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 (MSO) 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, an intuitionistic variant of MSO over infinite words, which is shown to be sound and complete w.r.t. synthesis thanks to an automata-based realizability model.

1. INTRODUCTION

A stream function $F : \Sigma^\omega \to \Gamma^\omega$ is *synchronous* (or *causal*) if it can produce a prefix of length n of its output from a prefix of length n of its input:

$$B(0).\cdots.B(n-1) = C(0).\cdots.C(n-1) \quad \Longrightarrow \quad F(B)(n) = F(C)(n) \quad (\text{for } B, C \in \Sigma^\omega)$$

A synchronous function is *finite-state* if it is induced by a deterministic letter-to-letter stream transducer (or *deterministic Mealy machine*, DMM). Church's synthesis [Chu57] consists in the automatic extraction of DMMs from input-output specifications, typically presented as closed formulae of the form

$$\forall X \in \Sigma^\omega \, \exists Y \in \Gamma^\omega \, \varphi(X; Y) \tag{1.1}$$

where φ is a formula of some subsystem of *Monadic Second-Order Logic* (MSO) over ω-words. A specification as in (1.1) is realized in the sense of Church by a (finite-state) synchronous $F : \Sigma^\omega \to \Gamma^\omega$ when $\varphi(B, F(B))$ holds for all $B \in \Sigma^\omega$.

MSO over ω-words is a decidable logic (Büchi's Theorem [Büc62]) which subsumes logics used in verification such as LTL (see e.g. [Tho97, PP04, VW08]). Traditional approaches to synthesis (see e.g. [Tho08, Tho09]) are based, via McNaughton's Theorem [McN66], on

---

the translation of MSO-formulae to *deterministic* automata on ω-words (such as *Muller* or *parity* automata).[1] These automata are then turned into infinite two-player sequential games on finite graphs, in which the *Opponent* ($\forall$bélard) plays input letters to which the *Proponent* ($\exists$loïse) replies with output letters. Solutions to Church's synthesis are then given by the Büchi-Landweber Theorem [BL69], which states that in such games, (exactly) one of the two players has a finite-state winning strategy (*i.e.* a strategy which only uses a finite memory).

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 (see e.g. [GTW02]), and McNaughton's Theorem involves a non-trivial powerset construction (such as *Safra construction*, see e.g. [Tho97, GTW02, PP04, VW08]). 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) considerable success for verifying concurrency properties, such techniques hardly managed up to now to give practical algorithms for the synthesis of large scale systems (even for fragments of LTL, see e.g. [BJP+12]).

In this work, we propose a Curry-Howard approach to Church's synthesis. The Curry-Howard correspondence asserts that, given a suitable proof system, any proof therein can be interpreted as a program. This interpretation of proofs as programs (as well as the soundness of many type systems) can be formalized using the technique of *realizability*, which tells how to read a formula from the logic as a specification for a program. More precisely, realizability can be seen as a relation between programs (the realizers) and formulae, usually defined by induction on the latter (see e.g. [SU06, Koh08]). Typical clauses state e.g. that realizers of conjunctions $\varphi_1 \wedge \varphi_2$ are pairs $\langle R_1, R_2 \rangle$ consisting of a realizer R_1 of φ_1 and a realizer R_2 of φ_2, and that realizers of existential formulae $\exists X \, \varphi(X)$ are pairs $\langle B, R \rangle$ consisting of a witness B for the $\exists X$ and a realizer R of $\varphi(B)$.

Our starting point is the fact that MSO on ω-words can be completely axiomatized as a subsystem of second-order Peano arithmetic [Sie70] (see also [Rib12]). From the classical axiomatization of MSO, we derive an intuitionistic variant SMSO (for *Synchronous* MSO). SMSO comes equipped with an extraction procedure which is sound and complete w.r.t. Church's synthesis: proofs in SMSO of formulae of the form $\exists \overline{Y} \, \varphi(\overline{X}; \overline{Y})$ (with only $\overline{X}$ free) can be translated to DMMs and such proofs exist for all solvable instances of Church's synthesis. Our approach is *Safraless* in the sense that while we do rely on McNaughton's Theorem for the *correctness* of the extraction procedure (*i.e.* for the "*Adequacy Lemma*" of realizability), we never have to actually use McNaughton's Theorem when extracting DMMs from SMSO-proofs (so that the extracted DMMs never involve determinization of automata on ω-words).[2]

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. We provide in §4 some technical material as well as detailed examples on the representation of DMMs in MSO, and §5 presents our realizability model. Finally, in §6 we rephrase the realizability model in terms of *indexed categories* (see e.g. [Jac01]), an essential step for further generalizations.

---

[1]A solution is also possible via tree automata [Rab72] (see also [KPV06, Tho09]).

[2]On the other hand, usual *Safraless* approaches to synthesis use McNaughton Theorem essentially to bound the search space for potential finite-state realizers, see e.g. [KV05, KPV06, FJR11].

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

2. Church's Synthesis and MSO on Infinite Words

2.1. **Notations.** Alphabets (denoted Σ, Γ, etc.) are finite non-empty sets. Concatenation of words s, t is denoted $s.t$, and ε is the empty word. We use the vectorial notation both for words and finite sequences, so that e.g. $\overline{B}$ denotes a finite sequence $B_1, \ldots, B_n$ and $\overline{\mathsf{a}}$ denotes a word $\mathsf{a}_1. \cdots .\mathsf{a}_n \in \Sigma^*$. Given an ω-word (or stream) $B \in \Sigma^\omega$ and $n \in \mathbb{N}$ we write $B{\restriction}n$ for the finite word $B(0). \cdots .B(n-1) \in \Sigma^*$. For each $k \in \mathbb{N}$, we still write k for the function from $\mathbb{N}$ to $\mathbf{2} = \{0, 1\}$ 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 deterministic letter-to-letter stream transducers (or *deterministic Mealy machines*) from input-output specifications (see e.g. [Tho08]).

Example 2.1. As a typical specification, consider, for a machine which outputs streams $C \in \mathbf{2}^\omega$ from input streams $B \in \mathbf{2}^\omega$, the behavior (from [Tho08]) expressed by

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

In words, the relation $\Phi(B, C)$ imposes $C(n) \in \mathbf{2}$ to be 1 whenever $B(n) \in \mathbf{2}$ is 1, C not be 0 at two consecutive positions, and moreover C to be infinitely often 0 whenever B is infinitely often 0.

We are interested in the realization of such specifications by finite-state deterministic letter-to-letter stream transducers or (deterministic) Mealy machines.

Definition 2.2 (Deterministic Mealy Machine)**.** A *deterministic Mealy machine* (DMM) $\mathcal{M}$ with input alphabet Σ and output alphabet Γ (notation $\mathcal{M} : \Sigma \to \Gamma$) is given by a finite set of states $Q_\mathcal{M}$ with a distinguished initial state $q^i_\mathcal{M} \in Q_\mathcal{M}$, 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: $\partial^*_\mathcal{M}(\varepsilon) := q^i_\mathcal{M}$ and $\partial^*_\mathcal{M}(\overline{\mathsf{a}}.\mathsf{a}) := \pi_1(\partial_\mathcal{M}(\partial^*_\mathcal{M}(\overline{\mathsf{a}}), \mathsf{a}))$.

A DMM $\mathcal{M} : \Sigma \to \Gamma$ 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 prefix of length n of its output from a prefix of length n of its input. These functions are called *synchronous* (or *causal*).

Definition 2.3 (Synchronous Function)**.** A function $F : \Sigma^\omega \to \Gamma^\omega$ is *synchronous* if for all $n \in \mathbb{N}$ and all $B, C \in \Sigma^\omega$ we have $F(B){\restriction}n = F(C){\restriction}n$ whenever $B{\restriction}n = C{\restriction}n$. We say that a synchronous function F is *finite-state* if it is induced by a DMM.

We write $F : \Sigma \to_\mathbf{S} \Gamma$ when F is a synchronous function $\Sigma^\omega \to \Gamma^\omega$, and $F : \Sigma \to_\mathbf{M} \Gamma$ when F is finite-state synchronous.

Examples 2.4.

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

(1) The identity function $\Sigma^\omega \to \Sigma^\omega$ is finite-state synchronous as being induced by the DMM with state set $\mathbf{1} = \{\bullet\}$ and identity transition function $\partial : (\bullet, \mathsf{a}) \longmapsto (\bullet, \mathsf{a})$.
(2) The DMM depicted in 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 DMM depicted in Fig. 1 (right), taken from [Tho08], induces a synchronous function $F : \mathbf{2} \to_\mathbf{M} \mathbf{2}$ such that $\Phi(B, F(B))$ holds for all $B \in \mathbf{2}^\omega$, where Φ is the relation of Ex. 2.1.
(4) Synchronous functions are obviously continuous (taking the product topology on Σ^ω and Γ^ω, with Σ, Γ discrete), but there are continuous functions which are not synchronous, for instance the function $P : \mathbf{2}^\omega \to \mathbf{2}^\omega$ such that $P(B)(n) = 1$ iff $B(n+1) = 1$.

For the definition and adequacy of our realizability interpretation (§5), it is convenient to note that alphabets and (finite-state) synchronous functions form a category.

Proposition 2.5. *Synchronous functions form a category $\mathbf{S}$ whose objects are alphabets and whose morphisms from Σ to Γ are the synchronous functions $\Sigma \to_\mathbf{S} \Gamma$. The identity on Σ is the synchronous function of Ex. 2.4.(1), and composition is usual function composition. Moreover, if the $\mathbf{S}$-maps $F : \Sigma \to_\mathbf{S} \Gamma$ and $G : \Gamma \to_\mathbf{S} \Delta$ are finite-state, then so is $G \circ F$.*

Proof. Since sets and functions form a category, $\mathbf{S}$ is a category as soon as composition of functions preserves synchronicity. Consider synchronous $G : \Gamma^\omega \to \Delta^\omega$ and $F : \Sigma^\omega \to \Gamma^\omega$. Assume $B, C \in \Sigma^\omega$ and $n \in \mathbb{N}$ such that $B{\restriction}n = C{\restriction}n$. Then since F is synchronous it follows that $F(B){\restriction}n = F(C){\restriction}n$, and since G is synchronous we deduce $G(F(B)){\restriction}n = G(F(C)){\restriction}n$, that is

$$(G \circ F)(B){\restriction}n \quad = \quad (G \circ F)(C){\restriction}n$$

For the second part of the statement, assume that G and F are induced respectively by $\mathcal{N} : \Gamma \to \Delta$ and $\mathcal{M} : \Sigma \to \Gamma$. Then $G \circ F$ is induced by the DMM

$$(\mathcal{K} : \Sigma \to \Delta) \quad := \quad \big(Q_\mathcal{M} \times Q_\mathcal{N} \,,\, (q_\mathcal{M}^i, q_\mathcal{N}^i) \,,\, \partial_\mathcal{K}\big)$$

whose transition function

$$\partial_\mathcal{K} \quad : \quad (Q_\mathcal{M} \times Q_\mathcal{N}) \times \Sigma \quad \longrightarrow \quad (Q_\mathcal{M} \times Q_\mathcal{N}) \times \Delta$$

takes $((q_\mathcal{M}, q_\mathcal{N}), \mathsf{a})$ to $((q_\mathcal{M}', q_\mathcal{N}'), \mathsf{d})$ with

$$\begin{aligned}
(q_\mathcal{N}', \mathsf{d}) &:= \partial_\mathcal{N}(q_\mathcal{N}, \mathsf{b}) \\
(q_\mathcal{M}', \mathsf{b}) &:= \partial_\mathcal{M}(q_\mathcal{M}, \mathsf{a})
\end{aligned}$$

$\square$

Atoms:	α	$::=$	$x \doteq y$	$\mid$	$x \overset{.}{\leq} y$	$\mid$	$\mathsf{S}(x,y)$	$\mid$	$\mathsf{Z}(x)$	$\mid$	$x \overset{.}{\in} X$	$\mid$	$\top$	$\mid$	$\bot$
Deterministic formulae:	δ, δ'	$::=$	α	$\mid$	$\delta \wedge \delta'$	$\mid$	$\neg \varphi$								
MSO formulae:	φ, ψ	$::=$	δ	$\mid$	$\varphi \wedge \psi$	$\mid$	$\exists x\, \varphi$	$\mid$	$\exists X\, \varphi$						

Figure 2: The Formulae of MSO and SMSO.

Proposition 2.5 implies that **S** has a wide subcategory consisting of *finite-state* functions.

Definition 2.6 (The Category **M**)**.** Let **M** be the category whose objects are alphabets and whose morphisms from Σ to Γ are finite-state synchronous *functions* $\Sigma \to_{\mathbf{M}} \Gamma$.

Note that for **M** to be a category (namely for the associativity and identity laws of composition) it is essential that **M**-maps consist of *functions* rather than *machines*. The following obvious fact is useful for our realizability model (§5).

Remark 2.7. Functions $\mathtt{f} : \Sigma \to \Gamma$ induce **M**-maps $[\mathtt{f}] : \Sigma \to_{\mathbf{M}} \Gamma$.

It is also worth noticing that the category **M** has finite products.

Proposition 2.8. *The category* **M** *has finite products. The product of* $\Sigma_1, \ldots, \Sigma_n$ *(for* $n \geq 0$*) is given by the* **Set***-product* $\Sigma_1 \times \cdots \times \Sigma_n$ *(so that* **1** *is terminal in* **M***).*

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

MSO formulae are interpreted in the standard model $\mathfrak{N}$ of ω-words as usual. Individual variables range over natural numbers $n, m, \ldots \in \mathbb{N}$ and predicate variables range over sets of natural numbers $B, C, \ldots \in \mathcal{P}(\mathbb{N}) \simeq \mathbf{2}^{\omega}$. The atomic predicates are interpreted as expected: $\doteq$ is equality, $\overset{.}{\in}$ is membership, $\overset{.}{\leq}$ is the relation $\leq$ on $\mathbb{N}$, S is the successor relation, and Z holds on n iff $n = 0$. We write $\mathfrak{N} \models \varphi$ when the closed formula φ holds under this interpretation.

We often write $X(x)$ or even Xx for $x \overset{.}{\in} X$. We also use the following abbreviations.

