École Normale Supérieure de Lyon
Laboratoire de l'Informatique du Parallélisme - UMR 5668 - LIP

Habilitation à diriger des recherches

Informatique

Présentée et soutenue publiquement le 2 décembre 2019 par
Colin Riba

Toward Curry-Howard Approaches to MSO and Automata on Infinite Words and Trees

devant le jury composé de

M.	Olivier	Laurent	Examinateur
M.	Paul-André	Melliès	Examinateur
M.	Sylvain	Salvati	Rapporteur
M.	Luigi	Santocanale	Rapporteur
Mme.	Alexandra	Silva	Rapportrice
M.	Alex	Simpson	Examinateur

Résumé

Nous présentons des travaux ayant pour objectif de proposer une approche Curry-Howard à la Logique Monadique du Second-Ordre (MSO) sur les arbres infinis et les ω-mots.

Le *Théorème de l'Arbre* de Rabin, à savoir la décidabilité de MSO sur les arbres infinis, est un outil puissant, qui a fourni des arguments de décidabilité pour de nombreuses logiques et théories mathématiques. Alors que ce résultat date de la fin des années 60, il y a eu depuis un travail considérable sur sa preuve, aboutissant à des arguments basés sur une correspondance triangulaire entre logiques, automates et jeux infinis.

L'objectif des travaux présentés ici est de revisiter cette correspondance selon la perspective de la correspondance *peuves-programmes* de Curry-Howard. Nous proposons un modèle de réalisabilité pour des automates d'arbres alternants, dans lequel les automates sont assimilés à des types et les stratégies d'acceptation sont vues comme des programmes. Nous observons, à travers ce modèle, que les opérations naturelles sur les automates utilisées dans les traductions de formules MSO en automates sous-jacentes au théorème de Rabin correspondent à des connecteurs de la logique (des prédicats) *linéaire intuitionniste* (ILL). En d'autres termes, le langage de ILL reflète des opérations sur les automates alternants ayant un grain plus fin que les connecteurs de MSO. Ainsi, ILL peut-être utilisé comme un langage intermédiaire entre MSO et les automates d'arbres.

Lorsque l'on restreint ce modèle au cas des ω-mots, on retrouve les équivalences usuelles entre automates déterministes, non-déterministes, universels et alternants. En s'appuyant sur l'axiomatisation complète par Siefkes de MSO sur les ω-mots en un sous-système de l'arithmétique de Peano du second-ordre (PA2), nous obtenons, grâce à une variante de l'interprétation fonctionnelle « Dialectica » de Gödel, une logique linéaire $\mathsf{LMSO}(\mathfrak{C})$ complète et *non-standard*, qui, *via* un système de polarités, est correcte et complète pour la synthèse de Church : il existe une classe syntaxique d'énoncés $\forall\exists$, extractibles et dont la prouvabilité correspond exactement aux instances résolubles de la synthèse de Church. Nous discutons aussi brièvement des questions reliées à l'axiomatisation de MSO sur les arbres infinis. Nos résultats dans cette directions nous semblent en revanche plus préliminaires.

Abstract

We present works aiming at proposing a Curry-Howard approach to Monadic Second-Order Logic (MSO) on infinite trees and ω-words.

Rabin's *Tree Theorem*, the decidability of MSO over infinite trees, is a powerfull tool, which provided decidability proofs for many logics and mathematical theories. While the result dates back to the late 60's, there have been since then considerable work on its proof, culminating in streamlined arguments based on a triangular correspondence between logics, automata, and infinite games.

The goal of the works presented here is to revisit this correspondence from the perspective of the Curry-Howard *proofs-as-programs* correspondence. We propose a realizability model for (alternating) tree automata, based on usual categories of *simple games*, and following the slogan "automata as types, strategies as programs". Within this model, we observe that natural operations on automata used in the translations of MSO-formulae to automata underlying Rabin's Tree Theorem correspond to connectives of *intuitionistic* (predicate) *linear logic* (ILL). In other words, the language of ILL reflects operations on alternating automata which are finer grained than the connectives of MSO. As a consequence, ILL can be used as an intermediate language between MSO and tree automata.

When we restrict this model to the case of MSO over ω-words, one recovers the usual equivalence between determinisitc, non-deterministic, universal and alternating automata. Building on Siefkes's complete axiomatization of MSO over ω-words as a subsystem of Second-Order Peano Arithmetic (PA2), we obtain, thanks to a variant of Gödel's *functional "Dialectica" interpretation*, a complete, *non-standard*, linear logic $\mathsf{LMSO}(\mathfrak{C})$, which, via a polarization policy, is sound and complete w.r.t. Church's synthesis: there is a class of extractible $\forall\exists$-statements whose provability exactly corresponds to the solvable instances of Church's synthesis. We also briefely discuss questions related to the axiomatization of MSO on infinite trees, seen as a subsystem of PA2. By contrast, we see our results in this direction as beeing more preliminary.

Remerciements

Je tiens tout d'abord à remercier chaleureusement les membres de mon jury, et en particulier Alexandra Silva, Luigi Santocanale et Sylvain Salvati, pour avoir accepté la tâche ingrate de rapporter ce manuscrit sans savoir au préalable qu'il allait contenir des détails techniques probablement peu habituels pour un mémoire d'habilitation.

Il m'est impossible de remercier nommément tout(e)s les collègues sans lesquel(le)s ce mémoire n'aurait pas existé. Ne pouvant cependant m'y résigner, je me dois de dire que les travaux présentés ici, bien que n'ayant pas été initiés en collaboration, ont bénéficié lors de leur genèse d'interactions avec différents spécialistes des domaines en jeu, en particulier Thomas Colcombet et Arnaud Carayol côté automate, ainsi qu'Alexandre Miquel, Paul-André Melliès et Thomas Streicher côté sémantique et réalisabilité. Bien entendu, j'ai été ravi de pouvoir attirer des chercheurs plus jeunes dans cette aventure, notamment Anupam Das et Pierre Pradic. Je tiens à les remercier ici pour la richesse des contacts que j'ai pu avoir avec eux, tant au niveau scientifique que personnel. Les membres de l'équipe Plume du LIP, et en particulier Pierre Clairambault, contribuent jour après jour à en faire un lieu où le genre de travail présenté ici est possible. Je tiens plus généralement à remercier l'ensemble des membres de la communauté CHoCoLa, qui la font vivre telle qu'elle est.

