3):** All these rules have for premises and conclusion sequents of the form $\overline{\varphi}\vdash\delta$ where $\delta$ is deterministic. The premises of these rules may be cut with the deterministic double-negation elimination rule of $\mathsf{SMSO}$. Supposing that the conclusion of such a rule is of shape $\overline{\varphi}\vdash\varphi$, we conclude from the fact that the following rule is derivable in $\mathsf{SMSO}$:

$$\dfrac{\overline{\varphi}\vdash\varphi}{\overline{\varphi}\vdash\neg\neg\varphi}$$

$\square$

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

**Theorem 3.5** (Main Theorem)**.** *Consider an* $\mathsf{MSO}$*-formula* $\varphi(\overline{X};\overline{Y})$*.*

(1) *From a proof of* $\exists\overline{Y}\,\neg\neg\varphi(\overline{X};\overline{Y})$ *in* $\mathsf{SMSO}$*, one can extract a finite-state synchronous realizer of* $\varphi(\overline{X};\overline{Y})$*.*
(2) *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 (1) of Thm. 3.5 will be proved in §5 using a notion of realizability for $\mathsf{SMSO}$ based on automata and synchronous finite-state functions. The completeness part (2) will be proved in §4.1, relying the completeness of the axiomatization of $\mathsf{MSO}$ (Thm. 2.12) together with the correctness of the negative translation $\neg\neg(-)$ (Thm. 3.4).

## 4. On the Representation of Mealy Machines in $\mathsf{MSO}$

This section gathers several (possibly known) results related to the representation of Mealy machines in $\mathsf{MSO}$. We begin in §4.1 with the completeness part of Thm. 3.5, which follows usual representations of automata in $\mathsf{MSO}$ (see e.g. [22, §5.3]). We then recall from [20, 18] the *Recursion Theorem*, which is a convenient tool to reason on runs of deterministic automata in $\mathsf{MSO}$ (§4.2). In §4.3 we state a Lemma for the correctness part of Thm. 3.5, which relies on the usual translation of $\mathsf{MSO}$-formulae over *finite words* to DFA's (see e.g. [22, §3.1]). Finally, in §4.4 we give a possible strengthening of the synchronous comprehension rule of $\mathsf{SMSO}$ (but which is based on Büchi's Theorem 2.8).

We work with the following notion of representation.

**Definition 4.1.** Let $\varphi$ be a formula with free variables among $z, x_1, \ldots, x_p, X_1, \ldots, X_q$. We say that $\varphi$ *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

$$F(\overline{k},\overline{A})(n)=1 \qquad\text{iff}\qquad \mathfrak{N}\models\varphi[n/z,\overline{k}/\overline{x},\overline{A}/\overline{X}] \tag{4.1}$$

4.1. **Internalizing Mealy Machines in** $\mathsf{MSO}$**.** The completeness part (2) of Thm. 3.5 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. [22, §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

$$\delta[\overline{X}, x] \;\; := \;\; \forall \overline{Q}, Y \left( \left[ \begin{array}{l} \forall t \mathrel{\dot{\leq}} x (\mathsf{Z}(t) \to \mathsf{I}[\overline{Q}(t)]) \quad \wedge \\ \forall t, t' \mathrel{\dot{\leq}} x \, (\mathsf{S}(t, t') \to \mathsf{H}[\overline{Q}(t), \overline{X}(t), Y(t), \overline{Q}(t')]) \end{array} \right] \longrightarrow \; Y(x) \right) \tag{4.2}$$

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

**Remark 4.3.** 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

$$\big(Y(t) \longleftrightarrow \mathsf{O}[\overline{Q}(t), \overline{X}(t)]\big) \quad \wedge \quad \bigwedge_{1 \leq i \leq q} \big(Q_i(t') \longleftrightarrow \mathsf{D}_i[\overline{Q}(t), \overline{X}(t)]\big) \tag{4.3}$$

**Example 4.4.** The function induced by the Mealy machine of Ex. 2.3.(3) (depicted on Fig. 1, right), is represented by a formula of the form (4.2), 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

$$\mathsf{O}[Q(t), X(t)] \quad = \quad \mathsf{D}[Q(t), X(t)] \quad = \quad (\neg Q(t) \;\vee\; [Q(t) \wedge X(t)]) \tag{4.4}$$

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

*Proof of Thm. 3.5.(2).* 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. 2.6), we assume $C = \overline{C} = C_1, \ldots, C_p$ 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 $\overline{\delta} = \delta_1, \ldots, \delta_p$, with free variables among $\overline{X}, x$, and such that (4.1) holds for all $i = 1, \ldots, p$. It thus follows that $\mathfrak{N} \models \forall \overline{X} \varphi[\overline{\delta[x]}/\overline{Y}]$. Then, by completeness (Thm. 2.12) we know that $\vdash \varphi[\overline{\delta}[x]/\overline{Y}]$ is provable in $\mathsf{MSO}$, and by negative translation (Thm. 3.4) we get $\mathsf{SMSO} \vdash \neg\neg\varphi[\overline{\delta}[x]/\overline{Y}]$. We can then apply ($p$ times) the synchronous comprehension scheme of $\mathsf{SMSO}$ and obtain $\mathsf{SMSO} \vdash \exists \overline{Y} \neg\neg\varphi(\overline{X}; \overline{Y})$. □

**Example 4.5.** Recall the specification (2.1) from [23], represented in $\mathsf{MSO}$ by the formula $\phi(X; Y)$ of Ex. 2.9. 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 $\phi_0$ and $\phi_1$ are monotonic in $Y$, while $\phi_2$ is anti-monotonic in $Y$. The formula $\phi_0$ is trivially realized by the identity function $\mathbf{2} \to_{\mathbf{M}} \mathbf{2}$ (see Ex. 2.3.(1)), which is itself represented by the deterministic uniformly bounded formula $\delta_0[X, x] := (x \mathrel{\dot{\in}} X)$. For $\phi_1$ (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 $\phi_2$ in Ex. 4.5 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 $\delta_1$ will output no 1's. On the other hand, the machine of Ex. 2.3.(3) involves an internal state, and can be represented using a fixpoint formula of the form (4.2). Reasoning on such formulae is easier with more advanced tools on $\mathsf{MSO}$, that we provide in §4.2.

4.2. **The Recursion Theorem.** Theorem 3.5.(2) ensures that $\mathsf{SMSO}$ 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 $\mathsf{MSO}$ [20] (see also [18]). The Recursion Theorem says that $\mathsf{MSO}$ allows to define predicates by well-founded induction w.r.t. the relation $\dot{<}$ defined as $(x \mathrel{\dot{<}} y) := (x \mathrel{\dot{\leq}} y \wedge \neg(x \doteq y))$. Given a formula $\psi$ and variables $X$ and $x$, we say that $\psi$ *is $x$-recursive in* $X$ when the following formula $\mathsf{Rec}^x_X(\psi)$ holds:

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

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

**Theorem 4.6** (Recursion Theorem [20])**.** *If* $\mathsf{MSO} \vdash \mathsf{Rec}^x_X(\psi)$ *then*

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

**Examples 4.7.**

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

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

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 4.6 implies that, provably in $\mathsf{MSO}$, for all $\overline{X}$ there are unique predicates $\overline{Q}, Y$ s.t. $\forall x.\theta[\overline{X}, \overline{Q}, Y, x]$. Indeed, assuming $\mathsf{I}$ and $\mathsf{H}$ are as in (4.3) 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

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

Now, apply Thm. 4.6 to $\mathsf{O}[\overline{Q}(t), \overline{X}(t)]$ (resp. $\tilde{\theta}_i$), which is $t$-recursive in $Y$ (resp. in $Q_i$).

(2) The machine of Ex. 2.3.(3) is represented as in (1) with $\mathsf{O}$ and $\mathsf{D}$ given by (4.4) (see Ex. 4.4, recalling that the machine as only two states). Hence $\mathsf{MSO}$ proves that for all $X$ there are unique $Q$, $Y$ such that $\forall x.\theta[X, Q, Y, x]$. Continuing now Ex. 4.5, 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. 4.6, 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.

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 derive

$$\exists^\infty t\ (\neg X t) \ \vdash_{\mathsf{MSO}}\ \exists^\infty t\ \exists Q, Y (\theta[X, Q, Y, t]\ \wedge\ \neg Y t)$$

One can proceed as follows. First note that if 0 is read from state 1, then the machine outputs 0. Moreover between two occurrences of 0 in the input with no intermediate input 1's, the only visited state of the machine is 1. Using induction, we thus have

$$\begin{array}{ll} & \theta[X, Q, Y, t']\,,\ t \dot{<} t'\,,\ \neg X t\,,\ \neg X t'\,,\ \neg\exists z (t \dot{<} z \dot{<} t'\ \wedge\ X(z))\ \vdash\ \neg Q(t') \\ \text{so that} & \theta[X, Q, Y, t']\,,\ t \dot{<} t'\,,\ \neg X t\,,\ \neg X t'\,,\ \neg\exists z (t \dot{<} z \dot{<} t'\ \wedge\ X(z))\ \vdash\ \neg Y(t') \end{array}$$

It follows that if there are infinitely many 0's in the input, then there are infinitely many 0's in the output. Indeed, using induction and the excluded middle, we have

$$\forall x. \theta[X, Q, Y, x]\,,\ \exists^\infty t\ (\neg X t)\ \vdash\ \exists t, t' \dot{\geq} n \left[\neg X t \wedge \neg X t' \wedge \neg\exists z (t \dot{<} x \dot{<} t'\ \wedge\ X(z))\right]$$

and therefore $\forall x. \theta[X, Q, Y, x]\,,\ \exists^\infty t\ (\neg X t)\ \vdash_{\mathsf{MSO}}\ \exists^\infty t\ (\neg Y t)$.

4.3. **From Bounded Formulae to Mealy Machines.** We now turn to a useful fact for part (1) of Thm. 3.5, 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. [22, §3.1]).

**Lemma 4.8.** *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 $\hat{\varphi}$.*

*Proof.* First, given an MSO-formula $\hat{\varphi}$ with free variables among $z, x_1, \dots, x_p, X_1, \dots, X_q$, if $\hat{\varphi}$ is bounded by $z$ then $\hat{\varphi}$ is of the form $\psi{\restriction}[- \dot{\leq} z]$, where the free variables of $\psi$ are among $z, x_1, \dots, x_p, X_1, \dots, X_q$. But note that $\psi{\restriction}[- \dot{\leq} z]$ is equivalent to the formula $(\exists t (\mathsf{last}[t] \wedge \psi[t/z])){\restriction}[- \dot{\leq} z]$, where $\mathsf{last}[t] := \forall x (t \dot{\leq} x \to t \doteq x)$ and where $t$ does not occur free in $\psi$. We can therefore assume that $\hat{\varphi}$ is of the form $\psi{\restriction}[- \dot{\leq} z]$ where $\psi$ has free variables among $x_1, \dots, x_p, X_1, \dots, X_q$.