Notation 2.9. Given formulae φ and ψ, we let

$$
\begin{aligned}
\varphi \vee \psi &:= \neg(\neg \varphi \wedge \neg \psi) & \forall x\, \varphi &:= \neg \exists x\, \neg \varphi \\
\varphi \to \psi &:= \neg(\varphi \wedge \neg \psi) & \forall X\, \varphi &:= \neg \exists X\, \neg \varphi \\
\varphi \leftrightarrow \psi &:= (\varphi \to \psi) \wedge (\psi \to \varphi) & x \overset{.}{<} y &:= (x \overset{.}{\leq} y \ \wedge \ \neg(x \doteq y)) \\
\exists x \overset{.}{\leq} y\, \varphi &:= \exists x\, (x \overset{.}{\leq} y \ \wedge \ \varphi) & \forall x \overset{.}{\leq} y\, \varphi &:= \forall x\, (x \overset{.}{\leq} y \ \to \ \varphi) \\
\exists x \overset{.}{<} y\, \varphi &:= \exists x\, (x \overset{.}{<} y \ \wedge \ \varphi) & \forall x \overset{.}{<} y\, \varphi &:= \forall x\, (x \overset{.}{<} y \ \to \ \varphi)
\end{aligned}
$$

We moreover let, for z not free in φ:

$$
\exists^{\infty} x\, \varphi := \forall z\, \exists x\, (z \overset{.}{\leq} x \ \wedge \ \varphi) \qquad \forall^{\infty} x\, \varphi := \exists z\, \forall x\, (z \overset{.}{\leq} x \ \to \ \varphi)
$$

MSO on ω-words is known to be decidable by Büchi's Theorem [Büc62].

Theorem 2.10 (Büchi [Büc62])**.** MSO *over* $\mathfrak{N}$ *is decidable.*

Following [Büc62] (but see also e.g. [PP04, VW08]), the (non-deterministic) automata method for deciding MSO proceeds by a recursive translation of MSO-formulae to *non-deterministic Büchi automata* (NBAs). An NBA is an NFA running on ω-words and for which an (infinite) run is accepting if it has infinitely many occurrences of final states.

The crux of Büchi's Theorem is the effective closure of NBAs under complement. Let us recall a few known facts on the complementation of NBAs (see e.g. [Tho97, GTW02]). First, the translation of MSO-formulae to automata is non-elementary. Second, its is known that *deterministic* Büchi automata (DBAs) are strictly less expressive than NBAs. Finally, it is known that complementation of NBAs is algorithmically hard: there is a family of languages $(\mathcal{L}_n)_{n>0}$ such that each $\mathcal{L}_n$ can be recognized by an NBA with $n+2$ states, but such that the complement of $\mathcal{L}_n$ cannot be recognized by an NBA with less than $n!$ states.

2.4. **Church's Synthesis for MSO.** Church's synthesis problem for MSO is the following. Given as input an MSO-formula $\varphi(\overline{X};\overline{Y})$ (with only free variables $\overline{X} = X_1,\ldots,X_p$ and $\overline{Y} = Y_1,\ldots,Y_q$),

(1) decide whether there exist finite-state synchronous functions $\overline{F} = F_1,\ldots,F_q$ (with each $F_i : \mathbf{2}^p \to_{\mathbf{M}} \mathbf{2}$, so that $\overline{F} : \mathbf{2}^p \to_{\mathbf{M}} \mathbf{2}^q$ collectively) such that $\mathfrak{N} \models \varphi(\overline{B};\overline{F}(\overline{B}))$ for all $\overline{B} \in (\mathbf{2}^\omega)^p \simeq (\mathbf{2}^p)^\omega$, and

(2) construct such $\overline{F}$ whenever they exist.

Given a formula $\varphi(\overline{X},\overline{Y})$ as above, we say that $\overline{F} : \mathbf{2}^p \to_{\mathbf{M}} \mathbf{2}^q$ realizes $\varphi(\overline{X};\overline{Y})$ in the sense of Church (or *Church-realizes* $\varphi(\overline{X};\overline{Y})$) when $\varphi(\overline{B},\overline{F}(\overline{B}))$ holds for all $\overline{B}$.

Example 2.11. The specification Φ of Ex. 2.1 can be officially written in the language of MSO as the following formula $\phi(X;Y)$:

$$\forall t\left(Xt \;\rightarrow\; Yt\right) \;\;\wedge\;\; \forall t\,\forall t'\bigl(\mathsf{S}(t,t') \;\rightarrow\; \neg Yt \;\rightarrow\; Yt'\bigr) \;\;\wedge\;\; \left((\exists^\infty t \,\neg Xt) \;\rightarrow\; (\exists^\infty t\,\neg Yt)\right)$$

Church's synthesis has been shown to be solvable by Büchi & Landweber [BL69], using automata on ω-words and infinite two-player games (a solution is also possible via tree automata [Rab72]): there is an algorithm which, on input $\varphi(\overline{X};\overline{Y})$, (1) decides whether a finite-state synchronous Church-realizer of $\varphi(\overline{X};\overline{Y})$ exists, and if yes (2) provides a DMM implementing it.

The standard algorithm solving Church's synthesis for MSO (see e.g. [Tho08]) proceeds via McNaughton's Theorem ([McN66], see also e.g. [PP04, Tho97]), 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. [Tho97, GTW02, PP04]). All of them can specify states which *must not* occur infinitely often in an accepting run. 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 2.12 (McNaughton [McN66])**.** *Each NBA is equivalent to a deterministic Muller automaton.*

There is a lower bound in $2^{\Omega(n\log(n))}$ on the number of states of a deterministic Muller automaton equivalent to an NBA with n states [Yan08]. The best known constructions for McNaughton's Theorem (such as *Safra's construction* or its variants) give deterministic Muller automata with $2^{O(n\log(n))}$ states from NBAs with n states.

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

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

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

Figure 3: Logical Rules of MSO and SMSO.

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

$$\frac{\overline{\varphi} \vdash \varphi}{\overline{\varphi} \vdash \varphi \vee \psi} \qquad \frac{\overline{\varphi} \vdash \psi}{\overline{\varphi} \vdash \varphi \vee \psi} \qquad \frac{\overline{\varphi} \vdash \varphi \vee \psi \qquad \overline{\varphi},\varphi \vdash \theta \qquad \overline{\varphi},\psi \vdash \theta}{\overline{\varphi} \vdash \theta}$$

$$\frac{\overline{\varphi} \vdash \varphi}{\overline{\varphi} \vdash \forall x\, \varphi} \; (x \text{ not free in } \overline{\varphi}) \qquad \frac{\overline{\varphi} \vdash \forall x\, \varphi}{\overline{\varphi} \vdash \psi[y/x]} \qquad \frac{\overline{\varphi} \vdash \varphi}{\overline{\varphi} \vdash \forall X\, \varphi} \; (X \text{ not free in } \overline{\varphi}) \qquad \frac{\overline{\varphi} \vdash \forall X\, \varphi}{\overline{\varphi} \vdash \psi[Y/X]}$$

Figure 4: Admissible Rules of MSO.

The standard solution to Church's synthesis for MSO starts by translating $\varphi(\overline{X};\overline{Y})$ to a deterministic Muller automaton, and then turns this deterministic automaton into a two-player sequential game, in which the Opponent ∀bélard plays input bit sequences in $\mathbf{2}^p$ while the Proponent ∃loïse replies with output bit sequences in $\mathbf{2}^q$, so that Proponent's strategies correspond to synchronous functions $\mathbf{2}^p \to_{\mathbf{S}} \mathbf{2}^q$. The game is equipped with an ω-regular winning condition (induced by the acceptance condition of the Muller automaton). The solution is then provided by Büchi-Landweber Theorem [BL69], which states that ω-regular games on finite graphs are effectively determined, and moreover that the winner always has a finite-state winning strategy.

Example 2.13. 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. 2.11. When translating ϕ_2 to a finite-state automaton, the positive occurrence of $(\exists^\infty t\ \neg Yt)$ can be translated to a DBA. However, the negative occurrence of $(\exists^\infty t\ \neg Xt)$ corresponds to $(\forall^\infty t\ Xt)$ and cannot be translated to a *deterministic* Büchi automaton. Even if a very simple two-states Muller automaton exists for $(\forall^\infty t\ Xt)$, McNaughton's Theorem 2.12 is in general required for Boolean combinations of $\exists^\infty t\,(-)$'s.

2.5. **An Axiomatization of** MSO. Our approach to Church's synthesis relies on the fact that the MSO-theory of $\mathfrak{N}$ can be completely axiomatized as a subsystem of second-order Peano arithmetic [Sie70] (see also [Rib12]). For the purpose of this paper, it is convenient to axiomatize MSO with the non-logical rules of Fig. 5 together with the following *comprehension* and *induction* rules:

$$\frac{\overline{\varphi} \vdash \varphi[\psi[y]/X]}{\overline{\varphi} \vdash \exists X\, \varphi} \qquad \frac{\overline{\varphi},\mathsf{Z}(z) \vdash \varphi[z/x] \qquad \overline{\varphi},\mathsf{S}(y,z),\varphi[y/x] \vdash \varphi[z/x]}{\overline{\varphi} \vdash \varphi} \qquad (2.1)$$

Equality Rules:

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

Partial Order Rules:

$$\frac{}{\overline{\varphi} \vdash x \stackrel{.}{\leq} x} \qquad \frac{\overline{\varphi} \vdash x \stackrel{.}{\leq} y \qquad \overline{\varphi} \vdash y \stackrel{.}{\leq} z}{\overline{\varphi} \vdash x \stackrel{.}{\leq} z} \qquad \frac{\overline{\varphi} \vdash x \stackrel{.}{\leq} y \qquad \overline{\varphi} \vdash y \stackrel{.}{\leq} x}{\overline{\varphi} \vdash x \doteq y}$$

Basic Z and S Rules (total injective relations):

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

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

Arithmetic Rules:

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

Figure 5: Arithmetic Rules of MSO and SMSO.

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 $(x \stackrel{.}{\in} X)[\psi[y]/X] = \psi[x/y]$. As for the logical rules of MSO, we consider the presentation of two-sorted classical logic consisting of the rules of Fig. 3 together with the following rule of *double negation elimination*:

$$\frac{\overline{\varphi} \vdash \neg\neg\varphi}{\overline{\varphi} \vdash \varphi} \tag{2.2}$$

Definition 2.14 (Deduction for MSO)**.** The deduction system of MSO is given by the rules of two-sorted classical logic (Fig. 3 and (2.2)) together with the rules of Fig. 5 and (2.1).

We write $\overline{\varphi} \vdash_{\mathsf{MSO}} \varphi$ if $\overline{\varphi} \vdash \varphi$ is provable in MSO. We also write $\mathsf{MSO} \vdash \varphi$ for $\vdash_{\mathsf{MSO}} \varphi$.

Remark 2.15. As usual with classical logic, the rules of Fig. 4 (where $\rightarrow, \vee, \forall$ are the defined connectives of Notation 2.9) are admissible in MSO.

As announced, deduction for MSO is complete w.r.t. the standard model $\mathfrak{N}$.

Theorem 2.16 (Siefkes [Sie70])**.** *For every closed formula φ, we have $\mathfrak{N} \models \varphi$ if and only if $\mathsf{MSO} \vdash \varphi$.*

Actually obtaining Thm. 2.16 from [Sie70] or [Rib12] requires some easy but tedious work. We discuss here the latter option. The difference between [Rib12] and the present system is that the axiomatization of [Rib12] is expressed in terms of the strict part of $\stackrel{.}{\leq}$ (written $\stackrel{.}{<}$, see Notation 2.9) and that comprehension is formulated with the following usual axiom scheme (where X is not free in φ):

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

We state here the properties required to bridge the gap between [Rib12] and the present axiomatization of MSO. Missing details are provided in App. A. First, the comprehension

$$
\begin{array}{rl}
(1) & \vdash \neg(x \mathrel{\dot{<}} x) \\[4pt]
(2) & x \mathrel{\dot{<}} y,\ y \mathrel{\dot{<}} z \vdash x \mathrel{\dot{<}} z \\[4pt]
(3) & \mathsf{S}(x,y),\ x \mathrel{\dot{=}} y \vdash \bot \\[4pt]
(4) & \vdash \forall x \exists y (x \mathrel{\dot{<}} y) \\[4pt]
(5) & \mathsf{S}(y,y'),\ x \mathrel{\dot{\leq}} y,\ x \mathrel{\dot{=}} y' \vdash \bot \\[4pt]
(6) & \mathsf{Z}(x) \vdash x \mathrel{\dot{\leq}} y \\[4pt]
(7) & x \mathrel{\dot{\leq}} y,\ \mathsf{Z}(y) \vdash \mathsf{Z}(x) \\[4pt]
(8) & \forall y (x \mathrel{\dot{\leq}} y) \vdash \mathsf{Z}(x) \\[4pt]
(9) & x \mathrel{\dot{<}} y,\ \mathsf{S}(x,x') \vdash x' \mathrel{\dot{\leq}} y \\[4pt]
(10) & x \mathrel{\dot{\leq}} y,\ \mathsf{S}(x,x'),\ \mathsf{S}(y,y') \vdash x' \mathrel{\dot{\leq}} y' \\[4pt]
(11) & \vdash \forall x \forall y \Big[y \mathrel{\dot{<}} x \ \longleftrightarrow\ \exists z \big(y \mathrel{\dot{\leq}} z \ \wedge\ \mathsf{S}(z,x) \big) \Big] \\[4pt]
(12) & \vdash x \mathrel{\dot{<}} y \vee x \mathrel{\dot{=}} y \vee y \mathrel{\dot{<}} x \\[4pt]
(13) & \vdash \forall x \forall y \Big[\mathsf{S}(x,y) \ \longleftrightarrow\ \big(x \mathrel{\dot{<}} y \ \wedge\ \neg \exists z (x \mathrel{\dot{<}} z \mathrel{\dot{<}} y) \big) \Big]
\end{array}
$$

Figure 6: Some Arithmetic Lemmas of MSO.

scheme of the present version of MSO directly implies (2.3), since using

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

we have

$$
\frac{\overline{\varphi} \vdash \forall x \big(\varphi[x/y] \ \longleftrightarrow\ \varphi[x/y] \big)}{\overline{\varphi} \vdash \exists X \forall x \big(X(x) \ \longleftrightarrow\ \varphi[x/y] \big)}
$$

In order to deal with the $\dot{<}$-axioms of [Rib12], we rely on a series of arithmetical lemmas of MSO displayed in Fig. 6.

Lemma 2.17. MSO *proves all the sequents of Fig. 6.*

Finally, the induction axiom of [Rib12] is the usual *strong induction* axiom:

$$
\forall X \Big[\forall x \big(\forall y (y \mathrel{\dot{<}} x \to Xy) \ \longrightarrow\ Xx \big) \ \longrightarrow\ \forall x \, Xx \Big] \tag{2.4}
$$

Lemma 2.18. MSO *proves the strong induction axiom (2.4).*

The detailed proofs of Lem. 2.17 and Lem. 2.18 are deferred to App. A.

3. SMSO: A SYNCHRONOUS INTUITIONISTIC VARIANT OF MSO

We now introduce SMSO, an intuitionistic variant of MSO 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 Church-realizers, and such proofs exist for each solvable instance of Church's synthesis (Thm. 3.7, §3.2).