Pour des raisons d'unité thématique, ce mémoire ne fait pas référence aux travaux menés en collaboration avec Valentin Blot lors de sa thèse. Je tiens cependant à le remercier chaleureusement pour m'avoir offert cette première expérience d'encadrement.

Last but not least, les travaux présentés ici doivent aussi énormément à toutes celles et ceux qui, dans et hors du monde académique, comptent ou ont compté mais ne liront probablement jamais ces lignes...

Contents

1. Introduction

This document presents works aiming at proposing a Curry-Howard approach to Monadic Second-Order Logic (MSO) on infinite trees and ω-words. It consists in two parts (I and II), preceded by a preliminary Chapter 2. Part II, which develops the Curry-Howard approach in itself is by far the largest, and has a specific technical introduction in Chap. 4. Part I, which presents material related to axiomatizations of MSO, merely consists of Chap. 3.

The works covered by this document are [Rib12, Rib15, Rib18], together with [DR19] (in collaboration with Anupam Das) and [PR17, PR18b, PR19, PR18a] (in collaboration with Pierre Pradic). We wished not to formally include [DR15, Rib13] (which arguably could have been part of the above list) for (distinct) reasons that we give in the introduction to Chap. 3.

The remaining of this Chapter outlines the components of *Rabin's Tree Theorem* [Rab69] (the decidability of MSO on infinite trees) which underlie the Curry-Howard approach developed in Part II. While some of these components are also important for Part I, we do not really cover the latter here because, even if it is instrumental to the Curry-Howard approach outlined here, Part I is (by now) *only* instrumental, in that the two subjects are (yet) mostly technically disconnected. We refer to §8.3 and §9.1 for directions of future work possibly involving more interactions between these two aspects.

Chapter 2 contains some further preliminaries, which are relevant to both Part I and Part II while being too technical to be discussed here. We also present in App. A a setting of simple games on which most of Part II is based.

1.1. Basic Notations. We fix a finite non-empty set $\mathfrak{D}$ of *tree directions*. We are interested in labelings of the full $\mathfrak{D}$-ary tree $\mathfrak{D}^*$ over different *alphabets*. Alphabets (denoted Σ, Γ, etc) are finite non-empty sets, and Σ-labeled $\mathfrak{D}$-ary trees are functions $T : \mathfrak{D}^* \to \Sigma$. Concatenation of sequences s, t is denoted either $s.t$ or $s \cdot t$, and ε is the empty sequence. We denote with overlines both vectors and finite words, so that e.g. $\overline{T}$ denotes a sequence $\overline{T} = T_1, \dots, T_n$, while $\overline{\mathtt{a}} \in \Sigma^*$ denotes a word $\overline{\mathtt{a}} = \mathtt{a}_1. \cdots .\mathtt{a}_n$ where each $\mathtt{a}_i$ is a letter of Σ. Given an ω-word (or stream) $B \in \Sigma^\omega$ and $n \in \mathbb{N}$, we write $B\restriction n$ for the finite word $B(0). \cdots .B(n-1) \in \Sigma^*$.

1.2. Monadic Second-Order Logic (MSO). There are different expressively equivalent ways of formulating MSO over infinite trees, each of them best suited for a specific situation. As MSO is looked at from different points of view in this document, we shall unfortunately consider different languages for MSO.

We believe that the simplest view is to see MSO as a two-sorted logic, with a sort of *individuals* ranging over the positions of the full $\mathfrak{D}$-ary tree $\mathfrak{D}^*$ (that is over $\mathfrak{D}^*$ itself) and a sort of *monadic second-order* variables ranging over sets of positions (that is over $\mathcal{P}(\mathfrak{D}^*)$). Note that one can identify sets of positions with labeled trees $\mathfrak{D}^* \to \mathbf{2}$. Conversely, a labeled tree $T : \mathfrak{D}^* \to \Sigma$ can be represented as a k-tuple

$$\langle T_1, \dots, T_k \rangle \quad : \quad \mathfrak{D}^* \quad \longrightarrow \quad \mathbf{2}^k$$

for some suitable k depending on Σ. There are also different possible choices for this, and the best option again depends on the context.

On the other hand, when discussing translations of formulae to automata, it is actually customary and convenient to only allow monadic variables, and to simulate quantifications over individuals via a (definable) singleton predicate. This is the setting we shall adopt for now. Assuming given a set At of atomic relations, we consider MSO formulae given by

$$\varphi, \psi \quad ::= \quad \alpha \quad | \quad \neg\varphi \quad | \quad \varphi \wedge \psi \quad | \quad (\exists X)\varphi \qquad \text{(where } \alpha \in \mathrm{At})$$

These formulae are interpreted in the full $\mathfrak{D}$-ary tree $\mathfrak{D}^*$ as expected, assuming an interpretation of the atomic predicates.

There are again different expressively equivalent choices for At. Maybe the simplest possibility is to let At consist of the binary relation symbol $(-) \dot{\subseteq} (-)$ together with one binary relation symbol $\mathsf{S}_d(-,-)$ for each $d \in \mathfrak{D}$. The symbol $(-) \dot{\subseteq} (-)$, interpreted as set inclusion, allows for the definition of set equality, but also e.g. of “being the empty set” and “being a singleton set”. The symbol S_d stands for the d-successor relation $\{(\{p\}, \{p.d\}) \mid p \in \mathfrak{D}^*\}$. Using quantification, one can define e.g. the prefix order (for singleton sets).

MSO on infinite trees is a rich system, which contains non trivial mathematical theories (see e.g. [Rab69, BGG97]), and which subsumes many logics, in particular modal logics (see e.g. [BdRV02]) and most logics used in verification (see e.g. [VW08]).