Then, 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 $\mathfrak{N} \models \psi[\overline{k}/\overline{x}, \overline{A}/\overline{X}]{\restriction}[- \dot{\leq} n]$ if and only if the formula $\psi$ holds (in the sense of MSO over finite words) in the finite word $\langle \overline{k}, \overline{A}{\restriction}(n+1)\rangle$. Let $\mathcal{A} = (Q, q^\imath, \partial, F)$ be a DFA recognizing the language of finite words satisfying $\psi$ [22, Thm. 3.1]. Consider the Mealy machine $\mathcal{M} = (Q, q^\imath, \partial_{\mathcal{M}})$ with $\partial_{\mathcal{M}}(q, \mathsf{a}) = (q', b)$ where $q' = \partial(q, \mathsf{a})$ and ($b = 1$ iff $q' \in F$), and let $C : \mathbf{2}^p \times \mathbf{2}^q \longrightarrow_{\mathbf{M}} \mathbf{2}$ be the function induced by $\mathcal{M}$. We then have

$$\begin{array}{lll} & \langle \overline{k}, \overline{A}{\restriction}(n+1)\rangle \models \psi{\restriction}[- \dot{\leq} n] \text{ (in the sense of MSO over finite words)} & \\ \iff & \mathcal{A} \text{ accepts the finite word } \langle \overline{k}, \overline{A}{\restriction}(n+1)\rangle \quad \iff \quad C(\overline{k}, \overline{A})(n) = 1 & \end{array}$$

$\square$

**Remark 4.9.** Given $C : \mathbf{2}^p \times \mathbf{2}^q \to_{\mathbf{M}} \mathbf{2}$ $z$-represented by $\psi\lceil[- \dot{\leq} z]$ (with $z$ not free in $\psi$), 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 $\langle \overline{k}, \overline{A}\lceil(n+1)\rangle \models \psi$ (in the sense of $\mathsf{MSO}$ 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 $\langle \overline{k}, \overline{A}\lceil(n+1)\rangle$. Hence $\mathcal{M}$ must pay the price of the non-elementary lower-bound for translating $\mathsf{MSO}$-formulae over finite words to DFAs (see e.g. [8, Chap. 13]).

**Example 4.10.** Recall the continuous but not synchronous function $P$ of Ex. 2.3.(4). The function $P$ can be used to realize a predecessor function, and thus is represented (in the sense of (4.1)) 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 $\varphi$ is not equivalent to a bounded formula, since by Lem. 4.8 bounded formulae represent synchronous functions.

4.4. **Internally Bounded Formulae.** The synchronous comprehension scheme of $\mathsf{MSO}$ is motivated by Lem. 4.8, 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 $\psi$):

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

**Theorem 4.11.** *If* $\mathsf{MSO} \vdash \mathsf{B}^x_{\overline{X}}(\psi[\overline{X}, x])$ *and the free variables of* $\psi$ *are among* $x, \overline{X}$*, then there is a uniformly bounded formula* $\hat{\varphi}[\overline{X}, x]$ *which is effectively computable from* $\psi$ *and such that* $\mathsf{MSO} \vdash \forall \overline{X}\, \forall x\, (\psi[\overline{X}, x] \longleftrightarrow \hat{\varphi}[\overline{X}, x])$.

**Remark 4.12.** Theorem 4.11 relies on the decidability of $\mathsf{MSO}$. Note that Thm. 4.11 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 SMSO

This Section presents our realizability model for $\mathsf{SMSO}$, and uses it to prove Thm. 3.5.(1). Our approach to Church's synthesis *via* realizability uses automata in two different ways. First, from a *proof* $\mathscr{D}$ in $\mathsf{SMSO}$ 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 [19]). 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. 2.6), and thus to interpret the positive existentials of $\mathsf{SMSO}$ 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 $\varphi$ will exhibit the strictly positive existentials of $\varphi$ as $M(\mathcal{A}) = M(\varphi)$ where

$$M(\alpha) \simeq M(\neg\varphi) \simeq \mathbf{1} \qquad M(\varphi \wedge \psi) \simeq M(\varphi) \times M(\psi) \qquad M(\exists(-)\,\varphi) \simeq \mathbf{2} \times M(\varphi) \tag{5.1}$$

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

$$\mathcal{A} \quad = \quad (Q_{\mathcal{A}}\,,\, q^{\imath}_{\mathcal{A}}\,,\, M(\mathcal{A})\,,\, \partial_{\mathcal{A}}\,,\, \Omega_{\mathcal{A}}) \tag{5.2}$$

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 $\omega$-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 $\omega$-word $B \in \Sigma^{\omega}$ is an $\omega$-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 $\omega$-words accepted by $\mathcal{A}$.

Following the usual terminology, an automaton $\mathcal{A}$ as in (5.2) is *deterministic* if $M(\mathcal{A}) \simeq \mathbf{1}$.

Let us now sketch how uniform automata will be used in our realizability interpretation of $\mathsf{SMSO}$. First, by adapting to our context usual constructions on automata (§5.2), to each $\mathsf{MSO}$-formula $\varphi$ with free variables among (say) $\overline{X} = X_1, \dots, X_q$, we associate a uniform automaton $[\![\varphi]\!]$ over $\mathbf{2}^q$ (Fig. 7). Then, from an $\mathsf{SMSO}$-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 $\mathsf{Aut}_{\Sigma}$)**.** For each alphabet $\Sigma$, the category $\mathsf{Aut}_{\Sigma}$, 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 $R \Vdash \mathcal{A}(B)$.

The identity morphism $\mathcal{A} \Vdash \mathrm{Id}_{\mathcal{A}} : \mathcal{A}$ is given by $\mathrm{Id}_{\mathcal{A}}(B, R) := R$, and the composition of morphisms $\mathcal{A} \Vdash F : \mathcal{B}$ and $\mathcal{B} \Vdash G : \mathcal{C}$ is the morphism $\mathcal{A} \Vdash G \circ F : \mathcal{C}$ given by $(G \circ F)(B, R) := G(B, F(B, R))$. It is easy to check the usual identity and composition laws of categories, namely:

$$\mathrm{Id} \circ F = F \qquad F \circ \mathrm{Id} = F \qquad (F \circ G) \circ H = F \circ (G \circ H)$$

**Remarks 5.3.**

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

*Proof.* Assume $\mathcal{B} \Vdash F : \mathcal{A}$ and $B \in \mathcal{L}(\mathcal{B})$ so that $R \Vdash \lfloor(B)$ for some $R \in M(\mathcal{B})^{\omega}$. Then by definition of $\mathcal{B} \Vdash F : \mathcal{A}$, we have $F(B, R) \Vdash \mathcal{A}(B)$, so that $B \in \mathcal{L}(\mathcal{A})$. $\square$

(2) One could also consider the category $\mathsf{AUT}_\Sigma$ defined as $\mathsf{Aut}_\Sigma$, but with maps not required to be finite-state. All statements of this Section hold for $\mathsf{AUT}_\Sigma$, but for Cor. 5.11, which would lead to non necessarily finite-state realizers and would not give Thm. 3.5.(1).
(3) Uniform automata are a variation of usual automata on $\omega$-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. [22]), we lose nothing by assuming their acceptance condition to be given by arbitrary $\omega$-regular sets.

5.2. **Constructions on Automata.** We gather here constructions on uniform automata that we will need to interpret $\mathsf{MSO}$ formulae. First, automata are closed under the following operation of *finite substitution.*

**Proposition 5.4.** *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 5.5.** Assume $\mathcal{A}$ interprets a formula $\varphi$ 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 $\varphi$ is also a formula with free variables among $\overline{X}, \overline{Y}$, and we have $\overline{BB'} \in \mathcal{L}(\mathcal{A}[\pi])$ 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 $\mathsf{Aut}_\Sigma$. This gives the interpretation of conjunctions.

**Proposition 5.6.** *For each $\Sigma$, the category $\mathsf{Aut}_\Sigma$ 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 $\Omega$ is $\omega$-regular since $\Omega_{\mathcal{A}}$ and $\Omega_{\mathcal{B}}$ are $\omega$-regular (see e.g.* [15, Ex. I.11.3.7]*). Moreover, $\mathcal{L}(\mathbf{I}) = \Sigma^\omega$ and $\mathcal{L}(\mathcal{A} \times \mathcal{B}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B})$.*

*Proof.* The Cartesian structure is directly inherited from $\mathbf{M}$ and is omitted. Moreover, we obviously have $\mathcal{L}(\mathbf{I}) = \Sigma^\omega$. Let us show that $\mathcal{L}(\mathcal{A}_1 \times \mathcal{A}_2) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$. The inclusion $(\subseteq)$ follows from Rem. 5.3.(1) applied to the projection maps $\mathcal{A}_1 \times \mathcal{A}_2 \Vdash \varpi_i : \mathcal{A}_i$ induced by the Cartesian structure. For the converse inclusion $(\supseteq)$, note that if $R_i \Vdash \mathcal{A}_i(B)$ for $i = 1, 2$, then $\langle R_1, R_2 \rangle \Vdash (\mathcal{A}_1 \times \mathcal{A}_2)(B)$. $\square$

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* $\Sigma$ 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 5.7.** *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*

$$C \ : \ \Sigma \times M(\mathcal{B}) \ \longrightarrow_{\mathbf{M}} \ \Gamma \qquad\qquad G \ : \ \Sigma \times M(\mathcal{B}) \ \longrightarrow_{\mathbf{M}} \ M(\mathcal{A})$$

*such that $G(B, R) \Vdash \mathcal{A}\langle B, C(B, R) \rangle$ for all $B \in A^\omega$ and all $R \Vdash \mathcal{B}(B)$.*

*Proof.* Consider a realizer $\mathcal{B} \Vdash F : \exists_\Gamma \mathcal{A}$, for some $\mathcal{B} : \Sigma$ with set of moves $V$. Then $F$ is a synchronous function from $\Sigma^\omega \times V^\omega$ to $(\Gamma \times U)^\omega \simeq \Gamma^\omega \times U^\omega$, and is therefore given by a pair $\langle C, G \rangle$ of synchronous functions

$$C \quad : \quad \Sigma \times V \quad \longrightarrow_{\mathbf{M}} \quad \Gamma \qquad\qquad G \quad : \quad \Sigma \times V \quad \longrightarrow_{\mathbf{M}} \quad U \tag{5.3}$$

Moreover, given $B \in \Sigma^\omega$ and $R \Vdash \mathcal{B}(B)$, since $F(B, R) \Vdash \exists_\Gamma \mathcal{A}(B)$, it is easy to see that $G(B, R) \Vdash \mathcal{A}(\langle B, C(B, R)\rangle)$. Conversely, given $C$ and $G$ as in (5.3), if $G(B, R) \Vdash \mathcal{A}(\langle B, C(B, R)\rangle)$ for all $B \in A^\omega$ and all $R \Vdash \mathcal{B}(B)$, then we have $\mathcal{B} \Vdash \langle C, G\rangle : \exists_\Gamma \mathcal{A}$. $\square$

The negation $\neg(-)$ of $\mathsf{SMSO}$ is interpreted by an operation $\sim(-)$ on uniform automata which involves McNaughton's Theorem 2.10.