As it is common with intuitionistic versions of classical systems, SMSO has the same language as MSO, and its deduction rules are based on intuitionistic predicate calculus

(Fig. 3). As expected, SMSO contains MSO via negative translation. Actually, our limited vocabulary without primitive universal quantifications allows for a Glivenko Theorem, in the sense that SMSO proves $\neg\neg\varphi$ iff MSO proves φ (Thm. 3.6, §3.1). In order for SMSO to contain a negative translation of MSO while admitting a computational interpretation in the sense of §5, one has to devise appropriate counterparts to the comprehension and induction rules of MSO (2.1):

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

(where z, y do not occur free in $\overline{\varphi}, \varphi$). First, SMSO cannot have the comprehension rule of MSO. The reason is that monadic variables are computational objects in the realizability interpretation of SMSO (§5), while the comprehension rule of MSO has instances in which the existential monadic quantification cannot be witnessed by computable functions from the parameters of $\overline{\varphi}$, ψ and φ. The situation is similar to that of higher-type intuitionistic (Heyting) arithmetic, in which predicates, represented as characteristic functions, are computational objects (see e.g. [Koh08]).[3] The usual solution in that setting is to only admit negative translations of comprehension. We take a similar approach for SMSO. In view of Glivenko's Theorem 3.6, this amounts to equip SMSO with the *negative comprehension* rule:

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

Second, for the extraction of *finite-state synchronous* functions from proofs, the induction scheme of MSO also has to be restricted. Recall the *deterministic formulae* of Fig. 2:

$$\delta, \delta' \quad ::= \quad \alpha \quad | \quad \delta \wedge \delta' \quad | \quad \neg\varphi$$

Deterministic formulae are to be interpreted by deterministic (not nec. Büchi) automata, and thus have trivial realizers in the sense of §5. As a consequence, we can trivially realize the following *deterministic induction* rule (where z, y do not occur free in $\overline{\varphi}, \delta$):

$$\frac{\overline{\varphi}, \mathsf{Z}(z) \vdash \delta[z/x] \qquad \overline{\varphi}, \mathsf{S}(y,z), \delta[y/x] \vdash \delta[z/x]}{\overline{\varphi} \vdash \delta} \tag{3.2}$$

In addition, since deterministic formulae have trivial realizers, we can safely assume in SMSO the elimination of double negation on deterministic formulae:

$$\frac{\overline{\varphi} \vdash \neg\neg\delta}{\overline{\varphi} \vdash \delta} \tag{3.3}$$

Note that (3.3) would follow, using the rules of Fig. 3, by simply assuming elimination of double negation for atomic formulae. Note also that (3.3) would follow from induction in a setting like Heyting arithmetic.

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

Definition 3.1 (Relativized and Bounded Formulae)**.**

---

[3]This contrasts with second-order logic based on Girard's System F [Gir72] (see also [GLT89]), in which second-order variables have no computational content.

(1) Given formulae φ and θ and a variable y, the *relativization of φ to $\theta[y]$* (notation $\varphi{\upharpoonright}\theta[y]$) is defined by induction on φ as usual:

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

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

where, in the clauses for $\exists$, the variables x and X are assumed not to occur free in θ.

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

Definition 3.2 (The Logic SMSO)**.** The logic SMSO has the same language as MSO. Its deduction rules are those given in Fig. 3 together with the rules of Fig. 5, the rules (3.1), (3.2), (3.3), and the following rule of *synchronous comprehension* in which $\hat{\varphi}$ is uniformly bounded by y:

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

Similarly as with MSO, we write $\overline{\varphi} \vdash_{\mathsf{SMSO}} \varphi$ if $\overline{\varphi} \vdash \varphi$ is provable in SMSO, and we write SMSO $\vdash \varphi$ for $\vdash_{\mathsf{SMSO}} \varphi$.

Remark 3.3. As usual with natural deduction systems, SMSO satisfies the substitution lemma, which gives the admissibility of the cut rule. We included that rule in SMSO because it corresponds to the composition of realizers in the realizability model, and thus has a natural computational interpretation.

Notation 3.4. In the following, we use a double horizontal line to denote admissible rules. For instance, we freely use the *weakening* rule

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

with the notation

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

Remark 3.5. Note that SMSO has a limited set of connectives. In contrast with MSO, which is based on classical logic, the derived connectives of Notation 2.9 do not define the usual corresponding intuitionistic connectives. For example, with $\psi \to \varphi = \neg(\psi \wedge \neg\varphi)$ as in Not. 2.9, while the usual $\to$-introduction rule is admissible in SMSO:

$$\frac{\dfrac{\dfrac{\overline{\varphi}, \psi \wedge \neg\varphi \vdash \psi \wedge \neg\varphi}{\overline{\varphi}, \psi \wedge \neg\varphi \vdash \psi} \quad \dfrac{\dfrac{\overline{\varphi}, \psi \vdash \varphi}{\overline{\varphi}, \psi \wedge \neg\varphi, \psi \vdash \varphi}}{\overline{\varphi}, \psi \wedge \neg\varphi \vdash \varphi} \quad \dfrac{\overline{\varphi}, \psi \wedge \neg\varphi \vdash \psi \wedge \neg\varphi}{\overline{\varphi}, \psi \wedge \neg\varphi \vdash \neg\varphi}}{\overline{\varphi}, \psi \wedge \neg\varphi \vdash \bot}}{\overline{\varphi} \vdash \psi \to \varphi}$$

the elimination rule of $\rightarrow$ is only admissible for implications with *deterministic* r.-h.s:

$$\dfrac{\dfrac{\dfrac{\overline{\varphi} \vdash \psi}{\overline{\varphi}, \neg\delta \vdash \psi} \quad \overline{\varphi}, \neg\delta \vdash \neg\delta}{\overline{\varphi}, \neg\delta \vdash \psi \wedge \neg\delta} \quad \dfrac{\overline{\varphi} \vdash \psi \rightarrow \delta}{\overline{\varphi}, \neg\delta \vdash \psi \rightarrow \delta}}{\dfrac{\dfrac{\overline{\varphi}, \neg\delta \vdash \bot}{\overline{\varphi} \vdash \neg\neg\delta}}{\overline{\varphi} \vdash \delta}}$$

On the other hand, usual $\neg$-rules are admissible in SMSO (even without using deterministic double negation elimination):

$$\dfrac{\overline{\varphi}, \varphi \vdash \neg\psi}{\overline{\varphi}, \psi \vdash \neg\varphi} \qquad \dfrac{}{\overline{\varphi}, \varphi \vdash \neg\neg\varphi} \qquad \dfrac{\overline{\varphi}, \varphi \vdash \psi}{\overline{\varphi}, \neg\psi \vdash \neg\varphi}$$

$$\dfrac{}{\overline{\varphi}, \neg\neg\neg\varphi \vdash \neg\varphi} \qquad \dfrac{\overline{\varphi}, \varphi \vdash \psi}{\overline{\varphi}, \neg\neg\varphi \vdash \neg\neg\psi} \qquad \dfrac{\overline{\varphi}, \psi \vdash \neg\varphi}{\overline{\varphi}, \neg\neg\psi \vdash \neg\varphi} \tag{3.4}$$

Indeed, the second rule of the first line follows from the first one, and the third rule is obtained from the first two ones. The rules of the second line all follow from the last two rules of the first line. Finally, the first rule of the first line is obtained as usual:

$$\dfrac{\dfrac{\dfrac{\overline{\varphi}, \varphi \vdash \neg\psi}{\overline{\varphi}, \psi, \varphi \vdash \neg\psi} \quad \overline{\varphi}, \psi, \varphi \vdash \psi}{\overline{\varphi}, \psi, \varphi \vdash \bot}}{\overline{\varphi}, \psi \vdash \neg\varphi}$$

3.1. **A Glivenko Theorem for MSO.** The limited vocabulary of MSO without primitive universal quantifications allows for a Glivenko Theorem, in the sense that SMSO proves $\neg\neg\varphi$ iff MSO proves φ. While Glivenko's Theorem is often stated for propositional logic, it also holds in presence of existential quantifications (see e.g. [Kle52, Thm. 59.(b), §81] or [Koh08, §10.1]), but does not extend to universal quantifications (see e.g. [Koh08, §11]). In particular, we would have relied on usual recursive negative translations if MSO had primitive universal quantifications.

Theorem 3.6 (Glivenko's Theorem for MSO and SMSO)**.** *If* MSO $\vdash \varphi$, *then* SMSO $\vdash \neg\neg\varphi$.

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

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

the following rule is admissible in SMSO:

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

The logical rules of MSO may be treated exactly as in the usual proof of Glivenko's Theorem. It remains to deal with the non-logical rules of MSO.

Comprehension: We have to prove that the following is admissible in SMSO:

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

But this directly follows from the negative comprehension scheme of SMSO together with the last rule of (3.4):

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

Induction: We need to show that the following is admissible in SMSO:

$$\frac{\overline{\varphi}, \mathsf{Z}(z) \vdash \neg\neg\varphi[z/x] \qquad \overline{\varphi}, \mathsf{S}(y,z), \varphi[y/x] \vdash \neg\neg\varphi[z/x]}{\overline{\varphi} \vdash \neg\neg\varphi} \qquad (z, y \text{ not free in } \overline{\varphi}, \varphi)$$

But this follows from deterministic induction together with the last rule of (3.4):

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

Arithmetic Rules (Fig. 5): All these rules can be treated the same way. We only detail the case of elimination of equality. We have to show that the following rule is admissible in SMSO:

$$\frac{\overline{\varphi} \vdash \neg\neg\varphi[x/z] \qquad \overline{\varphi} \vdash \neg\neg(x \doteq y)}{\overline{\varphi} \vdash \neg\neg\varphi[y/z]} \qquad (3.5)$$

First, note that elimination of equality in SMSO gives

$$\overline{\varphi}, \varphi[x/z], x \doteq y \vdash \varphi[y/z]$$

from which (3.4) gives

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

We then obtain the rule (3.5) by successively cutting $\neg\neg\varphi[x/z]$ and $\neg\neg(x \doteq y)$ with the corresponding premise of (3.5). $\square$

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

Theorem 3.7 (Main Theorem). *Consider a formula $\varphi(\overline{X}; \overline{Y})$ with only $\overline{X}, \overline{Y}$ free.*

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

(2) *If $\varphi(\overline{X}; \overline{Y})$ admits a (finite-state) synchronous Church-realizer, then $\exists \overline{Y} \neg\neg\varphi(\overline{X}; \overline{Y})$ is provable in SMSO.*

The correctness part (1) of Thm. 3.7 is be proved in §5 using a notion of realizability for SMSO based on automata and synchronous finite-state functions. The completeness part (2) is proved in §4.1, relying on the completeness of the axiomatization of MSO (Thm. 2.16) together with the correctness of the negative translation $\neg\neg(-)$ (Thm. 3.6).

4. On the Representation of Deterministic Mealy Machines in MSO

This section gathers several (possibly known) results related to the representation of DMMs in MSO. We begin in §4.1 with the completeness part of Thm. 3.7, which follows usual representations of automata in MSO (see e.g. [Tho97, §5.3]). In §4.2, we then recall from [Sie70, Rib12] the *Recursion Theorem*, which is a convenient tool to reason on runs of deterministic automata in MSO. In §4.3 we state a Lemma for the correctness part of Thm. 3.7, which relies on the usual translation of MSO-formulae over *finite words* to DFAs (see e.g. [Tho97, §3.1]). Finally, in §4.4 we give a possible strengthening of the synchronous comprehension rule of SMSO, based on Büchi's Theorem 2.10.

We work with the following notion of representation. Recall from §2.1 that for $k \in \mathbb{N}$, we still write k for the function from $\mathbb{N}$ to $\mathbf{2}$ which takes n to 1 iff $n = k$.

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

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

For $F : \mathbf{2}^\ell \times \mathbf{2}^p \to_\mathbf{M} \mathbf{2}$ as in Def. 4.1, we write $F : \mathbf{2}^p \to_\mathbf{M} \mathbf{2}$ (resp. $F : \mathbf{2}^\ell \to_\mathbf{M} \mathbf{2}$) in case $\ell = 0$ (resp. $p = 0$).

4.1. **Internalizing Deterministic Mealy Machines in MSO.** The completeness part (2) of Thm. 3.7 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. [Tho97, §5.3]). Let $F : \mathbf{2}^p \to_\mathbf{M} \mathbf{2}$ be induced by a DMM $\mathcal{M}$. W.l.o.g. we can assume the state set of $\mathcal{M}$ to be $\mathbf{2}^q$ for some $q \in \mathbb{N}$. The transition function ∂ of $\mathcal{M}$ is thus of the form

$$\partial \quad : \quad \mathbf{2}^q \times \mathbf{2}^p \quad \longrightarrow \quad \mathbf{2}^q \times \mathbf{2}$$

Let $\mathsf{I}[s_1, \ldots, s_q]$ be a propositional formula in the propositional variables $s_1, \ldots, s_q$ such that for $\overline{s} \in \mathbf{2}^q$, $\mathsf{I}[\overline{s}]$ holds iff $\overline{s}$ is the initial state of $\mathcal{M}$. Further, let

$$\mathsf{H}[s_1, \ldots, s_q \,, a_1, \ldots, a_p \,, b \,, s_1', \ldots, s_q']$$

be a propositional formula in the propositional variables $s_1, \ldots, s_q, a_1, \ldots, a_p, b, s_1', \ldots, s_q'$ such that for $\overline{s} \in \mathbf{2}^q$, $\overline{a} \in \mathbf{2}^p$, $\mathsf{b} \in \mathbf{2}$ and $\overline{s}' \in \mathbf{2}^q$, we have $\mathsf{H}[\overline{s}, \overline{a}, \mathsf{b}, \overline{s}']$ iff $\partial(\overline{s}, \overline{a}) = (\overline{s}', \mathsf{b})$. Then F is x-represented by the formula

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

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. $\square$

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 $\mathsf{H}[\overline{Q}(t), \overline{X}(t), Y(t), \overline{Q}(t')]$ to be of the form

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

where O codes the outputs of $\mathcal{M}$ while the D_i's represent its transition relation on states.