One of the central result around which this work is built is *Rabin's Tree Theorem*.

Theorem 1.2.1 (Rabin [Rab69])**.** MSO *over infinite trees is decidable.*

The original proof of [Rab69] relied on an effective translation of formulae to finite state automata running on infinite trees. Since then, there have been considerable work on Rabin's Tree Theorem, culminating in streamlined decidability proofs, as presented e.g. in [Tho97, GTW02, PP04]. Most current approaches to MSO on infinite trees (but with the notable exception of [Blu13]) are based on translations of MSO-formulae to automata.

1.3. (Non-Deterministic) Tree Automata. There are two families of tree automata involved in the interpretation of MSO formulae: *non-deterministic* tree automata and *alternating* tree automata[1]. The simplest notion is that of non-deterministic automaton, and it is sufficient for introducing the basic motivations and methodology of this work.

A tree automaton $\mathcal{A}$ consists of a finite set Q of states, with a distinguished[2] initial state $q^\imath \in Q$, an acceptance condition given by an ω-regular[3] set $\Omega \subseteq Q^\omega$, and a transition function ∂. A *non-deterministic* tree automaton $\mathcal{A}$ over Σ has a transition function of the form

$$\partial \quad : \quad Q \times \Sigma \quad \longrightarrow \quad \mathcal{P}(\mathfrak{D} \longrightarrow Q)$$

Acceptance for tree automata can equivalently be described by *games* or *run trees*. The notion of run tree is simpler and sufficient at various places of this Chapter. A *run tree* of $\mathcal{A}$ on $T : \mathfrak{D}^* \to \Sigma$ is a tree $\rho : \mathfrak{D}^* \to Q$ such that $\rho(\varepsilon) = q^\imath$, and which respects the transitions of $\mathcal{A}$, in the sense that for each tree position $p \in \mathfrak{D}^*$, there exists a $\mathfrak{D}$-tuple $(q_d)_{d \in \mathfrak{D}} \in \partial(\rho(p), T(p))$ such that $\rho(p.d) = q_d$ for all $d \in \mathfrak{D}$. The run ρ is *accepting* if all its infinite paths belong to Ω. We say that T is accepted by $\mathcal{A}$ if there exists an accepting run of $\mathcal{A}$ on T, and let $\mathcal{L}(\mathcal{A})$ be the set of trees accepted by $\mathcal{A}$. We moreover write $\mathcal{A}(T)$ for the set of accepting runs of $\mathcal{A}$ on T.

[1] Alternating automata are not always made explicit (see e.g. [Tho97]).

[2] It is also customary (and equivalent in terms of expressiveness) to allow several initial states.

[3] See §2.1 if a definition is needed.

1.4. Games and Alternating Automata. The main difficulty when translating MSO formulae to tree automata is the interplay between negation and (existential) quantification. Historically, Rabin [Rab69] translated MSO formulae to non-deterministic tree automata.

While non-deterministic automata are easily (and linearly in the number of states) closed under projections $\widetilde{\exists}_{\Sigma}(-)$, which implement the existential quantifications of MSO (see §1.6), the major achievement of Rabin [Rab69] was to show that non-deterministic automata on infinite trees are closed under complement.

Rabin's original construction [Rab69] of a complement $\sim\mathcal{A}$ of $\mathcal{A}$ has been considerably simplified by Gurevich and Harrington [GH82] thanks to the notion of *acceptance game*. The idea is to model the evaluation of an automaton $\mathcal{A}$ on an input tree T as an infinite two-players game $\mathcal{G}(\mathcal{A}, T)$. In this game, the *Proponent* P (also called ∃loïse or Automaton) plays for acceptance while its *Opponent* O (also called ∀bélard or Pathfinder) plays for rejection, and $\mathcal{A}$ accepts T when P has a winning strategy.[4] A typical (infinite) play χ in $\mathcal{G}(\mathcal{A}, T)$ has the form:

$$\begin{array}{ccccccccccccccc}
\mathsf{P} & & \mathsf{O} & & \mathsf{P} & & \mathsf{O} & & & & \mathsf{P} & & \mathsf{O} & & \\
(q_{0,d})_{d\in\mathfrak{D}} & \cdot & d_0 & \cdot & (q_{1,d})_{d\in\mathfrak{D}} & \cdot & d_1 & \cdot & \cdots & \cdot & (q_{n+1,d})_{d\in\mathfrak{D}} & \cdot & d_{n+1} & \cdot & \cdots \\
\Cap & & \Cap & & \Cap & & \Cap & & & & \Cap & & \Cap & & \\
\partial(q^{\imath}, T(\varepsilon)) & & \mathfrak{D} & & \partial(q_{0,d_0}, T(d_0)) & & \mathfrak{D} & & & & \partial(q_{n,d_n}, T(p)) & & \mathfrak{D} & &
\end{array}$$

where $p = d_0 \cdot \ldots \cdot d_n$. Then χ is winning for P if the sequence of states $q^{\imath}, q_{0,d_0}, q_{1,d_1}, \ldots$ belongs to Ω; otherwise it is winning for O. Note that P chooses transitions $(q_d)_{d\in\mathfrak{D}}$ while O chooses tree directions $d \in \mathfrak{D}$. Hence, there is a bijection between accepting runs $\rho \in \mathcal{A}(T)$ and winning P-strategies in $\mathcal{G}(\mathcal{A}, T)$. Since acceptance games are determined,[5] $\mathcal{A}$ does not accept T precisely when O has a winning strategy in $\mathcal{G}(\mathcal{A}, T)$. Gurevich and Harrington [GH82] show that in acceptance games, winning strategies can always be assumed to be finite state w.r.t. game positions of the form $(p, q) \in \mathfrak{D}^* \times Q$, that is to only depend on a finite memory in addition to the game positions in $\mathfrak{D}^* \times Q$.[6] This makes it possible to devise an automaton $\sim\mathcal{A}$ which, using a usual projection operation, non-deterministically checks the existence of winning O-strategies.