**Proposition 5.8.** *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 $B \notin \mathcal{L}(\mathcal{A})$.*

*Proof.* Let $U := M(\mathcal{A})$ and consider the (usual) deterministic automaton over $\Sigma \times U$ with the same states as $\mathcal{A}$ and with transition function $\partial_\mathcal{S}$ defined as $\partial_\mathcal{S}(q, (\mathsf{a}, u)) := \partial_\mathcal{A}(q, \mathsf{a}, u)$. Then $R \Vdash \mathcal{A}(B)$ iff $\mathcal{S}$ accepts $\langle B, R\rangle$. Since $\Omega_\mathcal{A}$ is $\omega$-regular, it is recognized by a non-det. Büchi automaton $\mathcal{C}$ over $Q_\mathcal{A}$. We then obtain a non-det. Büchi automaton $\mathcal{B}$ over $\Sigma \times U$ with state set $Q_\mathcal{A} \times Q_\mathcal{C}$ and s.t. $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{S})$. It follows that $B \in \mathcal{L}(\mathcal{A})$ iff $B \in \mathcal{L}(\tilde{\exists}_U \mathcal{B})$, where $\tilde{\exists}_U \mathcal{B}$ is the usual projection of $\mathcal{B}$ on $\Sigma$. By McNaughton's Theorem (Thm. 2.10), $\tilde{\exists}_U \mathcal{B}$ is equivalent to a *deterministic* Muller automaton $\mathcal{D}$ over $\Sigma$. Then we let $\sim\mathcal{A}$ be the deterministic uniform automaton defined as $\mathcal{D}$ but with $\Omega_{\sim\mathcal{A}}$ the $\omega$-regular set generated by the Muller condition $S \in \mathcal{T}$ iff $S \notin \mathcal{T}_\mathcal{D}$ (see e.g. [15, Thm. I.7.1 & Prop. I.7.4]). $\square$

5.3. **The Realizability Interpretation.** We are now going to define our realizability interpretation. This goes in two steps:

(1) To each formula $\varphi$ we associate a uniform automaton $[\![\varphi]\!]$.
(2) To each derivation $\mathscr{D}$ of a sequent $\varphi_1, \ldots, \varphi_n \vdash \varphi$ in $\mathsf{SMSO}$, we associate a finite state synchronous $F_\mathscr{D}$ such that $[\![\varphi_1]\!] \times \cdots \times [\![\varphi_n]\!] \Vdash F_\mathscr{D} : [\![\varphi]\!]$. Theorem 3.5.(1) (stated as Corollary 5.11 below) will then directly follow from the definition of $[\![\exists X\, \varphi]\!]$.

We first discuss (1). Consider a formula $\varphi$ with free variables among $\overline{x} = x_1, \ldots, x_p$ and $\overline{X} = X_1, \ldots, X_q$. Its interpretation will be a uniform automaton $[\![\varphi]\!]_{\overline{x},\overline{X}}$ over $\mathbf{2}^p \times \mathbf{2}^q$, defined by induction on $\varphi$, and such that $[\![\delta]\!]_{\overline{x},\overline{X}}$ is deterministic for a deterministic $\delta$. We thus have to devise a deterministic uniform automaton $\mathcal{A}_\alpha$ for each atomic formula $\alpha$ of $\mathsf{SMSO}$. The definitions of the $\mathcal{A}_\alpha$'s are easy and follow usual constructions (see e.g. [22]). They are deferred to Appendix C. Moreover, in order to handle individual variables, the interpretation also uses a deterministic uniform automaton $\mathsf{Sing} : \mathbf{2}$ accepting the $B \in \mathbf{2}^\omega \simeq \mathcal{P}(\mathbb{N})$ such that $B$ is a singleton. Appendix C also presents a possible definition for $\mathsf{Sing}$. The interpretation $[\![\varphi]\!]_{\overline{x},\overline{X}}$ is defined on Figure 7, where $\pi$, $\pi'$ 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 (5.1), so in particular $[\![\delta]\!]$ is indeed deterministic for a deterministic $\delta$. Moreover, as expected we get:

**Proposition 5.9.** *Given an $\mathsf{MSO}$-formula $\varphi$ 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 $\mathfrak{N} \models \varphi[\overline{k}/\overline{x}, \overline{B}/\overline{X}]$.*

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

Figure 7: Interpretation of $\mathsf{MSO}$-Formulae as Uniform Automata.

We now turn to (2). Let $\varphi_1, \dots, \varphi_n, \varphi$ be $\mathsf{MSO}$-formulae with free variables among $\overline{x} = x_1, \dots, x_p$ and $\overline{X} = X_1, \dots, X_q$. 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 $\varphi_1, \dots, \varphi_n \vdash \varphi$ (notation $\varphi_1, \dots, \varphi_n \Vdash F : \varphi$ or $\overline{\varphi} \Vdash F : \varphi$) if

$$\llbracket \varphi_1 \rrbracket_{\overline{x},\overline{X}} \times \cdots \times \llbracket \varphi_n \rrbracket_{\overline{x},\overline{X}} \quad \Vdash \quad F \quad : \quad \llbracket \varphi \rrbracket_{\overline{x},\overline{X}}$$

**Theorem 5.10** (Adequacy)**.** *Let $\overline{\varphi}, \varphi$ be $\mathsf{MSO}$-formulae with variables among $\overline{x}, \overline{X}$. From an $\mathsf{SMSO}$-derivation $\mathscr{D}$ of $\overline{\varphi} \vdash \varphi$, one can compute an $\mathbf{M}$-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 $\delta$ deterministic, it follows from Prop. 5.9 and (5.1) that the unique $\mathbf{M}$-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 $\perp$), but in this case the realizer of $\overline{\varphi} \vdash \varphi$ is not canonical, and elimination of equality is direct from Prop. 5.9. Adequacy for synchronous comprehension is deferred to §5.4. As for the rules of Fig. 5, the first two rules follow from the fact that $\mathbf{M}$ is a category with finite limits (Prop. 2.6), and the rules for conjunction (resp. existentials) follow from Prop. 5.6 (resp. Prop. 5.7). 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.(2)) together with the fact that $\mathsf{S}(-,-)$ is deterministic. The case of the former is similar and simpler. $\square$

Adequacy of realizability, together with Proposition 5.7, directly gives Theorem 3.5.(1).

**Corollary 5.11** (Thm. 3.5.(1))**.** *Given a derivation $\mathscr{D}$ in $\mathsf{SMSO}$ 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 $\overline{B} \in (\mathbf{2}^\omega)^q \simeq (\mathbf{2}^q)^\omega$.*

5.4. **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. 4.8) 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 5.12.** *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 $\mathsf{MSO}$-formula $\psi$ 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*

$$R \Vdash \llbracket \psi[\hat{\varphi}[y]/Y] \rrbracket_{\overline{x},\overline{X}}(\overline{k}, \overline{A}) \qquad \Longrightarrow \qquad H_\psi(R) \Vdash \llbracket \psi \rrbracket_{\overline{x},\overline{X},Y}(\overline{k}, \overline{A}, C_{\hat{\varphi}}(\overline{A})) \tag{5.4}$$

*Proof.* By induction on $\psi$, we show (5.4) and that for all $\overline{k} \in (\mathbf{2}^\omega)^p$, and all $\overline{A} \in (\mathbf{2}^\omega)^q$, we have

$$(\overline{k}, \overline{A}) \in \mathcal{L}(\llbracket \psi[\hat{\varphi}[y]/Y] \rrbracket) \qquad \Longleftrightarrow \qquad (\overline{k}, \overline{A}, C_{\hat{\varphi}}(\overline{A})) \in \mathcal{L}(\llbracket \psi \rrbracket) \tag{5.5}$$

- If $\psi$ is an atomic formula not of the form $(x \mathrel{\dot{\in}} Y)$, then $\psi[\hat{\varphi}[y]/Y] = \psi$. So we take the identity for $H_\psi$ and (5.4) and (5.5) are obvious.
- If $\psi$ is of the form $(x_i \mathrel{\dot{\in}} Y)$, then $\psi[\hat{\varphi}[y]/Y] = \hat{\varphi}[x_i/y]$. Since $\psi$ is deterministic, we can take for $H_\psi$ the unique map $M(\hat{\varphi}) \to_{\mathbf{M}} M(x_i \mathrel{\dot{\in}} Y) = \mathbf{1}$. Then by (4.1), for all $\overline{k} \in (\mathbf{2}^\omega)^p$, and all $\overline{A} \in (\mathbf{2}^\omega)^q$, we have
$$C(\overline{A})(k_i) = 1 \qquad \text{iff} \qquad \mathfrak{N} \models \hat{\varphi}[k_i/z, \overline{A}/\overline{X}]$$
that is
$$\mathfrak{N} \models k_i \mathrel{\dot{\in}} C(\overline{A}) \qquad \text{iff} \qquad \mathfrak{N} \models \hat{\varphi}[k_i/z, \overline{A}/\overline{X}]$$
and it then follows from Prop. 5.9 that
$$(\overline{k}, \overline{A}, C(\overline{A})) \in \mathcal{L}(\llbracket x_i \mathrel{\dot{\in}} Y \rrbracket_{\overline{x}, \overline{X}, Y}) \qquad \text{iff} \qquad (\overline{k}, \overline{A}) \in \mathcal{L}(\llbracket \hat{\varphi} \rrbracket_{\overline{x}, \overline{X}})$$
from which we also get (5.4).
- If $\psi$ is of the form $\varphi_1 \wedge \varphi_2$ (resp. $\exists X\, \varphi$, $\exists x\, \varphi$) then we conclude by induction hypothesis and Prop. 5.6 (resp. Prop. 5.7).
- If $\psi = \neg\varphi$, then we have $M(\psi) = M(\psi[\hat{\varphi}[y]/Y]) = \mathbf{1}$, and $H_\psi$ is the identity. Moreover, property (5.5) follows from the induction hypothesis and Prop. 5.8, and we thus also get (5.4). $\square$

Adequacy of synchronous comprehension then directly follows.

**Lemma 5.13.** *Let $\psi$ 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 $\psi$ and $\varphi$.*

*Proof.* Let $C_{\hat{\varphi}}$ satisfying (4.1) be given by Lem. 4.8, and let $H_\psi$ satisfying (5.4) be given by Lem. 5.12. It then directly follows from Prop. 5.7 and Len. 5.12 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 $\pi, \pi'$ are suitable projections. $\square$

## 6. Indexed Structure on Automata

In §5 we have defined one category $\mathsf{Aut}_\Sigma$ for each alphabet $\Sigma$. These categories are actually related by *substitution functors* arising from **M**-morphisms, inducing a *fibred structure*. Substitution functors are a basic notion of categorical logic, which allows for categorical axiomatizations of quantifications. We refer to e.g. [9, Chap. 1] for background.

This Section presents the fibred structure of the categories $\mathsf{Aut}_{(-)}$ and shows that the existential quantifiers $\exists_{(-)}$ and Cartesian product $- \times -$ of §5.2 satisfy the expected properties of existential quantifiers and conjunction in categorical logic. These properties essentially correspond to the adequacy of the logical rules of Figure 5 that do not mention negation $\neg$ or $\bot$. Although the fibred structure is not technically necessary to prove the adequacy of our realizability model, following such categorical axiomatization was a guideline in its design. Besides, categorical logic turns out to be an essential tool when dealing with generalizations to (say) alternating automata.

6.1. **The Basic Idea.** Before entering the details, let us try to explain the main ideas in the usual setting of first-order logic over a multisorted individual language. The categorical semantics of existential quantifications is given by an adjunction which is usually represented as

$$\frac{\exists x\, \varphi(x) \vdash \psi}{\varphi(x) \vdash \psi} \qquad (x \text{ not free in } \psi) \tag{6.1}$$

This adjunction induces a bijection between (the interpretations of) proofs of the sequents $\varphi(x) \vdash \psi$ and $\exists x.\varphi(x) \vdash \psi$, that we informally denote

$$\varphi(x) \vdash \psi \qquad \simeq \qquad \exists x\, \varphi(x) \vdash \psi$$

Now, in general the variable $x$ will occur free in $\varphi$. As a consequence, in order to properly formulate (6.1) one should be able to interpret sequents of the form $\varphi(x) \vdash \psi$ with free variables. More generally, the formulae $\varphi$ and $\psi$ should be allowed to contain free variables distinct from $x$.