Example 4.4. The function induced by the DMM of Ex. 2.4.(3) (depicted in Fig. 1, right), is represented by a formula of the form (4.2) with $\overline{Q} = Q$ (since the machine has state set **2**), $\overline{X} = X$, and where $\mathsf{I}[-] := [(-) \leftrightarrow \bot]$ (since state 0 is initial) and (following Rem. 4.3)

$$\mathsf{O}[Q(t), X(t)] \quad = \quad \mathsf{D}[Q(t), X(t)] \quad = \quad (\neg Q(t) \ \vee \ [Q(t) \wedge X(t)]) \tag{4.3}$$

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

Proof of Thm. 3.7.(2). Assume that $\varphi(\overline{X}; \overline{Y})$ admits a realizer $F : \mathbf{2}^p \longrightarrow_{\mathbf{M}} \mathbf{2}^q$. Using the Cartesian structure of $\mathbf{M}$ (Prop. 2.8), we write $F = \overline{F} = F_1, \ldots, F_q$ with $F_i : \mathbf{2}^p \to_{\mathbf{M}} \mathbf{2}$. We thus have $\mathfrak{N} \models \varphi[\overline{B}/\overline{X}, \overline{F}(\overline{B})/\overline{Y}]$ for all $\overline{B} \in (\mathbf{2}^\omega)^p \simeq (\mathbf{2}^p)^\omega$. Now, by Prop. 4.2 there are uniformly bounded (deterministic) formulae $\overline{\delta} = \delta_1, \ldots, \delta_q$, with free variables among $\overline{X}, x$, and such that (4.1) holds for all $i = 1, \ldots, q$. It thus follows that $\mathfrak{N} \models \forall \overline{X} \varphi[\overline{\delta}[x]/\overline{Y}]$. Then, by completeness (Thm. 2.16) we know that $\vdash \varphi[\overline{\delta}[x]/\overline{Y}]$ is provable in MSO, and by negative translation (Thm. 3.6) we get $\mathsf{SMSO} \vdash \neg\neg\varphi[\overline{\delta}[x]/\overline{Y}]$. We can then apply ($q$ times) the synchronous comprehension scheme of SMSO and obtain $\mathsf{SMSO} \vdash \exists \overline{Y} \neg\neg\varphi(\overline{X}; \overline{Y})$. $\square$

Example 4.5. Recall the specification of Ex. 2.1 from [Tho08], represented in MSO by the formula $\phi(X; Y)$ of Ex. 2.11. Write $\phi(X; Y) = \phi_0(X, Y) \wedge \phi_1(X, Y) \wedge \phi_2(X, Y)$ where

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

Note that ϕ_0 and ϕ_1 are monotonic in Y, while ϕ_2 is anti-monotonic in Y. The formula ϕ_0 is trivially realized by the identity function $\mathbf{2} \to_{\mathbf{M}} \mathbf{2}$ (see Ex. 2.4.(1)), which is itself represented by the deterministic uniformly bounded formula $\delta_0(X, x) := (x \,\dot{\in}\, X)$. For ϕ_1 (which asks Y not to have two consecutive occurrences of 0), consider

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

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

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

The case of ϕ_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 we have $\forall^\infty t \, \delta_1[x]$ (so that Y stays constantly 1 from some time on). On the other hand, the machine of Ex. 2.4.(3) involves internal states, and can be represented using a fixpoint formula of the form (4.2). Reasoning on such formulae is easier with more advanced tools on MSO, that we provide in §4.2.

4.2. **The Recursion Theorem.** Theorem 3.7.(2) ensures that SMSO is able to handle all solvable instances of Church's synthesis, but it gives no hint on how to actually produce proofs. When reasoning on fixpoint formulae as those representing DMMs in Prop. 4.2, a crucial role is played by the *Recursion Theorem* for MSO [Sie70] (see also [Rib12]). The Recursion Theorem makes it possible to define predicates by well-founded induction w.r.t. the relation $\dot{<}$ (Notation 2.9). Given formulae $\overline{\psi} = \psi_1, \ldots, \psi_q$ and variables x and $\overline{X} = X_1, \ldots, X_q$, we say that $\overline{\psi}$ *is x-recursive in $\overline{X}$* when the following formula $\mathsf{Rec}^x_{\overline{X}}(\overline{\psi})$ holds:

$$\forall z \forall \overline{Z} \forall \overline{Z}' \left(\bigwedge_{1 \le i \le q} \forall y \dot{<} z \left(Z_i y \longleftrightarrow Z_i' y \right) \quad \longrightarrow \quad \bigwedge_{1 \le i \le q} \left(\psi_i[\overline{Z}/\overline{X}, z/x] \longleftrightarrow \psi_i[\overline{Z}'/\overline{X}, z/x] \right) \right)$$

(where $z, \overline{Z}, \overline{Z}'$ do not occur free in $\overline{\psi}$). For $\overline{\psi(\overline{X}, x)}$ x-recursive in $\overline{X}$, the Recursion Theorem says that, provably in MSO, there are unique $\overline{X}$ such that $\forall x(X_i x \longleftrightarrow \psi_i(\overline{X}, x))$ holds for all $i = 1, \ldots, q$.

Theorem 4.6 (Recursion Theorem [Sie70]). MSO *proves the following:*

$$\mathsf{Rec}^x_{\overline{X}}(\overline{\psi}), \ \bigwedge_{1 \le i \le q} \forall z \left(Z_i z \longleftrightarrow \forall \overline{X} \left[\bigwedge_{1 \le j \le q} \forall x \dot{\le} z (X_j x \leftrightarrow \psi_j) \quad \to \quad X_i z \right] \right) \ \vdash$$
$$\bigwedge_{1 \le i \le q} \forall x \left(Z_i x \longleftrightarrow \psi_i[\overline{Z}/\overline{X}] \right)$$

$$\mathsf{Rec}^x_{\overline{X}}(\overline{\psi}), \ \bigwedge_{1 \le i \le q} \forall x (Z_i x \leftrightarrow \psi_i[\overline{Z}/\overline{X}]) \wedge \forall x (Z_i' x \leftrightarrow \psi_i[\overline{Z}'/\overline{X}]) \ \vdash \ \bigwedge_{1 \le i \le q} \forall x \left(Z_i x \leftrightarrow Z_i' x \right)$$

The following examples give instances of application of the Recursion Theorem in formal reasoning on Mealy machines in MSO. The corresponding proofs in SMSO are then obtained by Thm. 3.6.

Examples 4.7.

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

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

so that $\delta(\overline{X}, x) = \forall \overline{Q} \forall Y (\theta(\overline{X}, \overline{Q}, Y, x) \to Y x)$. The Recursion Theorem implies that, provably in MSO, for all $\overline{X}$ there are unique predicates $\overline{Q}, Y$ s.t. $\forall x \, \theta(\overline{X}, \overline{Q}, Y, x)$.

Indeed, assuming I and H are as in Rem. 4.3, we have that $\theta(\overline{X}, \overline{Q}, Y, x)$ is equivalent to $\theta^o(\overline{Q}, \overline{X}, Y, x) \wedge \bigwedge_{1 \le i \le q} \theta_i(\overline{Q}, \overline{X}, Y, x)$, where

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

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

(2) The machine of Ex. 2.4.(3) is represented as in item (1) with O and D given by (4.3) (see Ex. 4.4, recalling that the machine as only two states). Hence MSO proves that for all X there are unique Q, Y such that $\forall x \, \theta(X, Q, Y, x)$. Continuing now Ex. 4.5, let

$$\delta_2(X, x) \quad := \quad \forall Q \, \forall Y \big(\theta(X, Q, Y, x) \ \to \ Y x \big)$$

It is not difficult to derive MSO $\vdash \phi_0[\delta_2[x]/Y] \wedge \phi_1[\delta_2[x]/Y]$. The case of $\phi_2[\delta_2[y]/Y]$ amounts to showing $\exists^\infty t \, (\neg Xt) \ \vdash_{\mathsf{MSO}} \ \exists^\infty t \, \exists Q \, \exists Y (\theta(X, Q, Y, t) \ \wedge \ \neg Yt)$. 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.

4.3. **From Bounded Formulae to Deterministic Mealy Machines.** We now turn to the extraction of finite-state synchronous functions from bounded formulae. This provides realizers of synchronous comprehension for Thm. 3.7.(1). We rely on the standard translation of MSO-formulae *over finite words* to DFAs (see e.g. [Tho97, §3.1]).

Lemma 4.8. *Let $\hat{\varphi}$ be a formula with free variables among $z, x_1, \ldots, x_\ell, X_1, \ldots, X_p$, and which is bounded by z. Then $\hat{\varphi}$ z-represents a finite-state synchronous $F : \mathbf{2}^\ell \times \mathbf{2}^p \to_\mathbf{M} \mathbf{2}$ induced by a DMM computable from $\hat{\varphi}$.*

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

Then, for all $n \in \mathbb{N}$, all $\overline{k} \in \mathbb{N}^\ell$ with $k_i \leq n$, and all $\overline{B} \in (\mathbf{2}^\omega)^p$, we have $\mathfrak{N} \models \psi[\overline{k}/\overline{x}, \overline{B}/\overline{X}]{\restriction}[- \overset{.}{\leq} n]$ if and only if, *in the sense of* MSO *over finite words*, the formula ψ holds in the finite word $\langle \overline{k}, \overline{B}{\restriction}(n+1)\rangle$. Let $\mathcal{A} = (Q, q^\imath, \partial, F)$ be a DFA recognizing the language of finite words satisfying ψ [Tho97, Thm. 3.1]. Consider the DMM $\mathcal{M} = (Q, q^\imath, \partial_\mathcal{M})$ with $\partial_\mathcal{M}(q, \mathtt{a}) = (q', b)$ where $q' = \partial(q, \mathtt{a})$ and $(b = 1$ iff $q' \in F)$, and let $F : \mathbf{2}^\ell \times \mathbf{2}^p \longrightarrow_\mathbf{M} \mathbf{2}$ be the function induced by $\mathcal{M}$. We then have

$$\langle \overline{k}, \overline{B}{\restriction}(n+1)\rangle \models \psi{\restriction}[- \overset{.}{\leq} n] \text{ (in the sense of MSO over finite words)}$$
$$\Longleftrightarrow \quad \mathcal{A} \text{ accepts the finite word } \langle \overline{k}, \overline{B}{\restriction}(n+1)\rangle$$
$$\Longleftrightarrow \quad F(\overline{k}, \overline{B})(n) = 1$$

$\square$

Remark 4.9. There is a well-known non-elementary lower bound for translating MSO-formulae over finite words to DFAs (see e.g. [GTW02, Chap. 13]). This lower bound also applies to the DMMs which induce synchronous functions represented by bounded formulae in the sense of Def. 4.1. Indeed, given $F : \mathbf{2}^\ell \times \mathbf{2}^p \to_\mathbf{M} \mathbf{2}$ z-represented by $\psi{\restriction}[- \overset{.}{\leq} z]$ (with z not free in ψ), for all $n \in \mathbb{N}$, all $\overline{B} \in (\mathbf{2}^\omega)^p$ and all $\overline{k} \in \mathbb{N}^\ell$ with $k_i \leq n$, we have $F(\overline{k}, \overline{B})(n) = 1$ if and only if $\langle \overline{k}, \overline{B}{\restriction}(n+1)\rangle \models \psi$ (in the sense of MSO over finite words). It follows that if F is induced by a DMM $\mathcal{M} = (Q, q^\imath, \partial)$, 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 $F(\overline{k}, \overline{B})(n) = 1$ iff $\mathcal{A}$ accepts the finite word $\langle \overline{k}, \overline{B}{\restriction}(n+1)\rangle$. Since the size of $\mathcal{A}$ is in general non-elementary in the size of ψ, it follows that the size of $\mathcal{M}$ is in general non-elementary in the size of $\psi{\restriction}[- \overset{.}{\leq} z]$.

Example 4.10. Recall the continuous but not synchronous function P of Ex. 2.4.(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 $\varphi(X, x)$ such that $\mathfrak{N} \models \varphi(B, n)$ iff $n + 1 \in B$. Note that φ is not equivalent to a bounded formula, since by Lem. 4.8 bounded formulae represent synchronous functions.

4.4. **Semantically Bounded Formulae.** The synchronous comprehension scheme of MSO is motivated by Lem. 4.8, which tells that uniformly bounded formulae induce DMMs. Recall from Def. 3.1 that a uniformly bounded formula is of the form $\psi{\restriction}[- \overset{.}{\leq} x]$ with only x as free individual variable. Uniform boundedness is a purely syntactic restriction on

comprehension, which has the advantage of being easy to check and conceptually simple to interpret in a proof relevant semantics. We present here a more semantic criterion on the formulae for which comprehension remains sound in a synchronous setting. We call a formula $\psi(\overline{X}, x)$ with only $\overline{X}, x$ free *semantically bounded* if the following closed formula $\mathsf{B}\frac{x}{\overline{X}}(\psi(\overline{X}, x))$, expressing that the truth value of $\psi(\overline{X}, n)$ only depends on the values of $\overline{X}$ up to n, holds:

$$\forall z \forall \overline{ZZ'} \left(\bigwedge_{1 \leq i \leq q} \forall y \mathbin{\dot{\leq}} z \big(Z_i y \longleftrightarrow Z_i' y \big) \quad \longrightarrow \quad \big(\psi[\overline{Z}/\overline{X}, z/x] \longleftrightarrow \psi[\overline{Z'}/\overline{X}, z/x] \big) \right)$$

We show in Thm. 4.11 below that semantically bounded MSO formulae are equivalent to uniformly bounded formulae. Since all uniformly bounded formulae are obviously semantically bounded, we have a semantic characterization of the formulae available for the synchronous comprehension scheme.

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

Note that Thm. 4.11 in particular applies if SMSO $\vdash \mathsf{B}\frac{x}{\overline{X}}(\psi(\overline{X}, x))$. Moreover, if $\psi(X, x)$ is x-recursive in X (in the sense of §4.2), then $\mathsf{B}_X^x(\psi(X, x))$ holds, but not conversely.

Theorem 4.11 makes it possible to derive realizers for additional instances of comprehension, namely for formulae which are semantically but not uniformly bounded. However, the algorithm underlying Thm. 4.11 relies on Büchi's Theorem 2.10, and computing $\hat{\varphi}$ from ψ can become quickly prohibitively expensive.

The proof of Thm. 4.11 relies on the decidability of MSO and on two preliminary lemmas. The first one is the following usual transfer property (see e.g. [Rib12]). Given a set $A \subseteq \mathcal{P}(\mathbb{N})$, write $\mathfrak{N}{\restriction}A$ for the model defined as the standard model $\mathfrak{N}$, but with individuals ranging over A rather than $\mathbb{N}$.

Lemma 4.12 (Transfer). *Let φ be a formula with free variables among $\overline{x} = x_1, \ldots, x_\ell$ and $\overline{X} = X_1, \ldots, X_p$. Furthermore, let $A \in \mathbf{2}^\omega \simeq \mathcal{P}(\mathbb{N})$ be non-empty. Then for all $a_1, \ldots, a_\ell \in A$ and all $\overline{B} \in (\mathbf{2}^\omega)^p$ we have*

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

The second result is the following Splitting Lemma 4.13, reminiscent of the composition method from a technical point of view. The point of Lem. 4.13 is, given a formula φ and a distinguished individual variable z, to express φ using an elementary combination of formulae, each local either to the initial segment $[- \mathbin{\dot{\leq}} z]$ or to the final segment $[- \mathbin{\dot{>}} z]$. Its proof is deferred to App. B. Write $\mathrm{FV}^\iota(\varphi)$ for the set of free individual variables of the formula φ.