However, the construction of $\sim\mathcal{A}$ is still not trivial because the roles of P and O in acceptance games are not symmetric, so that dualizing the acceptance game of a non-deterministic automaton $\mathcal{A}$ does not directly give a *non-deterministic* automaton $\sim\mathcal{A}$. Since [MS87, EJ91, MS95] it is known that the construction of $\sim\mathcal{A}$ can be neatly decomposed using *alternating* automata. The original idea, as stated in e.g. [MS87, MS95], is for an alternating automaton $\mathcal{A}$ with state set Q to have transitions with values in the free distributive lattice over $Q \times \mathfrak{D}$.[7] Following [Wal02] we simply assume that transitions are of the form:

$$\partial \quad : \quad Q \times \Sigma \quad \longrightarrow \quad \mathcal{P}(\mathcal{P}(Q \times \mathfrak{D})) \tag{1.1}$$

and we read $\partial(q, \mathtt{a})$ as the disjunctive normal form

$$\bigvee_{\gamma\in\partial(q,\mathtt{a})} \ \bigwedge_{(q',d)\in\gamma} (q', d)$$

This results in acceptance games where intuitively P plays from disjunctions while O plays from

[4]We refer to App. A for a formal presentation of the game setting we are using in this document.
[5]See §3.4 for some comments on game determinacy from an axiomatic perspective.
[6]This is trivial for P-strategies but not for O-strategies.
[7]Many authors speak of "*positive Boolean formulae*" over $Q \times \mathfrak{D}$.

conjunctions. A typical play in the acceptance game $\mathcal{G}(\mathcal{A}, T)$ with $\mathcal{A}$ alternating has the form

$$\begin{array}{ccccccccccc} \mathsf{P} & & \mathsf{O} & & \mathsf{P} & & \mathsf{O} & & & \mathsf{P} & \\ \gamma_0 & \cdot & (q_0, d_0) & \cdot & \gamma_1 & \cdot & (q_1, d_1) & \cdot & \cdots & \cdot \quad \gamma_{n+1} & \cdot \quad \cdots \\ \Cap & & \Cap & & \Cap & & \Cap & & & \Cap & \\ \partial(q^\imath, T(\varepsilon)) & & \gamma_0 & & \partial(q_0, T(d_0)) & & \gamma_1 & & & \partial(q_n, T(p)) & \end{array}$$

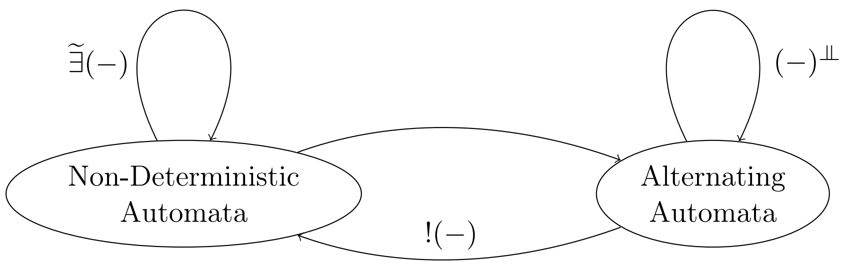
Hence, P chooses relations $\gamma_k \in \mathcal{P}(Q \times \mathfrak{D})$ instead of tuples $(q_{k,d})_{d \in \mathfrak{D}}$ while O chooses pairs $(q_k, d_k) \in \gamma_k$ instead of just tree directions $d_k \in \mathfrak{D}$. The main consequence is that O may now be allowed to choose between pairs $(q'_k, d_k), (q''_k, d_k) \in \gamma_k$ with *different states* q'_k, q''_k for the same tree direction $d_k \in \mathfrak{D}$.

The extra possibility for O to choose states in addition to tree directions allows us to define a complement of $\mathcal{A}$ which essentially simulates $\mathcal{A}$ while reversing the roles of P and O. This can be implemented with an alternating automaton $\mathcal{A}^{\perp\!\!\perp}$ having the same states as $\mathcal{A}$. The idea is that since the double powerset $\mathcal{P}(\mathcal{P}(Q \times \mathfrak{D}))$ in (1.1) represents disjunctive normal forms over $Q \times \mathfrak{D}$, the transition function of $\mathcal{A}^{\perp\!\!\perp}$ can just take $(q, \mathtt{a}) \in Q \times \Sigma$ to a disjunctive normal form representing the dual of $\partial(q, \mathtt{a})$. Then, if the acceptance condition of $\mathcal{A}^{\perp\!\!\perp}$ is the complement of Ω, it follows from game determinacy that $\mathcal{L}(\mathcal{A}^{\perp\!\!\perp})$ is the complement of $\mathcal{L}(\mathcal{A})$.

Every alternating automaton $\mathcal{A}$ can be simulated by a non-deterministic automaton $!\mathcal{A}$ of exponential size (this is the *Simulation Theorem* [MS87, EJ91, MS95], see also §2.2), while non-deterministic automata are linearly embedded into alternating automata via the obvious mapping

$$(q_d)_{d \in \mathfrak{D}} \in Q^{\mathfrak{D}} \quad \longmapsto \quad \{(q_d, d) \mid d \in \mathfrak{D}\} \in \mathcal{P}(Q \times \mathfrak{D}) \tag{1.2}$$

The situation can be pictured as follows:



(1.3)

Accordingly, in most modern approaches to MSO on infinite trees, the complementation of non-deterministic tree automata can be decomposed as

$$\sim\!\mathcal{A} \quad = \quad !(\mathcal{A}^{\perp\!\!\perp}) \tag{1.4}$$

It could be noted now that the actual logical connectives we took for MSO in §1.2 is in part motivated by the indication from (1.3) that natural operations on automata are not too far from (first-order) *tensorial logic* (see e.g. [Mel13] and [Mel17b, §5.6, p. 137]).