The idea underlying the general method (but see e.g. [9] for details), is to first devise a base category $\mathbb{B}$ of individuals, whose objects interpret products of sorts of the individual language, and whose maps from say $\iota_1 \times \cdots \times \iota_m$ to $o_1 \times \cdots \times o_n$ represent $n$-tuples $(t_1, \dots, t_n)$ of terms $t_i$ of sort $o_i$ whose free variables are among $x_{\iota_1}, \dots, x_{\iota_m}$ (with $x_{\iota_j}$ of sort $\iota_j$). Then, for each object $\iota = \iota_1 \times \cdots \times \iota_n$ of $\mathbb{B}$, one devises a category $\mathbb{E}_\iota$ whose objects represent formulae with free variables among $x_{\iota_1}, \dots, x_{\iota_n}$, and whose morphisms interpret proofs. Furthermore, $\mathbb{B}$-morphisms

$$t = (t_1, \dots, t_n) \quad : \quad \iota_1 \times \cdots \times \iota_m \quad \longrightarrow \quad o_1 \times \cdots \times o_n$$

induce *substitution functors*

$$t^\star \quad : \quad \mathbb{E}_{o_1 \times \cdots \times o_n} \quad \longrightarrow \quad \mathbb{E}_{\iota_1 \times \cdots \times \iota_m}$$

The functor $t^\star$ takes (the interpretation of) a formula $\varphi$ with free variables among $y_{o_1}, \dots, y_{o_n}$ to (the interpretation of) the formula $\varphi[t_1/y_{o_1}, \dots, t_n/y_{o_n}]$ with free variables among $x_{\iota_1}, \dots, x_{\iota_m}$. Its action on the morphisms of $\mathbb{E}_{o_1 \times \cdots \times o_n}$ allows to interpret the *substitution rule*

$$\frac{\varphi \vdash \psi}{\varphi[t_1/y_{o_1}, \dots, t_n/y_{o_n}] \vdash \psi[t_1/y_{o_1}, \dots, t_n/y_{o_n}]}$$

In very good situations, the operation $(-)^\star$ is itself functorial. Among the morphisms of $\mathbb{B}$, one usually requires the existence of projections, say

$$\pi \quad : \quad o \times \iota \quad \longrightarrow \quad o$$

Projections induce substitution functors, called *weakening* functors

$$\pi^\star \quad : \quad \mathbb{E}_o \quad \longrightarrow \quad \mathbb{E}_{o \times \iota}$$

which simply allow to see formula $\psi(y_o)$ with free variable $y_o$ as a formula $\psi(y_o, x_\iota)$ with free variables among $y_o, x_\iota$ (but with no actual occurrence of $x_\iota$). Then the proper formulation of (6.1) is that existential quantification over $x_\iota$ is a functor

$$\exists x_\iota\, (-) \quad : \quad \mathbb{E}_{o \times \iota} \quad \longrightarrow \quad \mathbb{E}_o$$

which is left-adjoint to $\pi^\star$:

$$\frac{\exists x_\iota\, \varphi(x_\iota, y_o) \vdash \psi(y_o)}{\varphi(x_\iota, y_o) \vdash \pi^\star(\psi)(x_\iota, y_o)} \tag{6.2}$$

(where $x_\iota$ does not occur free in $\psi$ since $\psi$ is assumed to be (interpreted as) an object of $\mathbb{E}_o$, thus replacing the usual side condition). Universal quantifications are dually axiomatized as right adjoints to weakening functors. In both cases, the adjunctions are subject to additional conditions (called the *Beck-Chevalley* conditions) which ensure that they are preserved by substitution and the soundness of their elimination rules.

6.2. **Substitution.** So far, for each alphabet $\Sigma$ we have defined a category $\mathsf{Aut}_\Sigma$ of Uniform Automata over $\Sigma$. Different categories $\mathsf{Aut}_\Sigma$, $\mathsf{Aut}_\Gamma$ can be related by means of Mealy-morphisms $F : \Sigma \to \Gamma$. This relies on a very simple *substitution operation* on automata, generalizing the substitution operation presented in Proposition 5.4.

**Definition 6.1** (Substitution)**.** Given a Mealy machine $\mathcal{M} : \Sigma \to \Gamma$ as in Definition 2.1 and an automaton $\mathcal{A} : \Gamma$ as in Definition 5.1, the automaton $\mathcal{A}[\mathcal{M}] : \Sigma$ is defined as follows:

$$\mathcal{A}[\mathcal{M}] \quad := \quad (Q_\mathcal{A} \times Q_\mathcal{M},\, (q^\imath_\mathcal{A}, q^\imath_\mathcal{M}),\, M(\mathcal{A}),\, \partial_{\mathcal{A}[\mathcal{M}]},\, \Omega_{\mathcal{A}[\mathcal{M}]})$$

where

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

is defined as

$$\partial_{\mathcal{A}[\mathcal{M}]}((q_\mathcal{A}, q_\mathcal{M}), \mathtt{a}, m) \quad := \quad (\partial_\mathcal{A}(q_\mathcal{A}, \mathtt{b}, m),\, q'_\mathcal{M}) \qquad \text{with} \qquad (q'_\mathcal{M}, \mathtt{b}) \quad := \quad \partial_\mathcal{M}(q_\mathcal{M}, \mathtt{a})$$

and where $(q_k, q'_k)_{k\in\mathbb{N}} \in \Omega_{\mathcal{A}[\mathcal{M}]}$ iff $(q_k)_{k\in\mathbb{N}} \in \Omega_\mathcal{A}$.

Note the reversed direction of the action of $\mathcal{M} : \Sigma \to \Gamma$: the substitution operation $(-)[\mathcal{M}]$ takes an automaton over $\Gamma$ to an automaton over $\Sigma$. Substitutions of the form $\mathcal{A}[\mathcal{M}]$ can be seen as generalizations of the substitutions presented in Proposition 5.4: Given a function $\mathtt{f} : \Sigma \to \Gamma$, the automaton $\mathcal{A}[\mathtt{f}]$ of Proposition 5.4 is isomorphic (in $\mathsf{Aut}_\Sigma$) to the automaton $\mathcal{A}[\mathcal{M}_\mathtt{f}]$ obtained by applying Definition 6.1 to the one-state Mealy machine inducing the $\mathbf{M}$-morphism $[\mathtt{f}] : \Sigma \to_\mathbf{M} \Gamma$ of Remark 2.5.

We now characterize the language of $\mathcal{A}[\mathcal{M}]$. To this end, it is useful to note that $\Sigma^\omega$ is in bijection with the set of synchronous functions $\mathbf{1}^\omega \to \Sigma^\omega$.

**Proposition 6.2.** *Given a Mealy machine $\mathcal{M} : \Sigma \to \Gamma$ and an automaton $\mathcal{A} : \Gamma$, for $B \in \Sigma^\omega$ we have:*

$$B \in \mathcal{L}(\mathcal{A}[\mathcal{M}]) \qquad \textit{iff} \qquad F_\mathcal{M} \circ B \in \mathcal{L}(\mathcal{A})$$

*where $F_\mathcal{M} \circ B$ is the composition of the synchronous function $F_\mathcal{M}$ induced by $\mathcal{M}$ with $B$ seen as a synchronous function $\mathbf{1}^\omega \to \Sigma^\omega$.*

Given $\mathcal{M} : \Sigma \to \Gamma$, an important property of the substitution operation $(-)[\mathcal{M}]$ is that it induces a functor $\mathsf{Aut}_\Gamma \to \mathsf{Aut}_\Sigma$. The action of this functor on objects of $\mathsf{Aut}_\Gamma$ has just been defined. Given a morphism $\mathcal{A} \Vdash F : \mathcal{B}$ of $\mathsf{Aut}_\Gamma$, the morphism $\mathcal{A}[\mathcal{M}] \Vdash F[\mathcal{M}] : \mathcal{B}[\mathcal{M}]$ is the finite-state synchronous function

$$F[\mathcal{M}] \quad : \quad \Sigma \times M(\mathcal{A}) \quad \longrightarrow \quad M(\mathcal{B})$$

taking $(B, R)$ to $F(F_\mathcal{M}(B), R)$, where $F_\mathcal{M}$ is the finite-state synchronous function induced by $\mathcal{M}$. It is easy to see that the action of $(-)[\mathcal{M}]$ on morphisms preserves identities and composition.

6.3. **Categorical Existential Quantifications.** Recall from §5.2 that uniform automata are equipped with existential quantifications, given by an adaption of the usual projection operation on non-deterministic automata. Given $\mathcal{A} : \Sigma \times \Gamma$, we defined $\exists_\Gamma \mathcal{A} : \Sigma$ as

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

We are now going to see that $\exists_{(-)}$ is an existential quantification in the usual categorical sense of *simple coproduct* [9, Def. 1.9.1]. First, the *weakening functors*

$$(-)[\pi] \quad : \quad \mathsf{Aut}_\Sigma \quad \longrightarrow \quad \mathsf{Aut}_{\Sigma \times \Gamma}$$

alluded to in §6.1 are the substitution functors induced by projections (see also Ex. 5.5):

$$[\pi] \quad : \quad \Sigma \times \Gamma \quad \longrightarrow_{\mathbf{M}} \quad \Sigma$$

We can now state the first required property, namely that $\exists_\Gamma$ induces a functor left adjoint to $(-)[\pi]$.

**Proposition 6.3.** *Each existential quantifier $\exists_\Gamma$ induces a functor $\mathsf{Aut}_{\Sigma \times \Gamma} \to \mathsf{Aut}_\Sigma$ which is left-adjoint to the weakening functor $(-)[\pi] : \mathsf{Aut}_\Sigma \to \mathsf{Aut}_{\Sigma \times \Gamma}$.*

*Proof.* Fix alphabets $\Sigma$ and $\Gamma$. According to [13, Thm. IV.1.2.(ii)], we have to show that for each automaton $\mathcal{A} : \Sigma \times \Gamma$, there is an $\mathsf{Aut}_{\Sigma \times \Gamma}$-morphism

$$\eta_\mathcal{A} \quad : \quad \mathcal{A} \quad \longrightarrow \quad (\exists_\Gamma \mathcal{A})[\pi]$$

satisfying the following universal property: for each automaton $\mathcal{B} : \Sigma$ and each $\mathsf{Aut}_{\Sigma \times \Gamma}$-morphism

$$F \quad : \quad \mathcal{A} \quad \longrightarrow \quad \mathcal{B}[\pi]$$

there is a unique $\mathsf{Aut}_\Sigma$-morphism

$$H \quad : \quad \exists_\Gamma \mathcal{A} \quad \longrightarrow \quad \mathcal{B}$$

such that we have

$$\begin{tikzcd} \mathcal{A} \arrow[r, "\eta_\mathcal{A}"] \arrow[d, "F"'] & (\exists_\Gamma \mathcal{A})[\pi] \arrow[dl, "H[\pi]"] \\ \mathcal{B}[\pi] & \end{tikzcd} \tag{6.3}$$

Note that $\eta_\mathcal{A}$ must be a $\mathbf{M}$-morphism

$$\eta_\mathcal{A} \quad : \quad (\Sigma \times \Gamma) \times M(\mathcal{A}) \quad \longrightarrow \quad \Gamma \times M(\mathcal{A})$$

So we let $\eta_\mathcal{A}$ be the $\mathbf{M}$-morphism induced by the projection $\Sigma \times \Gamma \times M(\mathcal{A}) \to \Gamma \times M(\mathcal{A})$. Given $\mathcal{A} \Vdash F : \mathcal{B}[\pi]$, we are left with the following trivial fact: there is a unique $\exists_\Gamma \mathcal{A} \Vdash H : \mathcal{B}$ such that

$$\forall \langle B, C \rangle \in (\Sigma \times \Gamma)^\omega,\ \forall R \in M(\mathcal{A})^\omega, \quad F(\langle B, C \rangle, R) \quad = \quad H(B, \langle C, R \rangle)$$

$\square$

The Beck-Chevalley condition of [9, Def. 1.9.1] amounts to the following isomorphism in $\mathsf{Aut}_\Delta$, where $\mathcal{A} : \Sigma \times \Gamma$ and $F : \Delta \to_{\mathbf{M}} \Sigma$:

$$(\exists_\Gamma \mathcal{A})[\mathcal{M}_F] \quad \simeq \quad \exists_\Gamma(\mathcal{A}[\mathcal{M}_{F \times \mathrm{id}_\Gamma}])$$

This isomorphism follows from the fact that the two above automata have the same set of moves (namely $\Gamma \times M(\mathcal{A})$).