Lemma 4.13 (Splitting). *Consider a formula ψ and some individual variable z. For every set of individual variables V with $z \in V$, one can produce a natural number N and two matching sequences of length N of left formulae $(L_j)_{j<N}$ and right formulae $(R_j)_{j<N}$ such that the following holds:*

• *For every $j < N$, $\mathrm{FV}^\iota(L_j) \subseteq \mathrm{FV}^\iota(\psi) \cap V$ and $\mathrm{FV}^\iota(R_j) \subseteq \mathrm{FV}^\iota(\psi) \setminus V$.*

- *If* $\mathrm{FV}^{\iota}(\psi) = \{\overline{x}, z, \overline{y}\}$ *with* $V \cap \mathrm{FV}^{\iota}(\psi) = \{\overline{x}, z\}$, *then for all* $n \in \mathbb{N}$, *all* $\overline{a} \leq n$ *and all* $\overline{b} > n$, *we have*

$$\mathfrak{N} \models \quad \psi[\overline{a}/\overline{x}, n/z, \overline{b}/\overline{y}] \quad \longleftrightarrow \quad \bigvee_{j < N} L_j[\overline{a}/\overline{x}, n/z] {\restriction} [- \dot{\leq} n] \ \wedge \ R_j[\overline{b}/\overline{y}] {\restriction} [- \dot{>} n]$$

We can now prove Thm. 4.11.

Proof of Thm. 4.11. We work in the standard model $\mathfrak{N}$ of MSO and obtain the result by completeness (Thm. 2.16). Using Lem. 4.13, we know that $\psi(\overline{X}, x)$ is equivalent to

$$\varphi(\overline{X}, x) \quad := \quad \bigvee_j L_j(x, \overline{X}) {\restriction} [- \dot{\leq} x] \ \wedge \ R_j(\overline{X}) {\restriction} [- \dot{>} x]$$

Then, by our assumption that $\psi(\overline{X}, x)$ (and thus $\varphi(\overline{X}, x)$) is semantically bounded, we have

$$\varphi(\overline{X}, x) \quad \longleftrightarrow \quad \varphi\big(\overline{X(-) \wedge - \dot{\leq} x}, x\big)$$
$$\longleftrightarrow \quad \bigvee_j L_j\big(x, \overline{X(-) \wedge - \dot{\leq} x}\big) {\restriction} [- \dot{\leq} x] \ \wedge \ R_j\big(\overline{X(-) \wedge - \dot{\leq} x}\big) {\restriction} [- \dot{>} x]$$

Again using Lem. 4.12, for every $j < N$ and $n \in \mathbb{N}$, $R_j(\overline{X(-) \wedge - \dot{\leq} n}) {\restriction} [- \dot{>} n]$ 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 $R'_j := R_j(\overline{\perp})$. Because R'_j is closed and $\mathfrak{N} {\restriction} [- \dot{>} n] \simeq \mathfrak{N}$, Lem. 4.12 moreover implies that

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

Since R'_j is closed, it follows from the decidability of MSO (Thm. 2.10) that we can decide whether $\mathfrak{N} \models R'_j$ or $\mathfrak{N} \models \neg R'_j$. Define accordingly closed formulae R''_j:

$$R''_j \quad := \quad \begin{cases} \top & \text{if } \mathfrak{N} \models R'_j \\ \perp & \text{if } \mathfrak{N} \models \neg R'_j \end{cases}$$

Notice in particular that, contrary to the R'_j, the R''_j are invariant under relativization, i.e., the formulae R''_j and $R''_j {\restriction} [- \dot{\leq} x]$ are syntactically equal. It thus follows that our initial ψ is equivalent to the following formula $\hat{\varphi}$, which is effectively computable from ψ:

$$\hat{\varphi}(\overline{X}, x) \quad := \quad \bigvee_j L_j(x, \overline{X}) {\restriction} [- \dot{\leq} x] \wedge R''_j$$

$\hat{\varphi}$ is uniformly bounded since it is syntactically equal to $\left(\bigvee_j L_j(x, \overline{X}) \wedge R''_j \right) {\restriction} [- \dot{\leq} x]$. $\quad\square$

5. The Realizability Interpretation of SMSO

We now present our realizability model for SMSO, and use it to prove Thm. 3.7.(1). This realizability interpretation bears some similarities with usual realizability constructions for the Curry-Howard correspondence (see e.g. [SU06, Koh08]). For instance, as in the usual setting, a realizer of a formula $\varphi_1 \wedge \varphi_2$ is a pair $\langle R_1, R_2 \rangle$ of a realizer of R_1 of φ_1 and a realizer R_2 of φ_2. Similarly, a realizer of $\exists X \varphi(X)$ is a pair $\langle B, R \rangle$ of an ω-word $B \in \mathbf{2}^{\omega}$ and a realizer R of $\varphi(B)$. However, our construction departs from the standard one on negation (for which we use McNaughton's Theorem 2.12), and for the fact that there is no primitive notion of implication in SMSO. In particular, in contrast with the usual settings, our notion of realizability for sequents of the form $\psi \vdash \varphi$ (see Thm. 5.10 and Def. 5.2) is not based on a notion of implication internal to the logic under consideration.

Our approach to Church's synthesis via realizability uses automata in two different ways. First, from a *proof* $\mathscr{D}$ in SMSO of an existential formula $\exists \overline{Y} \varphi(\overline{X}; \overline{Y})$, one can compute a finite-state synchronous Church-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 during the extraction procedure.

5.1. **Uniform Automata.** The adequacy of realizability relies on the notion of *uniform automata* (adapted from [Rib16]). 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 for a simple inheritance of the Cartesian structure of synchronous functions (Prop. 2.8), and thus to interpret the strictly positive existentials of SMSO similarly as usual (weak) sums of type theory. In particular, the set of moves $M(\mathcal{A})$ of an automaton $\mathcal{A}$ interpreting a formula φ exhibits the strictly positive existentials of φ as $M(\mathcal{A}) = M(\varphi)$ where

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

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

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

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

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

A *run* of $\mathcal{A}$ on an ω-word $B \in \Sigma^\omega$ is an ω-word $R \in M(\mathcal{A})^\omega$. We say that R is *accepting* (notation $R \Vdash \mathcal{A}(B)$) if $(q_k)_{k \in \mathbb{N}} \in \Omega_\mathcal{A}$ for the sequence of states $(q_k)_{k \in \mathbb{N}}$ defined as $q_0 := q_\mathcal{A}^\imath$ 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 $\mathcal{L}(\mathcal{A})$, the *language* of $\mathcal{A}$, be the set of ω-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 are used in our realizability interpretation of SMSO. First, by adapting to our context usual constructions on automata (§5.2), to each formula φ with free variables among (say) $\overline{X} = X_1, \ldots, X_p$, we associate a uniform automaton $[\![\varphi]\!]$ over $\mathbf{2}^p$ (Fig. 7). Then, from an SMSO-derivation $\mathscr{D}$ of a sequent (say) $\varphi \vdash \psi$, with free variables among $\overline{X}$ as above, we extract a finite-state synchronous function $F_\mathscr{D} : \mathbf{2}^p \times M([\![\varphi]\!]) \longrightarrow_\mathbf{M} M([\![\psi]\!])$ 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 $F_\mathscr{D}$ is of the form $\langle C, G \rangle$ with C and G finite-state synchronous functions $C : \mathbf{2}^p \longrightarrow_\mathbf{M} \mathbf{2}$ and $G : \mathbf{2}^p \longrightarrow_\mathbf{M} M(\phi)$ such that $G(\overline{B}) \Vdash [\![\phi]\!](\overline{B}, C(\overline{B}))$ for all $\overline{B}$. This motivates the following notion.

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

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:

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

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 $R \Vdash \mathcal{B}(B)$ for some $R \in M(\mathcal{B})^\omega$. Then by definition of $\mathcal{B} \Vdash F : \mathcal{A}$, we have $F(B, R) \Vdash \mathcal{A}(B)$ and thus $B \in \mathcal{L}(\mathcal{A})$. □

(2) One could also consider the category AUT_Σ defined as Aut_Σ, but with maps not required to be finite-state. All statements of §5 hold for AUT_Σ, but for Cor. 5.11, which would lead to non necessarily finite-state realizers and would not give Thm. 3.7.(1).

(3) Uniform automata are a variation of usual automata on ω-words, which is convenient for our purposes, namely the adequacy of our realizability interpretation. Hence, while it would have been possible to define uniform automata with any of the usual acceptance conditions (see e.g. [Tho97]), we lose nothing by assuming their acceptance conditions to be given by arbitrary ω-regular sets.

(4) Given automata $\mathcal{A}, \mathcal{B} : \Sigma$, checking the existence of a realizer $\mathcal{A} \Vdash F : \mathcal{B}$ can be reduced (e.g. using the tools of [PR18, Rib16]) to checking the existence of a winning strategy for the Proponent ($\exists$loïse) in an ω-regular game on a finite graph, which can in turn be decided by the Büchi-Landweber Theorem [BL69].

5.2. Constructions on Automata. We gather here constructions on uniform automata that we need to interpret 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 φ with free variables among $\overline{X}$, so that $\overline{B} \in \mathcal{L}(\mathcal{A})$ iff $\mathfrak{N} \models \varphi[\overline{B}/\overline{X}]$. Then φ is also a formula with free variables among $\overline{X}, \overline{Y}$, and we have $\overline{BB'} \in \mathcal{L}(\mathcal{A}[\pi])$ 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 Aut_Σ. This gives the interpretation of conjunctions.

Proposition 5.6. *For each* Σ, *the category* Aut_Σ *has finite products. Its terminal object is the automaton* $\mathbf{I} = (\mathbf{1}, \bullet, \mathbf{1}, \partial_{\mathbf{I}}, \mathbf{1}^\omega)$, *where* $\partial_{\mathbf{I}}(-, -, -) = \bullet$. *Binary products are given by*

$$\begin{aligned} \mathcal{A} \times \mathcal{B} &:= (Q_{\mathcal{A}} \times Q_{\mathcal{B}}, (q^\iota_{\mathcal{A}}, q^\iota_{\mathcal{B}}), M(\mathcal{A}) \times M(\mathcal{B}), \partial, \Omega) \\ \text{where} \quad \partial((q_{\mathcal{A}}, q_{\mathcal{B}}), \mathtt{a}, (u, v)) &:= (\partial_{\mathcal{A}}(q_{\mathcal{A}}, \mathtt{a}, u), \partial_{\mathcal{B}}(q_{\mathcal{B}}, \mathtt{a}, v)) \end{aligned}$$

and where $(q_n, q'_n)_n \in \Omega$ *iff* $((q_n)_n \in \Omega_{\mathcal{A}}$ *and* $(q'_n)_n \in \Omega_{\mathcal{B}})$. *Note that* Ω *is* ω-*regular since* $\Omega_{\mathcal{A}}$ *and* $\Omega_{\mathcal{B}}$ *are* ω-*regular. 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 $\langle R_1, R_2 \rangle \Vdash (\mathcal{A}_1 \times \mathcal{A}_2)(B)$. □

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

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

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* **M**-*pairs $\langle C, G \rangle$ of finite-state 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 $B \in \Sigma^\omega$ and all $R \Vdash \mathcal{B}(B)$.

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

$$C : \Sigma \times M(\mathcal{B}) \longrightarrow_\mathbf{M} \Gamma \qquad\qquad G : \Sigma \times M(\mathcal{B}) \longrightarrow_\mathbf{M} M(\mathcal{A}) \qquad (5.3)$$

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 $B \in \Sigma^\omega$ and all $R \Vdash \mathcal{B}(B)$, then we have $\mathcal{B} \Vdash \langle C, G \rangle : \exists_\Gamma \mathcal{A}$. $\qquad\square$

The negation $\neg(-)$ on formulae is interpreted by an operation $\sim(-)$ on uniform automata which involves McNaughton's Theorem 2.12.

Proposition 5.8. *Given a uniform automaton $\mathcal{A} : \Sigma$, there is a uniform deterministic automaton $\sim\mathcal{A} : \Sigma$ 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 $\mathcal{S}$ over $\Sigma \times U$ with the same states as $\mathcal{A}$ and with transition function $\partial_\mathcal{S}$ defined as $\partial_\mathcal{S}(q, (\mathtt{a}, u)) := \partial_\mathcal{A}(q, \mathtt{a}, u)$. Then $R \Vdash \mathcal{A}(B)$ iff $\mathcal{S}$ accepts $\langle B, R \rangle$. Since $\Omega_\mathcal{A}$ is ω-regular, it is recognized by a non-deterministic Büchi automaton $\mathcal{C}$ over $Q_\mathcal{A}$. We then obtain a non-deterministic 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 Σ (see e.g. [Tho97]). By McNaughton's Theorem 2.12, $\tilde{\exists}_U \mathcal{B}$ is equivalent to a *deterministic* Muller automaton $\mathcal{D}$ over Σ. Then we let $\sim\mathcal{A}$ be the deterministic uniform automaton defined as $\mathcal{D}$ but with $\Omega_{\sim\mathcal{A}}$ the ω-regular set generated by the Muller condition $S \in \mathcal{T}$ iff $S \notin \mathcal{T}_\mathcal{D}$ (see e.g. [PP04, Thm. I.7.1 & Prop. I.7.4]). $\qquad\square$

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

(1) To each formula φ we associate a uniform automaton $\llbracket \varphi \rrbracket$.
(2) To each derivation $\mathscr{D}$ of a (closed) sequent $\varphi_1, \ldots, \varphi_n \vdash \varphi$ in SMSO, we associate a finite-state synchronous $F_\mathscr{D}$ such that $\llbracket \varphi_1 \rrbracket \times \cdots \times \llbracket \varphi_n \rrbracket \Vdash F_\mathscr{D} : \llbracket \varphi \rrbracket$.

We first discuss step (1). Consider a formula φ with free variables among $\overline{x} = x_1, \ldots, x_\ell$ and $\overline{X} = X_1, \ldots, X_p$. Its interpretation is a uniform automaton $\llbracket \varphi \rrbracket_{\overline{x}, \overline{X}}$ over $\mathbf{2}^\ell \times \mathbf{2}^p$, defined by induction on φ, and such that $\llbracket \delta \rrbracket_{\overline{x}, \overline{X}}$ is deterministic for a deterministic δ. We thus have to devise a deterministic uniform automaton $\mathcal{A}(\alpha)$ for each atomic formula α of SMSO. The definitions of the $\mathcal{A}(\alpha)$'s are easy and follow usual constructions (see e.g. [Tho97]). They are deferred to App. C. Moreover, in order to handle individual variables, the interpretation also uses a deterministic uniform automaton $\mathsf{Sing} : \mathbf{2}$ accepting the language of ω-words

$$\llbracket \alpha \rrbracket_{\overline{x},\overline{X}} \; := \; \mathcal{A}(\alpha)[\pi] \qquad \llbracket \neg \psi \rrbracket_{\overline{x},\overline{X}} \; := \; \sim \llbracket \psi \rrbracket_{\overline{x},\overline{X}} \qquad \llbracket \exists X \, \psi \rrbracket_{\overline{x},\overline{X}} \; := \; \exists_{\mathbf{2}}(\llbracket \psi \rrbracket_{\overline{x},\overline{X},X})$$

$$\llbracket \psi_1 \wedge \psi_2 \rrbracket_{\overline{x},\overline{X}} \; := \; \llbracket \psi_1 \rrbracket_{\overline{x},\overline{X}} \times \llbracket \psi_2 \rrbracket_{\overline{x},\overline{X}} \qquad \llbracket \exists x \, \psi \rrbracket_{\overline{x},\overline{X}} \; := \; \exists_{\mathbf{2}}(\mathsf{Sing}[\pi'] \times \llbracket \psi \rrbracket_{\overline{x},x,\overline{X}}[\sigma])$$

Figure 7: Interpretation of SMSO-Formulae as Uniform Automata.