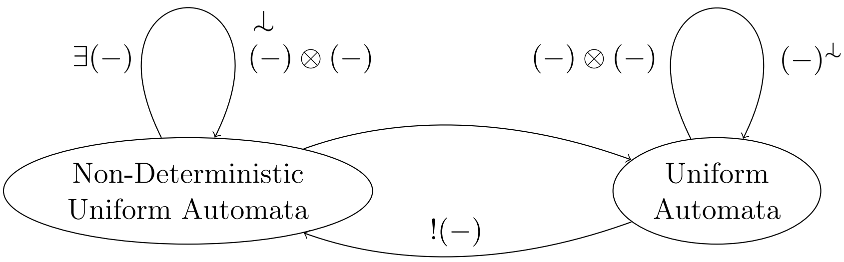
1.5. Toward Linear Logic. We believe that our main contribution is the observation that the operations on tree automata used in the translations of MSO formulae to automata correspond to the connectives of *Intuitionistic Multiplicative Exponential Linear Logic* (IMELL) [Gir87], to the effect that the language of IMELL makes it possible to reflect, at the logical level, the decomposition depicted in (1.3) and (1.4) of the translation of MSO formulae to non-deterministic tree automata via alternating automata.

Along the lines of the Curry-Howard "proofs-as-programs" correspondence (see e.g. [GLT89, SU06]), we devise a realizability semantics on top of tree automata. It consists in categories of games, based on usual categories of two-player sequential games called *simple games* (see e.g. [Abr97, Hyl97]), and which generalize usual acceptance games of tree automata.[8] Thanks in particular to the notion of *uniform automata* (to be introduced in Chap. 7), this allows us to give the following interpretation of (1.3) and (1.4):

- First, the usual direct synchronous product of alternating automata (which we denote $(-) \otimes (-)$) has a symmetric monoidal structure (with unit denoted $\mathbf{I}$). Moreover, uniform automata have a monoidal-*closed* structure w.r.t. $(-) \otimes (-)$. In particular, the set of morphisms from $\mathcal{G}(\mathcal{A}, T)$ to $\mathcal{G}(\mathcal{B}, T)$ is in bijection with the set of winning P-strategies in the acceptance game of an automaton $(\mathcal{A} \multimap \mathcal{B})$ over T. In particular, linear complements are obtained with
$$\mathcal{A}^{\curlywedge} \quad \simeq \quad \mathcal{A} \multimap \curlywedge$$
(where $\curlywedge$ is a particular automaton accepting no tree), with as expected $T \in \mathcal{L}(\mathcal{A}^{\curlywedge})$ iff $T \notin \mathcal{L}(\mathcal{A})$.

- Second, we show that the simulation operation $!(-)$ satisfies the *deduction rules* of the usual modality $!(-)$ of IMELL. Moreover, the symmetric monoidal product $(-) \otimes (-)$ is Cartesian on non-deterministic automata, so that the picture (1.3) is similar to the usual linear-non-linear adjunctions in models of IMELL. (Unfortunately, in our models the operation $!(-)$ is not a functor.[9])

As a consequence, we can redraw (1.3) as follows:



(1.5)

In other words, IMELL can provide an intermediate language between MSO and automata. A linear counterpart of MSO, call it LMSO, can be based on the following language:

$$\varphi, \psi \quad ::= \quad \alpha \quad | \quad \bot \quad | \quad \mathbf{I} \quad | \quad \varphi \otimes \psi \quad | \quad \varphi \multimap \psi \quad | \quad !\varphi \quad | \quad (\exists X)\varphi \qquad \text{(for } \alpha \in \text{At)}$$

This language must be seen as a refinement of MSO with finer-grained connectives which directly correspond to operations on automata. Given a deterministic automaton $\mathcal{A}(\alpha)$ for each atomic formula $\alpha \in \text{At}$, we associate a uniform automaton $\mathcal{A}(\varphi)$ to each LMSO formula φ.[10]

It would have been be natural to also consider the *additive* connectives & (conjunction) and $\oplus$ (disjunction) of linear logic, which do correspond to known constructions on alternating

---

[8]However, the IMLL-structure underlying our model differs from the usual IMLL-structure of simple games.

[9]It does not preserves composition, because of issues with positionality of strategies. Possible workarounds, left as future work, are discussed in §9.1.

[10]The usual projection operation $\widetilde{\exists}(-)$ can actually be extended to arbitrary automata, but is has its standard meaning only on non-deterministic ones. We elaborate on this in §7.2.3 and §8.2.

automata. However, the expected categorical properties of these connectives would require an extension of our setting that we leave for further work. With this in mind, the translation of MSO to non-deterministic automata induced by (1.4) factors via the map $(-)^{\mathrm{nd}} : \mathsf{MSO} \to \mathsf{LMSO}$ given by

$$\begin{array}{lcl} \alpha^{\mathrm{nd}} & := & !\alpha \\ (\neg\varphi)^{\mathrm{nd}} & := & !(\varphi^{\mathrm{nd}} \multimap \bot) \\ (\varphi \wedge \psi)^{\mathrm{nd}} & := & \varphi^{\mathrm{nd}} \otimes \psi^{\mathrm{nd}} \\ ((\exists X)\varphi)^{\mathrm{nd}} & := & (\exists X)\varphi^{\mathrm{nd}} \end{array}$$

while the translation of MSO to alternating automata factors via the map $(-)^{\mathrm{alt}} : \mathsf{MSO} \to \mathsf{LMSO}$ given by

$$\begin{array}{lcl} \alpha^{\mathrm{alt}} & := & \alpha \\ (\neg\varphi)^{\mathrm{alt}} & := & \varphi^{\mathrm{alt}} \multimap \bot \\ (\varphi \wedge \psi)^{\mathrm{alt}} & := & \varphi^{\mathrm{alt}} \otimes \psi^{\mathrm{alt}} \\ ((\exists X)\varphi)^{\mathrm{alt}} & := & (\exists X)!\varphi^{\mathrm{alt}} \end{array}$$

These factorizations are sound in the following sense, assuming the soundness of $\mathcal{A}(\alpha)$ for each atomic formula $\alpha \in \mathrm{At}$.