6.4. **Categorical Conjunction.** Recall from §5.2 that each category $\mathsf{Aut}_\Sigma$ has Cartesian products, which interpret conjunction, a necessary feature to interpret a sequent as a morphism from the conjunct of its premises to its conclusions. In the setting of categorical logic, it remains to be shown that these products are *fibred* in the sense of [9, Def. 1.8.1], i.e., they are preserved by substitution.

**Proposition 6.4.** *Given automata $\mathcal{A}, \mathcal{B} : \Gamma$ and a Mealy machine $\mathcal{M} : \Sigma \to \Gamma$, the product $\mathcal{A}[\mathcal{M}] \times \mathcal{B}[\mathcal{M}]$ is isomorphic to $(\mathcal{A} \times \mathcal{B})[\mathcal{M}]$ in $\mathsf{Aut}_\Sigma$.*

*Proof.* The isomorphism easily follows from the fact that

$$M(\mathcal{A}[\mathcal{M}] \times \mathcal{B}[\mathcal{M}]) \quad = \quad M(\mathcal{A}) \times M(\mathcal{B}) \quad = \quad M((\mathcal{A} \times \mathcal{B})[\mathcal{M}])$$

□

6.5. **Indexed Structure.** The substitution operation discussed in §6.2 allows for each $\mathbf{M}$-morphism $F : \Sigma \to \Gamma$ to induce a functor $(-)[\mathcal{M}_F] : \mathsf{Aut}_\Gamma \to \mathsf{Aut}_\Sigma$, where $\mathcal{M}_F$ is a *chosen* Mealy machine inducing $F$. As usual in categorical logic, we would like to extend substitution to a functor $(-)^\star : \mathbf{M}^{\mathrm{op}} \to \mathbf{Cat}$ taking alphabets $\Sigma$ to categories $\mathsf{Aut}_\Sigma$, and $\mathbf{M}$-morphisms $F : \Sigma \to \Gamma$ to functors $\mathsf{Aut}_\Gamma \to \mathsf{Aut}_\Sigma$. In order for $(-)^\star$ to be a functor, it should preserve identities and composition. In particular, given an automaton $\mathcal{A} : \Sigma$, for all $\mathbf{M}$-maps $G : \Delta \to \Gamma$ and $F : \Gamma \to \Sigma$ we should have

$$\mathcal{A} \quad = \quad \mathcal{A}[\mathcal{M}_{\mathrm{Id}_\Sigma}] \qquad \text{and} \qquad (\mathcal{A}[\mathcal{M}_F])[\mathcal{M}_G] \quad = \quad \mathcal{A}[\mathcal{M}_{F \circ G}] \tag{6.4}$$

But we see no reason for this to be possible. In particular there is no reason for the Mealy machine $\mathcal{M}_{F \circ G}$ chosen to induce $F \circ G$ to be a product of $\mathcal{M}_F$ and $\mathcal{M}_G$. However, since $\mathcal{A}[\mathcal{M}]$ always has the same moves as $\mathcal{A}$, we actually get (6.4) modulo isomorphisms.

This is a usual situation in categorical logic. It is indeed customary to relax the requirement of $(-)^\star$ to be a functor, and only ask it to be a *pseudo* functor, *i.e.* a functor for which identities and composition are only preserved up to natural isomorphisms, subject to some specific coherence conditions (see e.g. [9, Def. 1.4.4]). The required natural isomorphisms have the form

$$\begin{array}{rcccl} \eta_\Sigma & : & \mathrm{Id}_{\mathsf{Aut}_\Sigma} & \xrightarrow{\simeq} & (-)[\mathcal{M}_{\mathrm{Id}_\Sigma}] \\ \mu_{G,F} & : & (-)[\mathcal{M}_F][\mathcal{M}_G] & \xrightarrow{\simeq} & (-)[\mathcal{M}_{F \circ G}] \end{array} \tag{6.5}$$

Since $\mathcal{A}$ and $\mathcal{A}[\mathcal{M}]$ have the same moves, we can take for each components of $\eta_\Sigma$ and $\mu_{F,G}$ synchronous functions acting as identities on runs. It then follows that all the required diagrams commute.

We now proceed to the formal construction. Fix for each $\mathbf{M}$-morphism $F : \Sigma \to \Gamma$ a chosen Mealy machine $\mathcal{M}_F$ inducing $F$. For each $\mathcal{A} : \Sigma$, and each $\mathbf{M}$-morphisms $G : \Delta \to \Gamma$ and $F : \Gamma \to \Sigma$, we let

$$\mathcal{A} \Vdash \eta_{\Sigma,\mathcal{A}} : \mathcal{A}[\mathcal{M}_{\mathrm{Id}_\Sigma}] \qquad \text{and} \qquad \mathcal{A}[\mathcal{M}_F][\mathcal{M}_G] \Vdash \mu_{G,F,\mathcal{A}} : \mathcal{A}[\mathcal{M}_{F \circ G}]$$

$$\begin{tikzcd}
 & \mathcal{A}[\mathcal{M}_F] \arrow[dl, "\eta_{\Gamma,\mathcal{A}[\mathcal{M}_F]}"'] \arrow[d, equal] \arrow[dr, "\eta_{\Sigma,\mathcal{A}}[\mathcal{M}_F]"] & \\
\mathcal{A}[\mathcal{M}_F][\mathcal{M}_{\mathrm{Id}_\Gamma}] \arrow[r, "\mu_{\mathrm{Id}_\Gamma,F,\mathcal{A}}"'] & \mathcal{A}[\mathcal{M}_F] & \mathcal{A}[\mathcal{M}_{\mathrm{Id}_\Sigma}][\mathcal{M}_F] \arrow[l, "\mu_{F,\mathrm{Id}_\Sigma,\mathcal{A}}"]
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{A}[\mathcal{M}_F][\mathcal{M}_G][\mathcal{M}_H] \arrow[r, "\mu_{G,F,\mathcal{A}}[\mathcal{M}_H]"] \arrow[d, "\mu_{H,G,\mathcal{A}[\mathcal{M}_F]}"'] & \mathcal{A}[\mathcal{M}_{F\circ G}][\mathcal{M}_H] \arrow[d, "\mu_{H,F\circ G,\mathcal{A}}"] \\
\mathcal{A}[\mathcal{M}_F][\mathcal{M}_{G\circ H}] \arrow[r, "\mu_{G\circ H,F,\mathcal{A}}"'] & \mathcal{A}[F\circ G\circ H]
\end{tikzcd}$$

Figure 8: Coherence Diagrams for the Structure Maps of $(-)^\star : \mathbf{M}^{\mathrm{op}} \to \mathbf{Cat}$.

be given by

$$\eta_{\Sigma,\mathcal{A}} \;:\; \begin{array}{ccc} \Sigma \times M(\mathcal{A}) & \longrightarrow & M(\mathcal{A}) \\ (B,R) & \longmapsto & R \end{array} \qquad \text{and} \qquad \mu_{G,F,\mathcal{A}} \;:\; \begin{array}{ccc} \Sigma \times M(\mathcal{A}) & \longrightarrow & M(\mathcal{A}) \\ (B,R) & \longmapsto & R \end{array}$$

The following Proposition says that the coherence conditions required for structure maps of pseudo-functors (see e.g. [9, Def. 1.4.4]) are met by $\eta_{\Sigma,\mathcal{A}}$ and $\mu_{F,G,\mathcal{A}}$. The proof is trivial.

**Proposition 6.5.** *The maps $\eta_\Sigma$ and $\mu_{F,G}$ defined above are natural isomorphisms as in (6.5). Moreover, for each automaton $\mathcal{A} : \Sigma$ and each $\mathbf{M}$-maps $F, G, H$ of appropriate domains and codomains, the two diagrams of Figure 8 commute.*

The assignment $(-)^\star : \mathbf{M}^{\mathrm{op}} \to \mathbf{Cat}$ taking the alphabet $\Sigma$ to the category $\mathsf{Aut}_\Sigma$ and the morphism $F : \Gamma \to_{\mathbf{M}} \Sigma$ to the functor $(-)[\mathcal{M}_F] : \mathsf{Aut}_\Sigma \to \mathsf{Aut}_\Gamma$ is thus a pseudo-functor.

## 7. Conclusion

In this paper, we revisited Church's synthesis *via* an automata-based realizability interpretation of an intuitionistic proof system $\mathsf{SMSO}$ for $\mathsf{MSO}$ on $\omega$-words, and we demonstrated that our approach is sound and complete, in the sense of Thm. 3.5. As it stands, this approach must still pay the price of the non-elementary lower-bound for the translation of $\mathsf{MSO}$ formulae over finite words to DFA's (see Remark 4.9) and the system $\mathsf{SMSO}$ is limited by its set of connectives and its restricted induction scheme.

**Further Works.** First, the indexed structure (§6.5) induced by the substitution operation of §6.2 suggests that in our context, it may be profitable to work in a conservative extension of $(\mathsf{S})\mathsf{MSO}$, with one function symbol for each Mealy machine together with defining axioms of the form (4.2). In particular, this could help mitigate Remark 4.9 by giving the possibility, in the synchronous comprehension scheme of $\mathsf{SMSO}$, to give a term for a Mealy machine rather than the $\mathsf{MSO}$-formula representing it. 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).

Second, following the approach of [19], $\mathsf{SMSO}$ could be extended with primitive universal quantifications and implications as soon as one goes to a *linear* deduction system. Among outcomes of going to a linear deduction system, following [19] we expect similar proof-theoretical properties as with the usual *Dialectica* interpretation (see e.g. [10]), such as realizers of linear Markov rules and choices schemes. Also, having primitive universal quantifications may allow to take benefit of the reductions of MSO to its negative fragment, as provided by the *Safraless* approaches to synthesis [12, 11, 6].

Obtaining a good handle of induction in SMSO 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 possibility could be to base our deduction system on a complete equational theory for Mealy machines.