$B \in \mathbf{2}^{\omega} \simeq \mathcal{P}(\mathbb{N})$ such that B is a singleton. App. C also presents a possible definition for Sing. The interpretation $\llbracket \varphi \rrbracket_{\overline{x},\overline{X}}$ is defined in Fig. 7, where π, π' are suitable projections and σ is a suitable permutation. We write $\llbracket \varphi \rrbracket$ when $\overline{x}, \overline{X}$ are irrelevant or understood from the context. Note that the set of moves $M(\varphi)$ of $\llbracket \varphi \rrbracket$ indeed satisfies (5.1), so in particular $\llbracket \delta \rrbracket$ is indeed deterministic for a deterministic δ.

As expected, the interpretation $\llbracket - \rrbracket$ is correct in the following sense. For $k \in \mathbb{N}$, we keep on writing k for the function from $\mathbb{N}$ to $\mathbf{2}$ which takes n to 1 iff $n = k$.

Proposition 5.9. *Given a formula φ with free variables among $\overline{x} = x_1, \ldots, x_\ell$ and $\overline{X} = X_1, \ldots, X_p$, for all $\overline{k} \in \mathbb{N}^\ell$ and all $\overline{B} \in (\mathbf{2}^{\omega})^p \simeq (\mathbf{2}^p)^{\omega}$ we have $(\overline{k}, \overline{B}) \in \mathcal{L}(\llbracket \varphi \rrbracket_{\overline{x},\overline{X}})$ iff $\mathfrak{N} \models \varphi[\overline{k}/\overline{x}, \overline{B}/\overline{X}]$.*

We now turn to step (2). Let $\varphi_1, \ldots, \varphi_n, \varphi$ be formulae and consider variables $\overline{x} = x_1, \ldots, x_\ell$ and $\overline{X} = X_1, \ldots, X_p$ containing all the free variables of $\varphi_1, \ldots, \varphi_n, \varphi$. Then we say that a synchronous function

$$F \quad : \quad \mathbf{2}^\ell \times \mathbf{2}^p \times \mathbf{1}^\ell \times M(\varphi_1) \times \cdots \times M(\varphi_n) \quad \longrightarrow_{\mathbf{M}} \quad M(\varphi)$$

$\overline{x}, \overline{X}$*-realizes* the sequent $\varphi_1, \ldots, \varphi_n \vdash \varphi$ (notation $\varphi_1, \ldots, \varphi_n \Vdash_{\overline{x},\overline{X}} F : \varphi$ or $\overline{\varphi} \Vdash_{\overline{x},\overline{X}} F : \varphi$) if

$$\mathsf{Sing}^\ell[\overline{\pi}] \times \llbracket \varphi_1 \rrbracket_{\overline{x},\overline{X}} \times \cdots \times \llbracket \varphi_n \rrbracket_{\overline{x},\overline{X}} \quad \Vdash \quad F \quad : \quad \llbracket \varphi \rrbracket_{\overline{x},\overline{X}}$$

where $\overline{\pi}$ are suitable projections.

Theorem 5.10 (Adequacy). *Let $\overline{\varphi}, \varphi$ be formulae with variables among $\overline{x}, \overline{X}$. From an SMSO-derivation $\mathscr{D}$ of $\overline{\varphi} \vdash \varphi$, one can compute an $\mathbf{M}$-morphism $F_{\mathscr{D}}$ s.t. $\overline{\varphi} \Vdash_{\overline{x},\overline{X}} F_{\mathscr{D}} : \varphi$.*

Adequacy of realizability, together with Prop. 5.7, directly gives Theorem 3.7.(1).

Corollary 5.11 (Thm. 3.7.(1)). *Consider a formula $\varphi(\overline{X}; \overline{Y})$ with only $\overline{X}, \overline{Y}$ free, where $\overline{X} = X_1, \ldots, X_p$ and $\overline{Y} = Y_1, \ldots, Y_q$. Given a derivation $\mathscr{D}$ in SMSO of $\vdash \exists \overline{Y} \, \varphi(\overline{X}; \overline{Y})$, we have $F_{\mathscr{D}} \simeq \langle \overline{C}, G \rangle$ where $\overline{C} = C_1, \ldots, C_q$ with $C_i : \mathbf{2}^p \longrightarrow_{\mathbf{M}} \mathbf{2}$ and $\mathfrak{N} \models \varphi(\overline{B}, \overline{C}(\overline{B}))$ for all $\overline{B} \in (\mathbf{2}^{\omega})^p \simeq (\mathbf{2}^p)^{\omega}$.*

The proof of Thm. 5.10 goes by induction on derivations. Most of the rules of SMSO are straightforward, except the synchronous comprehension rule, that we discuss first. Adequacy for synchronous comprehension follows from the existence of finite-state characteristic functions for bounded formulae (Lem. 4.8) and from the following lemmas, which allow us, given a synchronous function C y-represented by $\hat{\varphi}$, to lift a realizer of $\psi[\hat{\varphi}[y]/Y]$ to a realizer of $\exists Y \psi$.

Lemma 5.12 (Substitution Lemma for Synchronous Comprehension). *Let $\overline{x} = x_1, \ldots, x_\ell$ and $\overline{X} = X_1, \ldots, X_p$. Let $\hat{\varphi}$ be a formula with free variables among $y, \overline{X}$, and which y-represents $C : \mathbf{2}^p \longrightarrow_{\mathbf{M}} \mathbf{2}$. Then for every formula ψ with free variables among $\overline{x}, \overline{X}, Y$, for all $\overline{k} \in \mathbb{N}^\ell$ and all $\overline{B} \in (\mathbf{2}^{\omega})^p \simeq (\mathbf{2}^p)^{\omega}$ we have*

$$(\overline{k}, \overline{B}) \in \mathcal{L}(\llbracket \psi[\hat{\varphi}[y]/Y] \rrbracket_{\overline{x},\overline{X}}) \qquad \Longleftrightarrow \qquad (\overline{k}, \overline{B}, C(\overline{B})) \in \mathcal{L}(\llbracket \psi \rrbracket_{\overline{x},\overline{X}Y})$$

Proof. By induction on ψ.

- If ψ is an atomic formula not of the form $(x \mathbin{\dot{\in}} Y)$, then $\psi[\hat{\varphi}[y]/Y] = \psi$ and the result is trivial.
- If ψ is of the form $(x_i \mathbin{\dot{\in}} Y)$, then $\psi[\hat{\varphi}[y]/Y] = \hat{\varphi}[x_i/y]$. Since $\hat{\varphi}$ y-represents C, by (4.1) (Def. 4.1), for all $\overline{k} \in \mathbb{N}^\ell$ and all $\overline{B} \in (\mathbf{2}^\omega)^p$ we have

$$C(\overline{B})(k_i) = 1 \qquad \text{iff} \qquad \mathfrak{N} \models \hat{\varphi}[k_i/z, \overline{B}/\overline{X}]$$

  that is

$$\mathfrak{N} \models k_i \mathbin{\dot{\in}} C(\overline{B}) \qquad \text{iff} \qquad \mathfrak{N} \models \hat{\varphi}[k_i/z, \overline{B}/\overline{X}]$$

  Then we are done since it follows from Prop. 5.9 that

$$\mathfrak{N} \models \hat{\varphi}[k_i/z, \overline{B}/\overline{X}] \qquad \text{iff} \qquad (\overline{k}, \overline{B}) \in \mathcal{L}([\![\hat{\varphi}[x_i/y]]\!]_{\overline{x}, \overline{X}})$$

- If $\psi = \psi_1 \wedge \psi_2$, then by Prop. 5.6 we have

$$\mathcal{L}\big([\![(\psi_1 \wedge \psi_2)[\hat{\varphi}[y]/Y]]\!]\big) \quad = \quad \mathcal{L}\big([\![\psi_1[\hat{\varphi}[y]/Y]]\!]\big) \ \cap \ \mathcal{L}\big([\![\psi_2[\hat{\varphi}[y]/Y]]\!]\big)$$
$$\text{and} \qquad \mathcal{L}\big([\![\psi_1 \wedge \psi_2]\!]\big) \quad = \quad \mathcal{L}\big([\![\psi_1]\!]\big) \ \cap \ \mathcal{L}\big([\![\psi_2]\!]\big)$$

  and we conclude by induction hypothesis.
- The cases of ψ of the form $\exists X \, \varphi$ or $\exists x \, \varphi$ are similar, using Prop. 5.7 instead of Prop. 5.6.
- The case of ψ of the form $\neg \varphi$ follows from Prop. 5.8 and the induction hypothesis. $\square$

Lemma 5.13 (Lifting Lemma for Synchronous Comprehension). *Let $\overline{x} = x_1, \ldots, x_\ell$ and $\overline{X} = X_1, \ldots, X_p$. Let $\hat{\varphi}$ be a formula with free variables among $y, \overline{X}$, and which y-represents $C : \mathbf{2}^p \longrightarrow_{\mathbf{M}} \mathbf{2}$. Then for every formula ψ with free variables among $\overline{x}, \overline{X}, Y$, there is a finite-state synchronous function*

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

such that for all $\overline{k} \in \mathbb{N}^\ell$, all $\overline{B} \in (\mathbf{2}^\omega)^p$ and all $R \in M(\psi[\hat{\varphi}[y]/Y])^\omega$, we have

$$R \Vdash [\![\psi[\hat{\varphi}[y]/Y]]\!]_{\overline{x}, \overline{X}}(\overline{k}, \overline{B}) \qquad \Longrightarrow \qquad H(R) \Vdash [\![\psi]\!]_{\overline{x}, \overline{X}, Y}(\overline{k}, \overline{B}, C(\overline{B})) \tag{5.4}$$

Proof. By induction on ψ.

- If ψ is an atomic formula not of the form $(x \mathbin{\dot{\in}} Y)$, then $\psi[\hat{\varphi}[y]/Y] = \psi$. So we take the identity for H and the result trivially follows.
- If ψ is of the form $(x_i \mathbin{\dot{\in}} Y)$, then $\psi[\hat{\varphi}[y]/Y] = \hat{\varphi}[x_i/y]$. Since ψ is deterministic, we can take for H the unique map $M(\hat{\varphi}[x_i/y]) \to_{\mathbf{M}} M(x_i \mathbin{\dot{\in}} Y) = \mathbf{1}$, and the result follows from Lem. 5.12.
- If ψ 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 is the identity. We then conclude by Lem. 5.12. $\square$

Adequacy for synchronous comprehension follows easily.

Lemma 5.14 (Adequacy of Synchronous Comprehension). *Let ψ with free variables among $\overline{x}, \overline{X}, Y$ and let $\hat{\varphi}$ be a formula with free variables among $y, \overline{X}$ and which is uniformly bounded by y. Then there is a finite-state realizer $\psi[\hat{\varphi}[y]/Y] \Vdash_{\overline{x}, \overline{X}} F : \exists Y \, \psi$, effectively computable from ψ and φ.*

Proof. Let C y-represented by $\hat{\varphi}$ be given by Lem. 4.8, and let H satisfying (5.4) be given by Lem. 5.13. It then directly follows from Prop. 5.7 and Lem. 5.13 that $\psi[\hat{\varphi}[y]/Y] \Vdash_{\overline{x}, \overline{X}}$ $\langle C \circ [\pi], H \circ [\pi'] \rangle : \exists Y \, \psi$, where π, π' are suitable projections. $\square$

We can finally prove of Thm. 5.10.

Proof of Thm. 5.10. The proof is by induction on derivations. Note that if $\overline{\varphi} \vdash_{\mathsf{SMSO}} \varphi$, then the universal closure of the implication $\wedge\overline{\varphi} \to \varphi$ holds in the standard model $\mathfrak{N}$. In particular, for all rules whose conclusion is of the form $\overline{\varphi} \vdash \delta$ with δ deterministic, it follows from Prop. 5.9 and (5.1) that the unique **M**-map with codomain $M(\delta) \simeq \mathbf{1}$ (and with appropriate domain) is a realizer. This handles the rules of *negative comprehension* (3.1), *deterministic induction* (3.2) and of elimination of double negation on deterministic formulae (3.3). This also handles all the rules of Fig. 5, excepted the rules of elimination of equality

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

as well as the rules $\overline{\overline{\varphi} \vdash \exists y\, \mathsf{Z}(y)}$ and $\overline{\overline{\varphi} \vdash \exists y\, \mathsf{S}(x,y)}$. For the latter, we use the DMM depicted in Fig. 1 (left) (Ex. 2.4.(2)) together with the fact that $\mathsf{S}(-,-)$ is deterministic. The case of the former is similar and simpler. As for elimination of equality, we take as realizer of the conclusion the realizer of the left premise. This realizer is trivially correct if there is no realizer of the assumptions $\overline{\varphi}$. Otherwise the result follows from Prop. 5.9 since the right premise ensures that the individual variables y and z are interpreted by the same natural number. Adequacy for synchronous comprehension is given by Lem. 5.14. It remains to deal with the rules of Fig. 3. The first two rules follow from the fact that each Aut_Σ is a category with finite products (Prop. 5.6). The rules for $\neg/\bot$ are trivial since their conclusions are of the form $\overline{\varphi} \vdash \delta$. The rules for conjunction follow from Prop. 5.6 and those for existential quantifications follow from Prop. 5.7. □

6. Indexed Structure on Automata

In §5 we have defined one category Aut_Σ for each alphabet Σ. These categories are actually related by *substitution functors* arising from **M**-morphisms, inducing an *indexed* (or *fibred*) structure. Substitution functors are a basic notion of categorical logic, which allows for categorical axiomatizations of quantifications. We refer to e.g. [Jac01, Chap. 1] for background.

We present here the fibred structure of the categories $\mathsf{Aut}_{(-)}$ and show 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 Fig. 3 that do not mention negation ($\neg$) nor falsity ($\bot$). 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 manysorted individual language. The categorical semantics of existential quantifications is given by an adjunction

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

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 occurs free in φ. 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 φ and ψ should be allowed to contain free variables distinct from x.