Proposition 1.5.1. *Let $(-)^{\mathrm{trans}}$ be either $(-)^{\mathrm{nd}}$ or $(-)^{\mathrm{alt}}$. A closed* MSO *formulae φ is true in the full $\mathfrak{D}$-ary tree if and only if $\mathcal{A}(\varphi^{\mathrm{trans}})$ accepts the unique $\mathbf{1}$-labeled $\mathfrak{D}$-ary tree.*

The image in (1.5) of the interpretation of LMSO formulae as automata gives rise to a polarized fragment of LMSO. The *deterministic* (notation $\varphi^{\pm}, \psi^{\pm}$) and the *(weakly) positive* (notation φ^{+}, ψ^{+}) formulae of LMSO are defined as

$$\begin{array}{lcl} \varphi^{\pm}, \psi^{\pm} & ::= & \mathbf{I} \mid \alpha \\ \varphi^{+}, \psi^{+} & ::= & \varphi^{\pm} \mid \bot \mid \psi^{\pm} \multimap \varphi^{+} \mid \varphi^{+} \otimes \psi^{+} \mid (\exists X)\varphi^{+} \mid !\varphi \end{array}$$

Hence positive formulae are interpreted as non-deterministic automata while deterministic formulae are interpreted as deterministic automata. Note that $\bot$ is positive since $\rotatebox[origin=c]{180}{$\bot$}$ is actually non-deterministic (see §7.2.2).

1.6. Computational Interpretation of Proofs. In our view, proposing LMSO as an intermediate system between MSO and automata should rely on a suitable computational interpretation of proofs, along the lines of the Curry-Howard proofs-as-programs correspondence. This in particular requires to devise deduction systems for MSO and LMSO, a delicate point that we leave for Part I (Chap. 3).

We explain here our view that runs of automata (or P-strategies in acceptance games) are relevant objects for a computational interpretation of LMSO proofs. We shall content ourselves for now with the deduction rules given in Fig. 1.1 for the purely logical part of LMSO (*i.e.* IMELL with existential quantifiers). This deduction system manipulates sequents of the form

$$\varphi_1, \dots, \varphi_n \vdash \varphi \tag{1.6}$$

We see these sequents with two different levels of interpretation. The first level interprets *provability*. Assuming the free variables of $\varphi_1, \dots, \varphi_n, \varphi$ are among $X_1, \dots, X_k$, if the sequent (1.6) is provable, then the (uniform) automaton $\mathcal{A}(\varphi)$ accepts a tree $T : \mathfrak{D}^* \to \mathbf{2}^k$ as soon as the (uniform) automata $\mathcal{A}(\varphi_1), \dots, \mathcal{A}(\varphi_n)$ all accept T.

The second level is the *computational* interpretation of *proofs* of the Curry-Howard correspondence. This is best exemplified with existential quantifications. The existential quantifications

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

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

$$\frac{}{\overline{\varphi} \vdash \mathbf{I}} \qquad \frac{\overline{\varphi}, \varphi, \psi \vdash \varphi'}{\overline{\varphi}, \varphi \otimes \psi \vdash \varphi'} \qquad \frac{\overline{\varphi} \vdash \varphi \qquad \overline{\psi} \vdash \psi}{\overline{\varphi}, \overline{\psi} \vdash \varphi \otimes \psi} \qquad \frac{\overline{\varphi}, \varphi \vdash \psi}{\overline{\varphi}, (\exists Z)\varphi \vdash \psi} \qquad \frac{\overline{\varphi} \vdash \varphi[Y/X]}{\overline{\varphi} \vdash (\exists X)\varphi}$$

$$\frac{\overline{\varphi}, \varphi \vdash \psi}{\overline{\varphi} \vdash \varphi \multimap \psi} \qquad \frac{\overline{\varphi} \vdash \varphi \qquad \overline{\psi}, \psi \vdash \psi'}{\overline{\varphi}, \overline{\psi}, \varphi \multimap \psi \vdash \psi'}$$

Figure 1.1.: Deduction Rules for LMSO (where Z is fresh).

of MSO are implemented by a *projection* operation on non-deterministic automata, that we present here in the usual setting of §1.3. Consider a non-deterministic automaton $\mathcal{A}$ over the alphabet $\Gamma \times \Sigma$. Its projection $\widetilde{\exists}_\Sigma \mathcal{A}$ is the non-deterministic automaton over Γ defined as $\mathcal{A}$ but with transition function

$$\begin{array}{lcccc} \partial_{\widetilde{\exists}_\Sigma \mathcal{A}} & : & Q_\mathcal{A} \times \Gamma & \longrightarrow & \mathcal{P}(\mathfrak{D} \to Q_\mathcal{A}) \\ & & (q, \mathtt{b}) & \longmapsto & \bigcup_{\mathtt{a} \in \Sigma} \partial_\mathcal{A}(q, (\mathtt{b}, \mathtt{a})) \end{array}$$

As expected, $\widetilde{\exists}_\Sigma \mathcal{A}$ accepts $T : \mathfrak{D}^* \to \Gamma$ iff there exists $U : \mathfrak{D}^* \to \Sigma$ such that $\mathcal{A}$ accepts $\langle T, U \rangle : \mathfrak{D}^* \to \Gamma \times \Sigma$.

Consider now a positive LMSO formula $\varphi(X)$. By *computational interpretation of proofs*, we mean that from a formal proof of the sequent

$$\vdash (\exists X)\varphi(X)$$

one should be able to extract a witness for the existential quantification $(\exists X)\varphi$, that is a $\mathbf{2}$-labeled tree accepted by $\mathcal{A}(\varphi)$ (seen for purposes of the current discussion as a usual non-deterministic automaton). Such witnesses can actually be extracted from the runs of $\widetilde{\exists}_\mathbf{2} \mathcal{A}(\varphi)$ on the unique tree $\mathbf{1} : \mathfrak{D}^* \to \mathbf{1}$. First note that a run ρ of a non-deterministic automaton $\mathcal{B}$ on a tree T defines a function $p \in \mathfrak{D}^* \longmapsto (q_d)_{q \in \mathfrak{D}} \in \partial_\mathcal{B}(\rho(p), T(p))$. It follows that given an accepting run ρ of $\widetilde{\exists}_\Sigma \mathcal{A}(\varphi)$ on $\mathbf{1}$, then from the induced function

$$p \in \mathfrak{D}^* \quad \longmapsto \quad (q_d)_{d \in \mathfrak{D}} \in \bigcup_{\mathtt{a} \in \mathbf{2}} \partial_{\mathcal{A}(\varphi)}(\rho(p), \mathtt{a})$$

one can get a $\mathbf{2}$-labeled tree T such that ρ is an accepting run of $\mathcal{A}(\varphi)$ on T.