## 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
[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. 1, 4, 5
[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. 1, 5
[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. 1
[6] E. Filiot, N. Jin, and J.-F. Raskin. Antichains and compositional algorithms for LTL synthesis. *Form. Method. Syst. Des.*, 39(3):261–296, Dec 2011. 26
[7] J.-Y. Girard. *Interprétation Fonctionnelle et Élimination des Coupures de l'Arithmétique d'Ordre Supérieur*. PhD thesis, Université Paris 7, 1972. 8
[8] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002. 5, 15
[9] B. Jacobs. *Categorical Logic and Type Theory*. Studies in logic and the foundations of mathematics. Elsevier, 2001. 2, 20, 21, 23, 24, 25
[10] U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Springer Monographs in Mathematics. Springer, 2008. 2, 8, 10, 26
[11] O. Kupferman, N. Piterman, and Y. Vardi, M. Safraless Compositional Synthesis. In T. Ball and R. B. Jones, editors, *Proceedings of CAV'06*, pages 31–44. Springer, 2006. 1, 26
[12] O. Kupferman and M. Y. Vardi. Safraless decision procedures. In *Proceedings of FOCS'05*, pages 531–542, Washington, DC, USA, 2005. IEEE Computer Society. 26
[13] S. Mac Lane. *Categories for the Working Mathematician*. Springer, 2nd edition, 1998. 4, 23
[14] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521 – 530, 1966. 1, 5
[15] D. Perrin and J.-É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Pure and Applied Mathematics. Elsevier, 2004. 1, 5, 17, 18
[16] P. Pradic and C. Riba. A Curry-Howard Approach to Church's Synthesis. In *Proceedings ot FSCD'17*, volume 84 of *LIPIcs*, pages 30:1–30:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. 1
[17] M. O. Rabin. Automata on infinite objects and Church's Problem. *Amer. Math. Soc.*, 1972. 1, 5
[18] 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. 2, 6, 7, 8, 11, 13, 27, 31
[19] C. Riba. A Dialectica-Like Approach to Tree Automata. Available on HAL (hal-01261183), 2016. URL: `https://hal.archives-ouvertes.fr/hal-01261183`. 15, 25, 26

**$\mathsf{S}$ is the Successor for $\dot{<}$:**

$$\forall x, y\,[\mathsf{S}(x,y) \quad \longleftrightarrow \quad (x \mathrel{\dot{<}} y \ \wedge\ \neg\exists z(x \mathrel{\dot{<}} z \mathrel{\dot{<}} y))]$$

**Strict Linear Order Axioms:**

$$\neg(x \mathrel{\dot{<}} x) \qquad\qquad (x \mathrel{\dot{<}} y \mathrel{\dot{<}} z \to x \mathrel{\dot{<}} z) \qquad\qquad (x \mathrel{\dot{<}} y \ \vee\ x \doteq y \ \vee\ y \mathrel{\dot{<}} x)$$

**Predecessor and Unboundedness Axioms:**

$$\forall x\,[\exists y(y \mathrel{\dot{<}} x) \quad\longrightarrow\quad \exists y \mathsf{S}(y,x)] \qquad\qquad \forall x \exists y(x \mathrel{\dot{<}} y)$$

Figure 9: The Arithmetic Axioms of [18].


[20] D. Siefkes. *Decidable Theories I : Büchi's Monadic Second Order Successor Arithmetic*, volume 120 of *LNM*. Springer, 1970. 2, 6, 7, 11, 13

[21] The Coq Developement Team. *The Coq Proof Assistant Reference Manual*, 2016. `http://coq.inria.fr/`. 2

[22] W. Thomas. Languages, Automata, and Logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume III, pages 389–455. Springer, 1997. 1, 2, 5, 11, 12, 14, 17, 18

[23] W. Thomas. Solution of Church's Problem: A tutorial. *New Perspectives on Games and Interaction*, 5:23, 2008. 1, 3, 5, 12

[24] W. Thomas. Facets of Synthesis: Revisiting Church's Problem. In L. de Alfaro, editor, *Proceedings of FOSSACS'09*, pages 1–14. Springer, 2009. 1


## Appendix A. Completeness of MSO (Thm. 2.12)

In this Appendix, we provide the missing details allowing to deduce the completeness of our axiomatization of MSO (Thm. 2.12) from [18].

The arithmetic axioms of [18] expressed with $\dot{<}$ and $\mathsf{S}$ are presented in Figure 9, where

$$(x \mathrel{\dot{<}} y) \quad := \quad \left[x \mathrel{\dot{\leq}} y \ \wedge\ \neg(x \doteq y)\right]$$

Note that the axioms of [18] are expressed in terms of $\dot{<}$ only. Figure 9 is thus expressed in an extension by definition of [18].

The axioms of Figure 9 follow from Lemma 2.13 (Figure 4) that we prove now.

*Proof of Lemma 2.13.* Let us recall the axioms of MSO (omitting equality):

- $\dot{\leq}$ is a partial order:

$$\frac{}{\overline{\varphi} \vdash x \mathrel{\dot{\leq}} x} \qquad \frac{\overline{\varphi} \vdash x \mathrel{\dot{\leq}} y \qquad \overline{\varphi} \vdash y \mathrel{\dot{\leq}} z}{\overline{\varphi} \vdash x \mathrel{\dot{\leq}} z} \qquad \frac{\overline{\varphi} \vdash x \mathrel{\dot{\leq}} y \qquad \overline{\varphi} \vdash y \mathrel{\dot{\leq}} x}{\overline{\varphi} \vdash x \doteq y}$$

- Basic $\mathsf{Z}$ and $\mathsf{S}$ axioms (total injective relations):

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

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

- Arithmetic axioms:

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

We now proceed to the proof of the properties listed in Figure 4.

(1) $\vdash \neg(x \mathrel{\dot{<}} x)$

*Proof.* From reflexivity of equality. □

(2) $x \mathrel{\dot{<}} y, y \mathrel{\dot{<}} z \vdash x \mathrel{\dot{<}} z$

*Proof.* We have $x \mathrel{\dot{<}} y, y \mathrel{\dot{<}} z \vdash x \mathrel{\dot{\leq}} z$ and $x \mathrel{\dot{<}} y, y \mathrel{\dot{<}} z, x \doteq z \vdash \bot$ by the partial order axioms for $\dot{\leq}$. □

(3) $\mathsf{S}(x, y), x \doteq y \vdash \bot$

*Proof.* By induction on $y$, we show

$$\phi[y] \quad := \quad \forall x(\mathsf{S}(x, y) \to \neg(x \doteq y))$$

We have $\mathsf{Z}(y) \vdash \phi[y]$ by the first arithmetic axiom. We now show $\phi[y], \mathsf{S}(y, y') \vdash \phi[y']$, that is

$$\phi[y], \mathsf{S}(y, y'), \mathsf{S}(x, y'), x \doteq y' \vdash \bot$$

Note that

$$\mathsf{S}(y, y'), \mathsf{S}(x, y'), x \doteq y' \vdash x \doteq y \ \wedge \ y \doteq y' \ \wedge \ \mathsf{S}(x, y)$$

From which follows that

$$\phi[y], \mathsf{S}(y, y'), \mathsf{S}(x, y'), x \doteq y' \vdash \bot$$

□

(4) $\vdash \forall x \exists y \ x \mathrel{\dot{<}} y$.

*Proof.* The basic and arithmetic axioms above ensure that every $x$ has a successor $y$, and that the successor satisfies $x \mathrel{\dot{\leq}} y$. Thus, in combination with (3), we get that $x \mathrel{\dot{<}} y$. □

(5) $\mathsf{S}(y, y'), x \mathrel{\dot{\leq}} y, x \doteq y' \vdash \bot$

*Proof.* Because

$$\mathsf{S}(y, y'), x \mathrel{\dot{\leq}} y, x \doteq y' \vdash y' \mathrel{\dot{\leq}} y$$

and by the partial order axioms for $\dot{\leq}$ together with the second arithmetic axiom, we have

$$\mathsf{S}(y, y'), x \mathrel{\dot{\leq}} y, x \doteq y' \vdash y' \doteq y$$

and we conclude by (3). □

(6) $\mathsf{Z}(x) \vdash x \mathrel{\dot{\leq}} y$

*Proof.* By induction on $y$. □

(7) $x \mathrel{\dot{\leq}} y, \mathsf{Z}(y) \vdash \mathsf{Z}(x)$

*Proof.* By (6), we have $\mathsf{Z}(y) \vdash y \mathrel{\dot{\leq}} x$ and we conclude by the partial order axioms for $\dot{\leq}$. □

(8) $\forall y(x \mathrel{\dot{\leq}} y) \vdash \mathsf{Z}(x)$

*Proof.* We have

$$\forall y(x \mathrel{\dot{\leq}} y), \mathsf{Z}(z) \vdash x \mathrel{\dot{\leq}} z$$

Hence by (7) we get

$$\forall y(x \mathrel{\dot{\leq}} y), \mathsf{Z}(z) \vdash \mathsf{Z}(x)$$

and we conclude by the basic axioms for $\mathsf{Z}$. □

(9) $x \dot{<} y, \mathsf{S}(x,x') \vdash x' \dot{\leq} y$

*Proof.* By induction on $y$, we show
$$\phi[y] \quad := \quad \forall x, x' \left(x \dot{<} y \to \mathsf{S}(x,x') \to x' \dot{\leq} y\right)$$
First, the base case $\mathsf{Z}(y) \vdash \phi[y]$ follows from the fact that $\mathsf{Z}(y), x \dot{<} y \vdash \bot$ by (7). For the induction step, we show
$$\mathsf{S}(y,y'), \phi[y], x \dot{<} y', \mathsf{S}(x,x') \vdash x' \dot{\leq} y'$$
We use the excluded middle on $x \doteq y$, and we are left to show
$$\mathsf{S}(y,y'), \phi[y], x \dot{<} y', \neg(x \doteq y), \mathsf{S}(x,x') \vdash x' \dot{\leq} y'$$
But by the arithmetic axioms, $\mathsf{S}(y,y'), x \dot{<} y' \vdash x \dot{\leq} y$, so that
$$\mathsf{S}(y,y'), \phi[y], x \dot{<} y', \neg(x \doteq y), \mathsf{S}(x,x') \vdash x \dot{<} y$$
But
$$\phi[y], x \dot{<} y, \mathsf{S}(x,x') \vdash x' \dot{\leq} y$$
and we are done. □

(10) $x \dot{\leq} y, \mathsf{S}(x,x'), \mathsf{S}(y,y') \vdash x' \dot{\leq} y'$

*Proof.* By induction on $z$ we show
$$\phi[z] \quad := \quad \forall x, x', y \left(x \dot{\leq} y \to \mathsf{S}(x,x') \to \mathsf{S}(y,z) \to x' \dot{\leq} z\right)$$
We trivially have $\mathsf{Z}(z) \vdash \phi[z]$. We now show $\mathsf{S}(z,z'), \phi[z] \vdash \phi[z']$, that is
$$\mathsf{S}(z,z'), \phi[z], x \dot{\leq} y, \mathsf{S}(x,x'), \mathsf{S}(y,z') \vdash x' \dot{\leq} z'$$
By the basic $\mathsf{Z}$ and $\mathsf{S}$ axioms, this amounts to show
$$\phi[z], x \dot{\leq} z, \mathsf{S}(x,x'), \mathsf{S}(z,z') \vdash x' \dot{\leq} z'$$
Now, using the excluded middle on $x \doteq z$, we are left with showing
$$\phi[z], \mathsf{S}(x,x'), \mathsf{S}(z,z'), x \dot{<} z \vdash x' \dot{\leq} z'$$
But by (9) we have
$$x \dot{<} z, \mathsf{S}(x,x') \vdash x' \dot{\leq} z$$
and we are done. □