The idea underlying the general method (see e.g. [Jac01] 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 $(t_1, \ldots, t_n)$ of terms t_i of sort o_i whose free variables are among $x_{\iota_1}, \ldots, x_{\iota_m}$, with x_{ι_j} of sort ι_j. Then, for each object $\iota = \iota_1 \times \cdots \times \iota_m$ of $\mathbb{B}$, one devises a category $\mathbb{E}_\iota$ whose objects represent formulae with free variables among $x_{\iota_1}, \ldots, x_{\iota_m}$, and whose morphisms interpret proofs. Furthermore, $\mathbb{B}$-morphisms

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

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 φ whose free variables are among $y_{o_1}, \ldots, y_{o_n}$ to (the interpretation of) the formula $\varphi[t_1/y_{o_1}, \ldots, t_n/y_{o_n}]$ with free variables among $x_{\iota_1}, \ldots, x_{\iota_m}$. Its action on the morphisms of $\mathbb{E}_{o_1 \times \cdots \times o_n}$ allows us to interpret the *substitution rule*

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

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 us 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_ι (but with no actual occurrence of x_ι). Then the proper formulation of (6.1) is that existential quantification over x_ι is a functor

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

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

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

(where x_ι does not occur free in ψ since ψ 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.

6.2. **Substitution.** So far, for each alphabet Σ we have defined a category Aut_Σ of uniform automata over Σ. Following §6.1, different categories Aut_Σ, Aut_Γ can be related by means of **M**-morphisms $F : \Sigma \to \Gamma$. This relies on a very simple *substitution operation* on automata, generalizing the substitution operation presented in Prop. 5.4.

Definition 6.1 (Substitution). Given a DMM $\mathcal{M} : \Sigma \to \Gamma$ as in Def. 2.2 and an automaton $\mathcal{A} : \Gamma$ as in Def. 5.1, the automaton $\mathcal{A}[\mathcal{M}] : \Sigma$ is defined as follows:

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

where

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

is defined as

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

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 Γ to an automaton over Σ. Substitutions of the form $\mathcal{A}[\mathcal{M}]$ can be seen as generalizations of the substitutions presented in Prop. 5.4: Given a function $\mathsf{f} : \Sigma \to \Gamma$, the automaton $\mathcal{A}[\mathsf{f}]$ of Prop. 5.4 is isomorphic (in Aut_Σ) to the automaton $\mathcal{A}[\mathcal{M}_\mathsf{f}]$ obtained by applying Def. 6.1 to the one-state DMM inducing the **M**-morphism $[\mathsf{f}] : \Sigma \to_\mathbf{M} \Gamma$ of Rem. 2.7.

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

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

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

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 Aut_Γ has just been defined. Given a morphism $\mathcal{A} \Vdash F : \mathcal{B}$ of Aut_Γ, the morphism $\mathcal{A}[\mathcal{M}] \Vdash F[\mathcal{M}] : \mathcal{B}[\mathcal{M}]$ is the finite-state synchronous function

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

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

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

We are now going to see that $\exists_{(-)}$ is an existential quantification in the usual categorical sense of *simple coproducts* (see e.g. [Jac01, 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 Σ and Γ. According to [ML98, 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 Aut_Σ-morphism

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

such that we have

Note that $\eta_{\mathcal{A}}$ must be an **M**-morphism

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

We let $\eta_{\mathcal{A}}$ be the **M**-morphism induced by the usual 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

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

$\square$

The Beck-Chevalley condition of [Jac01, Def. 1.9.1] asks for the following isomorphism in Aut_Δ, 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 Aut_Σ has Cartesian products, which interpret conjunction, a necessary feature to interpret a sequent as a morphism from the conjunct of its premises to its conclusion. In the setting of categorical logic, it remains to be shown that these products are *fibred* in the sense of [Jac01, Def. 1.8.1], i.e. that they are preserved by substitution.

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

Proof. The isomorphism trivially follows from the fact that

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

$\square$

6.5. **Indexed Structure.** Thanks to the substitution operation discussed in §6.2, each **M**-morphism $F : \Sigma \to \Gamma$ induces a functor $(-)[\mathcal{M}_F] : \mathsf{Aut}_\Gamma \to \mathsf{Aut}_\Sigma$, where $\mathcal{M}_F$ is a *chosen* DMM 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 Σ to categories Aut_Σ, and **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 **M**-maps $G : \Delta \to \Gamma$ and $F : \Gamma \to \Sigma$ we should have

$$\mathcal{A} \quad = \quad \mathcal{A}[\mathcal{M}_{\mathrm{Id}_\Sigma}] \qquad \text{and} \qquad (\mathcal{A}[\mathcal{M}_F])[\mathcal{M}_G] \quad = \quad \mathcal{A}[\mathcal{M}_{F \circ G}] \tag{6.2}$$

But we see no reason for this to be possible. In particular there is no reason for the DMM $\mathcal{M}_{F \circ G}$ 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 (6.2) 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. [Jac01, Def. 1.4.4]). The required natural isomorphisms have the form

$$\begin{aligned} \eta_\Sigma \quad &: \quad \mathrm{Id}_{\mathsf{Aut}_\Sigma} \quad \overset{\simeq}{\longrightarrow} \quad (-)[\mathcal{M}_{\mathrm{Id}_\Sigma}] \\ \mu_{G,F} \quad &: \quad (-)[\mathcal{M}_F][\mathcal{M}_G] \quad \overset{\simeq}{\longrightarrow} \quad (-)[\mathcal{M}_{F \circ G}] \end{aligned} \tag{6.3}$$

Since $\mathcal{A}$ and $\mathcal{A}[\mathcal{M}]$ have the same moves, we can take for each components of η_Σ 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 **M**-morphism $F : \Sigma \to_{\mathbf{M}} \Gamma$ a chosen DMM $\mathcal{M}_F$ inducing F. For each $\mathcal{A} : \Sigma$, and each **M**-morphisms $G : \Delta \to_{\mathbf{M}} \Gamma$ and $F : \Gamma \to_{\mathbf{M}} \Sigma$, we let

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

be given by

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

The following says that the coherence conditions required for structure maps of pseudofunctors (see e.g. [Jac01, Def. 1.4.4]) are met by $\eta_{\Sigma,\mathcal{A}}$ and $\mu_{F,G,\mathcal{A}}$. The proof is trivial.

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

Proposition 6.5. *The morphisms η_Σ and $\mu_{F,G}$ defined above are natural isomorphisms as in (6.3). Moreover, for each automaton $\mathcal{A} : \Sigma$ and each $\mathbf{M}$-maps F, G, H of appropriate domains and codomains, the two diagrams of Fig. 8 commute.*

The assignment $(-)^\star : \mathbf{M}^{\mathrm{op}} \to \mathbf{Cat}$ taking the alphabet Σ to the category Aut_Σ 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 SMSO for MSO on ω-words, and we demonstrated that our approach is sound and complete, in the sense of Thm. 3.7. As it stands, this approach must still pay the price of the non-elementary lower-bound for the translation of MSO formulae over finite words to DFAs (see Rem. 4.9) and the system 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 (S)MSO, with one function symbol for each Mealy machine together with defining axioms of the form (4.2). In particular, this could help mitigate Rem. 4.9 by giving the possibility, in the synchronous comprehension scheme of SMSO, to give a term for a Mealy machine rather than the 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 [Rib16], 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 [Rib16] we expect similar proof-theoretic properties as with the usual *Dialectica* interpretation (see e.g. [Koh08]), such as realizers of linear Markov rules and choices schemes. Also, having primitive universal quantifications may allow us to take benefit of the reductions of MSO to its negative fragment, as provided by the *Safraless* approaches to synthesis [KV05, KPV06, FJR11].

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 for 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.

Acknowledgment

The authors would like to thank to anonymous referees for their thorough readings of previous versions of this paper, which helped a lot in raising its quality.

References

[BJP+12] 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

[BL69] J. R. Büchi and L. H. Landweber. Solving Sequential Conditions by Finite-State Strategies. *Transation of the American Mathematical Society*, 138:367–378, 1969. 2, 6, 7, 21

[Büc62] 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, 5, 6

[Chu57] 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

[FJR11] 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. 2, 30

[Gir72] J.-Y. Girard. *Interprétation Fonctionnelle et Élimination des Coupures de l'Arithmétique d'Ordre Supérieur*. PhD thesis, Université Paris 7, 1972. 10

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989. 10

[GTW02] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002. 2, 6, 17

[Jac01] B. Jacobs. *Categorical Logic and Type Theory*. Studies in logic and the foundations of mathematics. Elsevier, 2001. 2, 25, 26, 27, 28, 29

[Kle52] S.C. Kleene. *Introduction to Metamathematics*. North Holland, 1952. 12

[Koh08] U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Springer Monographs in Mathematics. Springer, 2008. 2, 10, 12, 19, 30

[KPV06] 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. 2, 30

[KV05] O. Kupferman and M. Y. Vardi. Safraless decision procedures. In *Proceedings of FOCS'05*, pages 531–542, Washington, DC, USA, 2005. IEEE Computer Society. 2, 30

[McN66] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521 – 530, 1966. 1, 6

[ML98] S. Mac Lane. *Categories for the Working Mathematician*. Springer, 2nd edition, 1998. 28

[PP04] D. Perrin and J.-É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Pure and Applied Mathematics. Elsevier, 2004. 1, 2, 6, 22

[PR17] 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

[PR18] P. Pradic and C. Riba. LMSO: A Curry-Howard Approach to Church's Synthesis via Linear Logic. In *Proceedings of LICS'18*. ACM, 2018. 21

[Rab72] M. O. Rabin. Automata on infinite objects and Church's Problem. *Amer. Math. Soc.*, 1972. 2, 6

[Rib12] 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, 7, 8, 9, 14, 16, 18, 32, 36

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

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

Strict Linear Order Axioms:

$$\neg (x \dot{<} x) \qquad\quad (x \dot{<} y \dot{<} z \rightarrow x \dot{<} z) \qquad\quad (x \dot{<} y \ \vee \ x \dot{=} y \ \vee \ y \dot{<} x)$$

Predecessor and Unboundedness Axioms:

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

Figure 9: The Arithmetic Axioms of [Rib12].

[Rib16]   C. Riba. Monoidal-Closed Categories of Tree Automata. To appear in *Mathematical Structures in Computer Science*. Available on HAL (hal-01261183), 2016. 20, 21, 30

[Sie70]   D. Siefkes. *Decidable Theories I : Büchi's Monadic Second Order Successor Arithmetic*, volume 120 of *LNM*. Springer, 1970. 2, 7, 8, 14, 16

[SU06]   M. H. Sørensen and P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism*, volume 149 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science Inc., 2006. 2, 19

[Tho97]   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, 6, 14, 17, 21, 22

[Tho08]   W. Thomas. Solution of Church's Problem: A tutorial. *New Perspectives on Games and Interaction*, 5:23, 2008. 1, 3, 4, 6, 15

[Tho09]   W. Thomas. Facets of Synthesis: Revisiting Church's Problem. In L. de Alfaro, editor, *Proceedings of FOSSACS'09*, pages 1–14. Springer, 2009. 1, 2

[VW08]   M. Y. Vardi and T. Wilke. Automata: from logics to algorithms. In *Logic and Automata*, volume 2 of *Texts in Logic and Games*, pages 629–736. Amsterdam University Press, 2008. 1, 2, 6

[Yan08]   Q. Yan. Lower Bounds for Complementation of omega-Automata Via the Full Automata Technique. *Logical Methods in Computer Science*, 4(1), 2008. 6

Appendix A. Completeness of MSO (Thm. 2.16)

In this Appendix, we provide the missing details to deduce the completeness of our axiomatization of MSO (Thm. 2.16) from [Rib12]. The arithmetic axioms of [Rib12] expressed with $\dot{<}$ and S are presented in Fig. 9, where

$$(x \dot{<} y) \quad := \quad \big(x \dot{\leq} y \ \wedge \ \neg (x \dot{=} y) \big)$$

The axioms of Fig. 9 follow from Lem. 2.17 (Fig. 6) that we prove now.

Proof of Lem. 2.17. Let us recall the non-logical rules of MSO (omitting equality):

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

$$\frac{}{\overline{\varphi} \vdash x \dot{\leq} x} \qquad \frac{\overline{\varphi} \vdash x \dot{\leq} y \quad\quad \overline{\varphi} \vdash y \dot{\leq} z}{\overline{\varphi} \vdash x \dot{\leq} z} \qquad \frac{\overline{\varphi} \vdash x \dot{\leq} y \quad\quad \overline{\varphi} \vdash y \dot{\leq} x}{\overline{\varphi} \vdash x \dot{=} y}$$

- Basic Z and S rules (total injective relations):

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

$$\frac{}{\overline{\varphi} \vdash \exists y \, \mathsf{S}(x,y)} \qquad \frac{\overline{\varphi} \vdash \mathsf{S}(y,x) \quad\quad \overline{\varphi} \vdash \mathsf{S}(z,x)}{\overline{\varphi} \vdash y \dot{=} z} \qquad \frac{\overline{\varphi} \vdash \mathsf{S}(x,y) \quad\quad \overline{\varphi} \vdash \mathsf{S}(x,z)}{\overline{\varphi} \vdash y \dot{=} z}$$

- Arithmetic rules:

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

We now proceed to the proof of the properties listed in Fig. 6.