In other words, *runs* of automata convey the kind of information one is usually interested in with computational interpretations of proofs. We will however rather rely on the more complex notions of acceptance games and strategies. There are two reasons for this choice. First, as discussed in §1.4 above, games give a smooth treatment of complementation of tree automata. The second reason, which we motivate with more details in Chap. 4, is that games and strategies are equipped with well-known categorical structures, which allow to easily define compositional interpretations of proofs.

1.7. Toward Realizability Interpretations of MSO. The ultimate motivation for our Curry-Howard approach to automata on infinite trees, together with the underlying decomposition of the translation of MSO formulae to tree automata via LMSO, is to provide realizability interpretations of MSO (in the spirit of e.g. [SU06, Koh08]).

The methodology behind our realizability interpretation targets interactive proofs systems, allowing possible human simplifications or decompositions of the goals given to automatic tools, and moreover to *combine* the corresponding witnessing strategies. Our motivation is that even if Rabin's Tree Theorem proves the existence of a decision procedures for MSO on infinite trees, there is (as far as we know) no working implementation of such procedures. The reason is that all known translations of formulae to tree automata involve at some stage the determinization of automata on ω-words (McNaughton's Theorem [McN66]), which is believed not to be amenable to tractable implementation (see e.g. [KV05]). We instead target semi-automatic approaches in which the user can delegate sufficiently simple subgoals to automatic non-emptiness checkers (solving parity games). The partial proof tree built by the user is then translated to a combinator able to compose the strategies obtained by the algorithms.

2. Preliminaries

This short Chapter gathers some well-known basic material on MSO, games and automata, which will be implicitly and explicitly used in Part I as well as Part II.

2.1. Automata on ω-Words. We briefly discuss here the case of ω-words, *i.e.* when $\mathfrak{D} = \mathbf{1}$, for which the decidability of MSO is much simpler than for infinite trees. The result in itself, which actually was proved not too long before Rabin's Theorem, is known as Büchi's Theorem.

Theorem 2.1.1 (Büchi [Büc62]). MSO *over ω-words is decidable.*

The original proof method of Thm. 2.1.1 consists, similarly as for Rabin's Tree Theorem, in translating formulae to Büchi automata. A *non-deterministic Büchi automaton* is a non-deterministic word automaton run over ω-words, and which accepts an ω-word if there exists an infinite run with infinitely many final states. Checking the non-emptiness of a non-deterministic Büchi automaton thus simply amounts to checking the existence of a reachable cycle containing a final state.

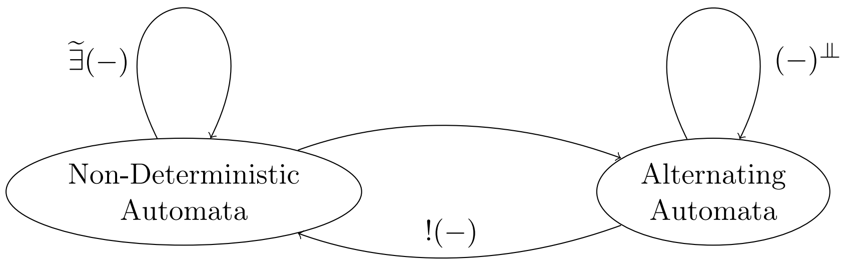
The crux of Büchi's Theorem 2.1.1 is the effective closure of Büchi automata under complement. It is remarkable that complementation of non-deterministic Büchi automata can be performed without going via intermediate deterministic automata. Besides, *deterministic* Büchi automata are strictly less expressive than non-deterministic ones. On the other hand, McNaughton's Theorem [McN66] states that non-deterministic Büchi automata can be translated to equivalent deterministic finite state automata, but equipped with stronger acceptance conditions. There are different variants of such conditions (*Muller*, *Rabin*, *Streett* or *parity* conditions, see e.g. [Tho97, GTW02]). All of them can specify states which *must not* occur infinitely often in an accepting run. Maybe the most widespread one are the parity conditions. A parity condition assumes a coloring of states by natural numbers, and a run is accepting if the least color seen infinitely often is even.

Theorem 2.1.2 (McNaughton [McN66]). *Each non-deterministic Büchi automaton is effectively equivalent to a deterministic parity automaton.*

We say that a set of ω-words over a given alphabet is *ω-regular* if it is the language of a parity automaton, or equivalently of a non-deterministic Büchi automaton. Via suitable representations of alphabets, being an ω-regular language is of course equivalent to being the set ω-words which satisfy a given MSO formula. McNaughton's Theorem 2.1.2 gives the important fact that each ω-regular language is a (finite) Boolean combination of $\mathbf{\Pi}^0_2$ sets.[1] This property is not apparent with non-deterministic Büchi automata (let alone MSO) because of the existential quantifications over infinite runs imposed by non-determinism.

2.2. Games on Graphs. Let us come back to the general picture (1.3) on tree automata we gave in §1.4:

[1] A $\mathbf{\Pi}^0_2$ set is a countable intersection of open sets.