(11) $\vdash \forall x \forall y \left[y \dot{<} x \quad \longleftrightarrow \quad \exists z (y \dot{\leq} z \ \wedge \ \mathsf{S}(z,x))\right]$

*Proof.* The right-to-left direction follows from (3). For the left-to-right direction, by induction on $x$, we show
$$\phi[x] \quad := \quad \forall y \left(y \dot{<} x \to \exists z (y \dot{\leq} z \ \wedge \ \mathsf{S}(z,x))\right)$$
For the base case $\mathsf{Z}(x) \vdash \phi[x]$, by (7) we have $\mathsf{Z}(x), y \dot{\leq} x \vdash \mathsf{Z}(y)$ and we conclude by the basic $\mathsf{Z}$-axioms. For the induction step, we have to show
$$\mathsf{S}(x,x'), \phi[x], y \dot{<} x' \vdash \exists z (y \dot{\leq} z \ \wedge \ \mathsf{S}(z,x'))$$
By the last arithmetic axiom,
$$\mathsf{S}(x,x'), y \dot{<} x' \vdash y \dot{\leq} x$$
and we are done. □

(12) $\vdash x \mathrel{\dot{<}} y \vee x \doteq y \vee y \mathrel{\dot{<}} x$

*Proof.* By induction on $x$, we show

$$\phi[x] \quad := \quad \forall y\,(x \mathrel{\dot{<}} y \vee x \doteq y \vee y \mathrel{\dot{<}} x)$$

The base case $\mathsf{Z}[x] \vdash \phi[x]$ follows from (6). For the induction step, we have to show

$$\mathsf{S}(x, x'), \phi[x] \vdash \forall y(x' \mathrel{\dot{<}} y \vee x' \doteq y \vee y \mathrel{\dot{<}} x')$$

By induction on $y$ we show $\mathsf{S}(x, x'), \phi[x] \vdash \psi[y, x']$ where

$$\psi[y, x'] \quad := \quad x' \mathrel{\dot{<}} y \vee x' \doteq y \vee y \mathrel{\dot{<}} x'$$

The base case follows again from (6). For the induction step, we have to show

$$\mathsf{S}(x, x'), \mathsf{S}(y, y'), \phi[x], \psi[y, x'] \vdash \forall y(x' \mathrel{\dot{<}} y' \vee x' \doteq y' \vee y' \mathrel{\dot{<}} x')$$

and we are done since (10) gives

$$\forall x, x', y, y' \left(x \mathrel{\dot{<}} y \to \mathsf{S}(x, x') \to \mathsf{S}(y, y') \to x' \mathrel{\dot{<}} y'\right)$$

□

(13) $\vdash \forall x, y\,[\mathsf{S}(x, y) \quad \longleftrightarrow \quad (x \mathrel{\dot{<}} y \ \wedge\ \neg\exists z(x \mathrel{\dot{<}} z \mathrel{\dot{<}} y))]$

*Proof.* For the left-to-right direction, thanks to (3) we are left to show

$$\mathsf{S}(x, y), x \mathrel{\dot{<}} z, z \mathrel{\dot{<}} y \vdash \bot$$

Assume that $y$ is the successor of $x$ and an intermediate $z$ such that $x \mathrel{\dot{<}} z \mathrel{\dot{<}} y$ (in particular, we have $x \mathrel{\dot{<}} y$ by (2)). By (10), $y \mathrel{\dot{\leq}} z$. But then it means that $z = y$ by $z \mathrel{\dot{\leq}} y$ by antisymmetry, which is contradictory.

Conversely, assume that $x \mathrel{\dot{<}} y$ without any intermediate $z$. Assume $\neg\mathsf{S}(x, y)$ towards a contradiction. But there exists some $z$ such that $\mathsf{S}(x, z)$. Clearly, $x \mathrel{\dot{<}} z$ by (3). But we also have $z \mathrel{\dot{\leq}} y$ by (9) and $\neg y \doteq z$ by assumption, so $z \mathrel{\dot{<}} y$. This contradicts our assumption of non-existence of intermediate $z$. □

This concludes the proof of Lemma 2.13. □

The linear order axioms ((1), (2), (12)), the successor axiom (13), the unboundedness (4) and predecessor (11) axioms are thus proved in our axiomatic.

Finally, we have to prove Lemma 2.14, namely that strong induction is derivable in $\mathsf{MSO}$. The proof holds no surprise.

*Proof of Lemma 2.14.* We have to show

$$\forall x(\forall y(y \mathrel{\dot{<}} x \to X(y)) \to X(x)) \vdash \forall x\ X(x)$$

By induction on $x$ we show

$$\forall x(\forall y(y \mathrel{\dot{<}} x \to X(y)) \to X(x)) \vdash \phi[x]$$

where

$$\phi[x] \quad := \quad \forall y(y \mathrel{\dot{\leq}} x \ \to\ X(y))$$

The base case

$$\forall x(\forall y(y \mathrel{\dot{<}} x \to X(y)) \to X(x)), \mathsf{Z}(x) \vdash \phi[x]$$

is trivial since by (6),

$$\mathsf{Z}(x) \vdash \neg\exists y(y \mathrel{\dot{<}} x)$$

For the induction step, we have to show

$$\forall x(\forall y(y \mathrel{\dot{<}} x \to X(y)) \to X(x)), \mathsf{S}(x, x'), \phi[x], z \mathrel{\dot{\leq}} x' \vdash X(z)$$

Notice that $\phi[x]$ is equivalent to $\phi'[x'] := \forall y(y \mathrel{\dot{<}} x' \to X(y))$ using (11). Using trichotomy (12), we have three subcase according to:

$$z \mathrel{\dot{<}} x' \vee z \doteq x' \vee x' \mathrel{\dot{<}} z$$

The first case enables to use $\phi'[x']$ directly, and the second requires to use the hypothesis of $\dot{<}$-induction $\forall x(\forall y(y \mathrel{\dot{<}} x \to X(y)) \to X(x)$ together with $\phi'[x']$. The last one leads to a contradiction using antisymmetry of $\dot{\leq}$ together with $z \mathrel{\dot{<}} x' = z \mathrel{\dot{\leq}} x' \wedge \neg z \doteq x'$. $\square$

## Appendix B. Internally Bounded Formulae (Thm. 4.11)

We work in the standard model $\mathfrak{N}$ of $\mathsf{MSO}$. Our goal is to show Thm. 4.11, that if a formula $\psi[\overline{X}, x]$ can be proven to behave as a bounded formula (in the sense of $\mathsf{B}^{x}_{\overline{X}}(\psi[\overline{X}, x])$) then $\psi[\overline{X}, x]$ is actually equivalent to a uniformly bounded formula.

In the following, we consider $\mathsf{MSO}$-formulae over the vocabulary of [18], that is $\mathsf{MSO}$-formulae given by the grammar

$$\varphi, \psi \in \Lambda \quad ::= \quad x \mathrel{\dot{\in}} X \quad | \quad x \mathrel{\dot{<}} y \quad | \quad \neg\varphi \quad | \quad \varphi \vee \psi \quad | \quad \exists X\, \varphi \quad | \quad \exists x\, \varphi$$

Following §2.5 (see also §A), defining the atomic formulae $\doteq$, $\mathsf{S}(-,-)$, $\dot{\leq}$ and $\mathsf{Z}(-)$ as

$$\begin{array}{rcl} x \doteq y & := & \forall X\,(x \mathrel{\dot{\in}} X \;\to\; y \mathrel{\dot{\in}} X) \\ \mathsf{S}(x, y) & := & (x \mathrel{\dot{<}} y \;\wedge\; \neg\exists z(x \mathrel{\dot{<}} z \mathrel{\dot{<}} y)) \\ x \mathrel{\dot{\leq}} y & := & (x \mathrel{\dot{<}} y) \vee (x \doteq y) \\ \mathsf{Z}(x) & := & \forall y\,(x \mathrel{\dot{\leq}} y) \end{array}$$

we obtain for each formula in the sense of Fig. 2 an equivalent formula in $\Lambda$. Moreover, we write $\mathrm{FV}^{\iota}(\varphi)$ for the set of free individual variables of the formula $\varphi$.

We shall make use of the following usual transfer property (see e.g. [18]).

**Lemma B.1** (Transfer)**.** *Let $\varphi \in \Lambda$ be a formula with free variables among $\overline{x} = x_1, \ldots, x_p$ and $\overline{X} = X_1, \ldots, X_q$. Furthermore, let $A \in \mathbf{2}^{\omega} \simeq \mathcal{P}(\mathbb{N})$ be non-empty. Then for all $a_1, \ldots, a_n \in A$ and all $\overline{B} \in (\mathbf{2}^{\omega})^q$ we have*

$$\mathfrak{N}{\restriction}A \models \varphi[\overline{a}/\overline{x}, \overline{B \cap A}/\overline{X}] \qquad \Longleftrightarrow \qquad \mathfrak{N} \models (\varphi[\overline{a}/\overline{x}, \overline{B}/\overline{X}]){\restriction}[A(-)]$$

Lemma B.1 gives in particular that if $B_0, B_1 \in \mathbf{2}^{\omega} \simeq \mathcal{P}(\mathbb{N})$ are disjoint, then

$$\mathfrak{N} \quad \models \quad \exists X(\phi_0{\restriction}B_0 \wedge \phi_1{\restriction}B_1) \;\longleftrightarrow\; (\exists X(\phi_0{\restriction}B_0) \;\wedge\; \exists X(\phi_1{\restriction}B_1)) \tag{B.1}$$

**Lemma B.2** (Splitting)**.** *Let $\psi \in \Lambda$ be an $\mathsf{MSO}$-formula and $z$ be an individual variables. For every $V \subseteq \mathrm{FV}^{\iota}(\psi)$ with $z \in V$, one can produce natural number $N_\psi$ and two matching sequences of length $N_\psi$ of left formulae $(L_V(\psi)_j)_{j<N_\psi}$ and right formulae $(R_V(\psi)_j)_{j<N_\psi}$ such that the following holds:*

- *For every $j < N_\psi$, $\mathrm{FV}^{\iota}(L_V(\psi)_j) \subseteq V$ and $\mathrm{FV}^{\iota}(R_V(\psi)_j) \subseteq \mathrm{FV}^{\iota}(\psi) \setminus V$.*
- *If $\mathrm{FV}^{\iota}(\psi) = \{\overline{x}, z, \overline{y}\}$ with $V = \{\overline{x}, z\}$, then for all $n \in \mathbb{N}$, all $\overline{a} \leq n$ and all $\overline{b} > n$, we have*

$$\mathfrak{N} \quad \models \quad \psi[\overline{a}/\overline{x}, n/z, \overline{b}/\overline{y}] \;\longleftrightarrow\; \bigvee_{j<N_\psi} L_V(\psi)_j[\overline{a}/\overline{x}, n/z]{\restriction}[- \mathrel{\dot{\leq}} n] \wedge R_V(\psi)_j[\overline{b}/\overline{y}]{\restriction}[- \mathrel{\dot{>}} n]$$

*Proof.* The proof proceeds by induction on $\psi$.