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

Proof. From reflexivity of equality. □

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

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

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

Proof. By induction on y, we show

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

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

$$\phi(y), \mathsf{S}(y,y'), \mathsf{S}(x,y'), x \dot{=} y' \vdash \bot$$

Note that

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

From which follows that

$$\phi(y), \mathsf{S}(y,y'), \mathsf{S}(x,y'), x \dot{=} y' \vdash \bot$$

□

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

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

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

Proof. We have

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

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

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

and we conclude by (3). □

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

Proof. By induction on y. □

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

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

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

Proof. We have

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

Hence by (7) we get

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

and we conclude by the basic rules for Z. □

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

Proof. By induction on y, we show

$$\phi(y) \quad := \quad \forall x, x' \left(x \mathbin{\dot{<}} y \to \mathsf{S}(x, x') \to x' \mathbin{\dot{\le}} y\right)$$

First, the base case $\mathsf{Z}(y) \vdash \phi(y)$ follows from the fact that $\mathsf{Z}(y), x \mathbin{\dot{<}} y \vdash \bot$ by (7). For the induction step, we show

$$\mathsf{S}(y, y'), \phi(y), x \mathbin{\dot{<}} y', \mathsf{S}(x, x') \vdash x' \mathbin{\dot{\le}} y'$$

We use the excluded middle on $x \mathbin{\dot{=}} y$, and we are left with showing

$$\mathsf{S}(y, y'), \phi(y), x \mathbin{\dot{<}} y', \neg(x \mathbin{\dot{=}} y), \mathsf{S}(x, x') \vdash x' \mathbin{\dot{\le}} y'$$

But by the arithmetic rules, $\mathsf{S}(y, y'), x \mathbin{\dot{<}} y' \vdash x \mathbin{\dot{\le}} y$, so that

$$\mathsf{S}(y, y'), \phi(y), x \mathbin{\dot{<}} y', \neg(x \mathbin{\dot{=}} y), \mathsf{S}(x, x') \vdash x \mathbin{\dot{<}} y$$

But

$$\phi(y), x \mathbin{\dot{<}} y, \mathsf{S}(x, x') \vdash x' \mathbin{\dot{\le}} y$$

and we are done.   □

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

Proof. By induction on z we show

$$\phi(z) \quad := \quad \forall x, x', y \left(x \mathbin{\dot{\le}} y \to \mathsf{S}(x, x') \to \mathsf{S}(y, z) \to x' \mathbin{\dot{\le}} z\right)$$

We trivially have $\mathsf{Z}(z) \vdash \phi(z)$. We now show $\mathsf{S}(z, z'), \phi(z) \vdash \phi(z')$, that is

$$\mathsf{S}(z, z'), \phi(z), x \mathbin{\dot{\le}} y, \mathsf{S}(x, x'), \mathsf{S}(y, z') \vdash x' \mathbin{\dot{\le}} z'$$

By the basic Z and S rules, this amounts to show

$$\phi(z), x \mathbin{\dot{\le}} z, \mathsf{S}(x, x'), \mathsf{S}(z, z') \vdash x' \mathbin{\dot{\le}} z'$$

Now, using the excluded middle on $x \mathbin{\dot{=}} z$, we are left with showing

$$\phi(z), \mathsf{S}(x, x'), \mathsf{S}(z, z'), x \mathbin{\dot{<}} z \vdash x' \mathbin{\dot{\le}} z'$$

But by (9) we have

$$x \mathbin{\dot{<}} z, \mathsf{S}(x, x') \vdash x' \mathbin{\dot{\le}} z$$

and we are done.   □

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

Proof. The right-to-left direction follows from (3). For the left-to-right direction, by induction on x, we show

$$\phi(x) \quad := \quad \forall y \left(y \mathbin{\dot{<}} x \ \longrightarrow \ \exists z (y \mathbin{\dot{\le}} z \ \wedge \ \mathsf{S}(z, x))\right)$$

For the base case $\mathsf{Z}(x) \vdash \phi(x)$, by (7) we have $\mathsf{Z}(x), y \mathbin{\dot{\le}} x \vdash \mathsf{Z}(y)$ and we conclude by the basic rules for Z. For the induction step, we have to show

$$\mathsf{S}(x, x'), \phi(x), y \mathbin{\dot{<}} x' \vdash \exists z (y \mathbin{\dot{\le}} z \ \wedge \ \mathsf{S}(z, x'))$$

By the last arithmetic rule,

$$\mathsf{S}(x, x'), y \mathbin{\dot{<}} x' \vdash y \mathbin{\dot{\le}} x$$

and we are done.   □

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

   Proof. By induction on x, we show

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

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

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

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

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

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

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

   and we are done since (10) gives

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

   $\square$

(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 with showing

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

   But the last arithmetic rule gives $z \mathrel{\dot{\leq}} x$ from $\mathsf{S}(x, y)$ and $z \mathrel{\dot{<}} y$, which together with $x \mathrel{\dot{<}} z$ gives $z \mathrel{\dot{=}} x$ by antisymmetry of $\mathrel{\dot{\leq}}$, contradicting $x \mathrel{\dot{<}} z$.

   Conversely, assume that $x \mathrel{\dot{<}} y$ without any intermediate z. By the basic rules for S, we have $\mathsf{S}(x, z)$ for some z. Note that $x \mathrel{\dot{<}} z$ by (3). Since $x \mathrel{\dot{<}} y$ it follows from (9) that $z \mathrel{\dot{\leq}} y$. But this implies $z \mathrel{\dot{=}} y$ as $\neg(z \mathrel{\dot{<}} y)$. $\square$

This concludes the proof of Lem. 2.17. $\square$

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 Lem. 2.18, namely that strong induction is derivable in MSO. The proof holds no surprise.

Proof of Lem. 2.18. We have to show

$$\forall x \big(\forall y (y \mathrel{\dot{<}} x \;\to\; Xy) \;\longrightarrow\; Xx \big) \;\vdash\; \forall x \, Xx$$

By induction on x we show

$$\forall x \big(\forall y (y \mathrel{\dot{<}} x \;\to\; Xy) \;\longrightarrow\; Xx \big) \;\vdash\; \phi(x)$$

where

$$\phi(x) \quad := \quad \forall y (y \mathrel{\dot{\leq}} x \;\to\; Xy)$$

The base case

$$\forall x \big(\forall y (y \mathrel{\dot{<}} x \;\to\; Xy) \;\longrightarrow\; X(x) \big), \mathsf{Z}(x) \;\vdash\; \phi(x)$$

is trivial since by (6),

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

For the induction step, we have to show

$$\forall x \big(\forall y (y \mathrel{\dot{<}} x \;\to\; Xy) \;\longrightarrow\; Xx \big), \mathsf{S}(x, x'), \phi(x), z \mathrel{\dot{\leq}} x' \;\vdash\; Xz$$

Notice that $\phi(x)$ is equivalent to $\phi'(x') := \forall y(y \mathrel{\dot{<}} x' \to Xy)$ thanks to (11). By (12), we have three subcase according to:

$$z \mathrel{\dot{<}} x' \;\vee\; z \mathrel{\dot{=}} x' \;\vee\; x' \mathrel{\dot{<}} z$$

The first case enables us to use $\phi'(x')$ directly, and the second one follows from the assumption $\forall x(\forall y(y \mathrel{\dot{<}} x \to Xy) \to Xx)$ together with $\phi'(x')$. The last one leads to a contradiction using the antisymmetry of $\mathrel{\dot{\le}}$. $\qquad\square$

APPENDIX B. INTERNALLY BOUNDED FORMULAE (§4.4)

We prove here the Splitting Lemma 4.13, used in the proof of Thm. 4.11. We consider formulae over the vocabulary of [Rib12], that is formulae given by the grammar

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

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

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

we obtain for each formula in the sense of Fig. 2 an equivalent formula in Λ (w.r.t. the standard model $\mathfrak{N}$). Note that the Transfer Lemma 4.12 gives in particular that if $B_0, B_1 \in \mathbf{2}^\omega \simeq \mathcal{P}(\mathbb{N})$ are disjoint, then

$$\mathfrak{N} \models \;\; \exists X(\varphi_0{\upharpoonright}B_0 \;\wedge\; \varphi_1{\upharpoonright}B_1) \;\;\longleftrightarrow\;\; \big(\exists X(\varphi_0{\upharpoonright}B_0) \;\wedge\; \exists X(\varphi_1{\upharpoonright}B_1)\big) \qquad (\text{B.1})$$

Lemma B.1 (Splitting (Lem. 4.13))**.** *Let ψ be a formula in Λ and let z be an individual variable. For every set of individual variables V with $z \in V$, one can produce a natural number N and two matching sequences of length N of left formulae $(L_j)_{j<N}$ and right formulae $(R_j)_{j<N}$ such that the following holds:*

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

$$\mathfrak{N} \models \;\; \psi[\overline{a}/\overline{x}, n/z, \overline{b}/\overline{y}] \;\;\longleftrightarrow\;\; \bigvee_{j<N} L_j[\overline{a}/\overline{x}, n/z]{\upharpoonright}[- \mathrel{\dot{\le}} n] \;\wedge\; R_j[\overline{b}/\overline{y}]{\upharpoonright}[- \mathrel{\dot{>}} n]$$

Proof. The proof proceeds by induction on ψ. The cases of $\top$ and $\bot$ are trivial and omitted.

Case of $(x \mathrel{\dot{<}} y)$**:** We take $N := 1$ and we define suitable left and right formulae according to V. In each case the choice of z is irrelevant.

- If $x, y \in V$, then $L_0 := \psi$ and $R_0 := \top$.
- If $x, y \notin V$, then $L_0 := \top$ and $R_0 := \psi$.
- If $x \in V$ and $y \notin V$, then $L_0 := R_0 := \top$.
- If $y \in V$ and $x \notin V$, then $L_0 := R_0 := \bot$.

Case of $(x \mathrel{\dot{\in}} X)$**:** We take $N := 1$. Then one of the produced formula is ψ and the other one is $\top$ according to whether $x \in V$ or not.

Case of $(\psi \vee \psi')$**:** Let $(L_j, R_j)_{j<N}$ and $(L'_k, R'_k)_{k<N'}$ be obtained by applying the induction hypothesis on ψ and ψ' respectivelly. Then for $\psi \vee \psi'$ we take $N'' := N + N'$ and the sequence $(L''_i, R''_i)_{i<N''}$ given by

$$
\begin{array}{llllll}
L''_j & := & L_j & R''_j & := & R_j & \text{(for } j < N) \\
L''_{N+k} & := & L'_k & R''_{N+k} & := & R'_k & \text{(for } k < N')
\end{array}
$$

Case of $(\exists x\,\psi)$**:** Note that we can assume $x \notin V$. We apply the induction hypothesis on ψ twice: once with $V \cup \{x\}$ and once with V. This gives sequences resp. $(L_j, R_j)_{j<N}$ and $(L'_k, R'_k)_{k<N'}$. For $\exists x\,\psi$ we take $N'' := N + N'$ and the sequence $(L''_i, R''_i)_{i<N''}$ given by

$$
\begin{array}{llllll}
L''_j & := & \exists x\, L_j & R''_j & := & R_j & \text{(for } j < N) \\
L''_{N+k} & := & L'_k & R''_{N+k} & := & \exists x\, R'_k & \text{(for } k < N')
\end{array}
$$

The disjunction is then seen to be equivalent to $\exists x\,\psi$ by making a case analysis over whether $x \overset{\cdot}{\le} n$ holds.

Case of $(\exists X\,\psi)$**:** Let $(L_j, R_j)_{j<N}$ be obtained by induction hypothesis on ψ. Then for $\exists X\,\psi$ we keep the same N and it directly follows from (B.1) that we can take the sequence $(L'_j, R'_j)_{j<N}$ given by $L'_j := \exists X L_j$ and $R'_j := \exists X R_j$.

Case of $(\neg\psi)$**:** By induction hypothesis, we have a natural number N and two sequences of formulae $(L_j, R_j)_{j<N}$ such that $\psi \longleftrightarrow \bigvee_{j<N} L_j\!\restriction\![- \overset{\cdot}{\le} n] \wedge R_j\!\restriction\![- \overset{\cdot}{>} n]$. Hence all we need to do is to add the negation, push it through the disjunction and conjunctions using De Morgan laws and make the disjuncts commute over the conjunction in the obtained formula. More explicitly (leaving the parameters implicit), we have:

$$
\begin{aligned}
\neg\psi \quad &\longleftrightarrow \quad \bigwedge_{j<N} \neg L_j\!\restriction\![- \overset{\cdot}{\le} n] \vee \neg R_j\!\restriction\![- \overset{\cdot}{>} n] \\
&\longleftrightarrow \quad \bigvee_{f\in\mathbf{2}^{\{0,\dots,N-1\}}} \bigwedge_{j\in f^{-1}(0)} \neg L_j\!\restriction\![- \overset{\cdot}{\le} n] \wedge \bigwedge_{j\in f^{-1}(1)} \neg R_j\!\restriction\![- \overset{\cdot}{>} n]
\end{aligned}
$$

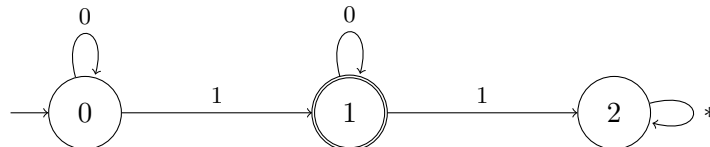$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Remark B.2. Note that there is a combinatorial explosion in the case of $\neg\psi$ in Lem. B.1 since $N_{\neg\psi} = 2^{N_\psi}$. It follows that the sizes of the formulae computed in Lem. B.1 and the subsequent Thm. 4.11 are non-elementary in the size of ψ.
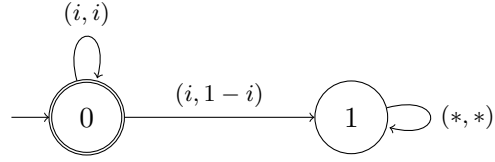
Appendix C. Automata for Atomic Formulae (§5.3)

We give below the automaton Sing (of §5.3) and automata for the atomic formulae of Fig. 2. 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, quantifications over individuals are relativized to Sing.
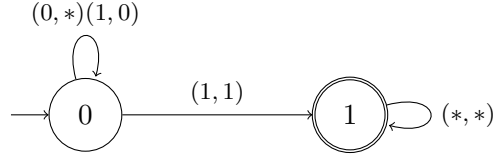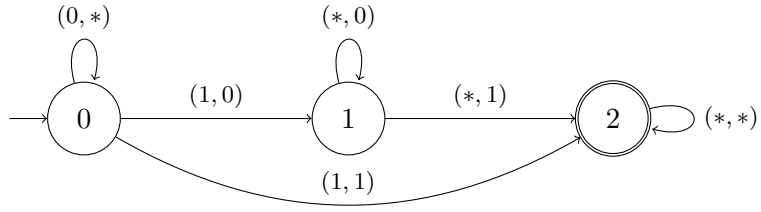
- **Sing :**

- $(x_1 \overset{.}{=} x_2)$:
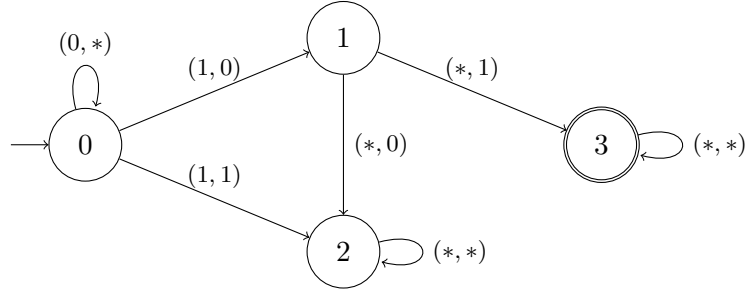
(i,i)

$$\rightarrow \boxed{0} \xrightarrow{(i,1-i)} \boxed{1} \quad (*,*)$$

- $(x_1 \overset{.}{\in} X_1)$:

$(0,*)(1,0)$

$$\rightarrow \boxed{0} \xrightarrow{(1,1)} \boxed{1} \quad (*,*)$$

- $(x_1 \overset{.}{\leq} x_2)$:

$(0,*)$ $(*,0)$

$$\rightarrow \boxed{0} \xrightarrow{(1,0)} \boxed{1} \xrightarrow{(*,1)} \boxed{2} \quad (*,*)$$

$(1,1)$

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

$(0,*)$

$$\rightarrow \boxed{0} \xrightarrow{(1,0)} \boxed{1} \xrightarrow{(*,1)} \boxed{3} \quad (*,*)$$

$(1,1)$ $(*,0)$

$$\boxed{2} \quad (*,*)$$

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

$$\boxed{1} \quad *$$

$$\rightarrow \boxed{0} \quad 1$$

$$0 \quad \boxed{2} \quad *$$

- $\top$ and $\bot$:

$$\rightarrow \boxed{0} \quad * \qquad \text{and} \qquad \rightarrow \boxed{0} \quad *$$