- If $\psi = x \mathrel{\dot{<}} y$, then $N_\psi := 1$. We define suitable left and right formulae according to $V$:
  - If $V = \{x, y\}$, then $L_V(\psi)_0 := \psi$ and $R_V(\psi)_0 := \top$
  - If $V = \emptyset$, then $L_V(\psi)_0 := \top$ and $R_V(\psi)_0 := \psi$
  - If $V = \{x\}$, then $L_V(\psi)_0 := R_V(\psi)_0 := \top$
  - If $V = \{y\}$, then $L_V(\psi)_0 := R_V(\psi)_0 := \bot$
- If $\psi = X(x)$, then $N_\psi := 1$. One of the produced formula is $\psi$ and the other is $\top$ according to whether $x \in V$ or not.
- If $\psi = \varphi \vee \phi$, then it is clear that taking the concatenation of the sequences of formulae given by the induction hypothesis will be enough
$$\begin{array}{lclclcl} L_V(\psi)_j & := & L_V(\varphi)_j & \qquad & R_V(\psi)_j & := & R_V(\varphi)_j \\ L_V(\psi)_{j+N_\varphi} & := & L_V(\phi)_j & & R_V(\psi)_{j+N_\varphi} & := & R_V(\phi)_j \end{array}$$
- If $\psi = \exists x.\phi$, using our induction hypothesis, we have $N_\phi \in \mathbb{N}$ and sequences of formulae $(L_V(\phi)_j, R_V(\phi)_j, L_{V\cup\{x\}}(\phi)_j, R_{V\cup\{x\}}(\phi)_j)_{j<N_\phi}$ satisfying the conclusion of the theorem.
  For $i < N_\phi$, define
$$\begin{array}{lclclcl} L_V(\psi)_j & := & \exists x. L_{V\cup\{x\}}(\phi)_j & \qquad & R_V(\psi)_j & := & R_{V\cup\{x\}}(\phi)_j \\ L_V(\psi)_{j+N_\phi} & := & L_V(\phi)_j & & R_V(\psi)_{j+N_\phi} & := & \exists x. R_V(\phi)_j \end{array}$$
  The disjunction is equivalent to $\exists x.\phi$ by making a case analysis over whether $x \leq n$ is true.
- If $\psi = \exists X.\varphi$, then it directly follows from (B.1) that we can take $L_V(\psi)_j := \exists X. L_V(\varphi)_j$ and $R_V(\psi)_j := \exists X. R_V(\varphi)_j$ for $j < N_\varphi$.
- If $\psi = \neg\phi$, using our induction hypothesis, we have a natural number $N_\phi$ and two sequences of formulae $(L_V(\phi), R_V(\phi))_{j<N_\phi}$ such that $\phi \longleftrightarrow \bigvee_{j<N_\phi} L_V(\phi)_j {\upharpoonright} [- \mathrel{\dot{\leq}} n] \wedge R_V(\phi)_j {\upharpoonright} [- \mathrel{\dot{>}} n]$. Hence all we need to do is to add the negation, push it through the disjunction and convert the obtained CNF into a DNF.
  More explicitly (leaving the parameters implicit), we have:
$$\begin{array}{lcl} \neg\phi & \longleftrightarrow & \displaystyle\bigwedge_{j<N_\phi} \neg L_V(\phi)_j {\upharpoonright} [- \mathrel{\dot{\leq}} n] \vee \neg R_V(\phi)_j {\upharpoonright} [- \mathrel{\dot{>}} n] \\ & \longleftrightarrow & \displaystyle\bigvee_{f \in \mathbf{2}^{[\![0, N_\phi - 1]\!]}} \bigwedge_{j \in f^{-1}(0)} \neg L_V(\phi)_j {\upharpoonright} [- \mathrel{\dot{\leq}} n] \wedge \bigwedge_{j \in f^{-1}(1)} \neg R_V(\phi)_j {\upharpoonright} [- \mathrel{\dot{>}} n] \end{array}$$

□

**Remark B.3.** Note that there is a combinatorial explosion in the case of $\psi = \neg\phi$ in Lem. B.2 since $N_{\neg\phi} = 2^{N_\phi}$. It follows that the size of the formulae produced in Lem. B.2 is non-elementary in the size of $\psi$.

We can now prove Thm. 4.11. Recall that
$$\mathsf{B}^x_{\overline{X}}(\psi[\overline{X}, x]) \quad := \quad \forall z \forall \overline{Z Z'} (\forall y \mathrel{\dot{\leq}} z \overline{[Zz \longleftrightarrow Z'z]} \;\longrightarrow\; [\psi[\overline{Z}/\overline{X}, z/x] \longleftrightarrow \psi[\overline{Z'}/\overline{X}, z/x]])$$

**Theorem B.4** (Thm. 4.11)**.** *If $\mathfrak{N} \models \mathsf{B}^x_{\overline{X}}(\psi[\overline{X}, x])$ then there is a uniformly bounded formula $\hat{\varphi}[\overline{X}, x]$ which is effectively computable from $\psi$ and such that $\mathfrak{N} \models \forall \overline{X} \forall x\, (\psi[\overline{X}, x] \longleftrightarrow \hat{\varphi}[\overline{X}, x])$.*

*Proof.* We assume $\psi \in \Lambda$. Using Lem. B.2, we know that $\psi[\overline{X}, x]$ is equivalent to
$$\varphi[\overline{X}, x] \quad := \quad \bigvee_j L_j(x, \overline{X}) {\upharpoonright} [- \mathrel{\dot{\leq}} x] \wedge R_j(\overline{X}) {\upharpoonright} [- \mathrel{\dot{>}} x]$$

writing $L_j$ as a shorthand for $L_{\{x\}}(\psi(x,\overline{X}))_j$ (and similarly for $R_j$). Then

$$\begin{array}{rcl} \varphi[\overline{X},x] & \longleftrightarrow & \varphi[\overline{X(-)\wedge - \dot{\leq} x}, x] \\ & \longleftrightarrow & \bigvee_j L_j(x, \overline{X(-)\wedge - \dot{\leq} x})\restriction[- \dot{\leq} x] \wedge R_j(\overline{X \wedge - \dot{\leq} x})\restriction[- \dot{>} x] \end{array}$$

Again using Lem. B.1, we know that, for every $j$, $R_j(\overline{X(-)\wedge - \dot{\leq} n})\restriction[- \dot{>} n]$ is equivalent to $R_j(\overline{X(-)\wedge - \dot{\leq} n \wedge - \dot{>} n})\restriction[- \dot{>} n]$. By substitutivity, it is equivalent to $R'_j(n)\restriction[- \dot{>} n]$, where we set $R'_j := R_j(\overline{\bot})$. Lemma B.1 moreover implies (since $R'_j$ is a sentence and since $\mathfrak{N}\restriction[- \dot{>} n] \simeq \mathfrak{N}$), that

$$\mathfrak{N} \models \left(R'_j \longleftrightarrow R'_j\restriction[- \dot{>} n]\right)$$

But now, since $R'_j$ is closed, it follows from the decidability of MSO (Thm. 2.8) that we can decide whether $\mathfrak{N} \models R'_j$ or $\mathfrak{N} \models \neg R'_j$. It follows that our initial $\psi$ is equivalent to the following formula, which is effectively computable from $\psi$:

$$\left(\bigvee_j L_j(n) \wedge R''_j\right)\restriction[- \leq n]$$

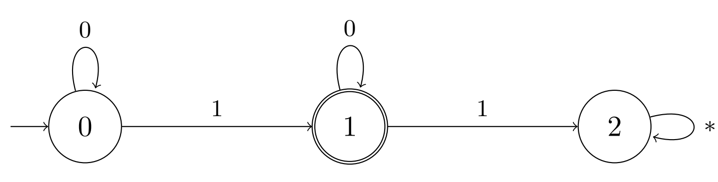
where each $R''_j$ is either $\top$ or $\bot$ according $\mathfrak{N} \models R'_j$ or $\mathfrak{N} \models \neg R'_j$ .

□

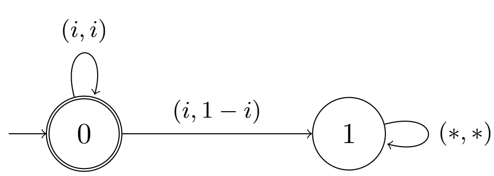
## Appendix C. Automata for Atomic Fomulae (§5.3)

We give below the automaton Sing (of §5.3) and automata for the atomic formulae of MSO. These automata are presented as deterministic Büchi automata (with accepting states circled). As uniform automata, each of them has set of moves **1**. Note that automata for atomic formulae involving individual variables do not detect if the corresponding inputs actually represent natural numbers. This is harmless, since all statements of §5 actually assume streams representing natural numbers to be singletons, and since in Fig. 7, MSO-quantifications over individuals are relativized to Sing.

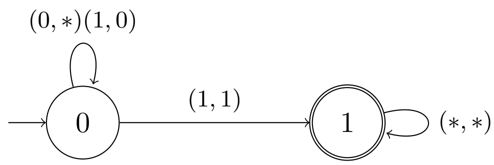
- **Sing :**



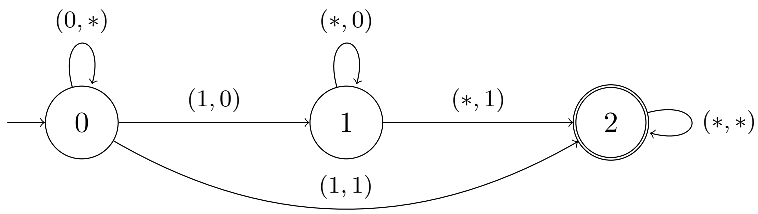

- $\boldsymbol{(x_1 \doteq x_2)}$ **:**



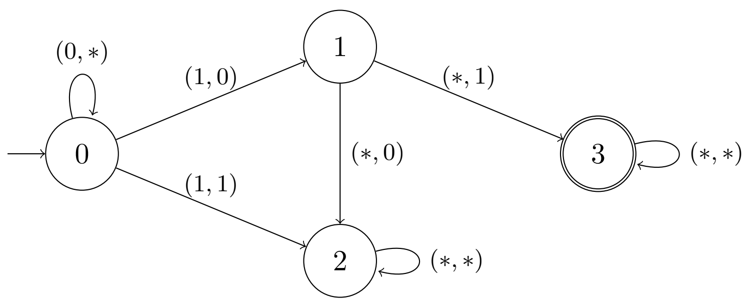
- $\boldsymbol{(x_1 \mathrel{\dot{\in}} X_1)}$ **:**



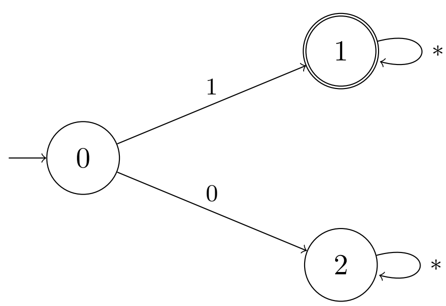

- $\boldsymbol{(x_1 \mathrel{\dot{\leq}} x_2)}$ **:**



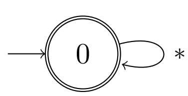

- $\mathsf{S}\boldsymbol{(x_1, x_2)}$ **:**



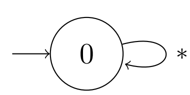

- $\mathsf{Z}\boldsymbol{(x_1)}$ **:**




- $\top$ **and** $\bot$**:**




and



## Contents