(2.1)

While complementation $(-)^{\perp\!\!\perp}$ of alternating automata follows from determinacy of ω-regular games, the operation

$$!(-) \quad : \quad \text{Alternating Automata} \quad \longrightarrow \quad \text{Non-Deterministic Automata}$$

of the *Simulation Theorem* [MS87, EJ91, MS95] actually requires a stronger property. The point is intuitively the following (we refer to e.g. [MS95, Wal02] for details). For an alternating automaton $\mathcal{A}$, the non-deterministic automaton $!\mathcal{A}$ has to check for the existence of accepting P-strategies on $\mathcal{A}$. With non-deterministic automata, since O only chooses tree directions, the states appearing in a play of a P-strategy only depend on the tree positions and the strategy itself. As a consequence, P-strategies correspond to run trees, that is to state-labeled $\mathfrak{D}$-ary trees, so that existential quantifications over these can be expressed by the projection operation of non-deterministic automata. This is not the case with alternating automata. Because O can choose states as well as tree directions (*i.e.* choose among different possible states for the same tree direction), a P-strategy may not be representable as a labeled $\mathfrak{D}$-ary tree, and in particular, existential quantifications over these may not be expressible with the projection operation of non-deterministic automata.

The usual solution is provided by a suitable *positionality* constraint on strategies. The idea is that acceptance games are seen as graphs (actually dags), whose vertices are pairs of a tree position and a state, and that strategies are asked to only depend on the current graph vertex. This exactly amounts to ask strategies to be representable as $\mathfrak{D}$-ary trees on suitable alphabets.

We shall also be concerned with another (but related) aspect of games played on graphs, namely the *Büchi-Landweber Theorem* [BL69], which states that ω-regular games played on finite graphs are *effectively* determined (see §2.4).

We thus consider the following notion.

Definition 2.2.1 (Graph Games)**.** *A (rooted)* graph game *has the form* $G = (V_\mathsf{P}, V_\mathsf{O}, E, v^\imath, \mathcal{W})$ *where* V_P *and* V_O *are disjoint sets of resp.* P-positions *and* O-positions*,* $E \subseteq (V_\mathsf{P} \times V_\mathsf{O}) \cup (V_\mathsf{O} \times V_\mathsf{P})$ *is the* edge relation*, and, for* $V := V_\mathsf{P} + V_\mathsf{O}$*,* $v^\imath \in V$ *is the* game root *and* $\mathcal{W} \subseteq V^\omega$ *is the* winning condition*.*

Given a game G and vertices $v, w \in V$, we often write $v \to w$ (or even $v \to_G w$ or $v \to_E w$) to mean $(v, w) \in E$. For simplicity, we sometimes (but not always) assume our games to have no dead-end, in the sense that for each $v \in V$ there is some $w \in V$ such that $v \to w$. Given a graph game G and a vertex $v \in V$, we write G_v for the game defined as G but with initial position v:

$$G_v \quad := \quad (V_\mathsf{P},\, V_\mathsf{O},\, E,\, v,\, \mathcal{W})$$

When playing a graph game G with (say) $v^\imath \in V_\mathsf{P}$ is that, starting from P-position $v^\imath$, P plays a move $v_0 \in V_{\mathsf{O}}$ with $v^\imath \to v_0$, and then O plays a move $v_1 \in V_{\mathsf{P}}$ with $v_0 \to v_1$, and so on:

$$\begin{array}{ccccccccccccc} & \mathsf{P} & & \mathsf{O} & & \mathsf{P} & & \mathsf{O} & & \mathsf{P} & & \mathsf{O} & \\ v^\imath & \longrightarrow & v_0 & \longrightarrow & v_1 & \longrightarrow & \cdots & \longrightarrow & v_{2n-1} & \longrightarrow & v_{2n} & \longrightarrow & v_{2n+1} \quad \cdots \\ \Cap & & \Cap & & \Cap & & & & \Cap & & \Cap & & \Cap \\ V_{\mathsf{P}} & & V_{\mathsf{O}} & & V_{\mathsf{P}} & & & & V_{\mathsf{P}} & & V_{\mathsf{O}} & & V_{\mathsf{P}} \end{array}$$

We see P as playing *moves* in V_{O} from P-*positions* in V_{P}, *i.e.* as actually playing O-positions. This is made formal in the following construction of a simple game $A(G)$ in the sense of App. A from a graph game G. Given a graph game $G = (V_{\mathsf{P}}, V_{\mathsf{O}}, E, v^\imath, \mathcal{W})$, we define the simple game with winning

$$A(G) \quad := \quad (V_{\mathsf{O}},\, V_{\mathsf{P}},\, L,\, \mathcal{W})$$

where $L \subseteq (V_{\mathsf{P}} + V_{\mathsf{O}})^\omega$ is the smallest set containing ε, such that $v \in L$ whenever $v^\imath \to v$, and such that $\overline{v}.v.w \in L$ whenever $\overline{v}.v \in L$ and $v \to w$. Note that $A(G)$ is positive if $v^\imath \in V_{\mathsf{P}}$ and negative if $v^\imath \in V_{\mathsf{O}}$, and that the P-*moves* of $A(G)$ are the O-*positions* of G. In the following, when speaking about (general) strategies on the graph game G, we always mean strategies in the simple game $A(G)$. As expected, if G has no dead-end then a P-strategy on G is winning provided all its infinite plays (from $v^\imath$) belong to $\mathcal{W}$, while an O-strategy is winning if all its infinite plays (from $v^\imath$) avoid $\mathcal{W}$.

In the context of MSO and automata on infinite words or trees, one is primarily interested in ω-regular games. An *ω-regular game* is a graph game G equipped with a coloring $\mathsf{c} : V \to C$, where C is a finite set, and such that $\mathcal{W}$ is induced by an ω-regular language $W \subseteq C$, in the sense that $(v_n)_n \in \mathcal{W}$ iff $(\mathsf{c}(v_n))_n \in W$. Since ω-regular sets are Boolean combinations of $\mathbf{\Pi}^0_2$-sets, it follows from Davis's Theorem [Dav64] that ω-regular games are determined (from every position, either P or O has a winning strategy).

We now turn to *positional* strategies.

Definition 2.2.2 (Positional Strategy)**.** *Fix a graph game G. A positional P-strategy (resp. positional O-strategy) on G is a function $\sigma : V_{\mathsf{P}} \to V_{\mathsf{O}}$ (resp. $\sigma : V_{\mathsf{O}} \to V_{\mathsf{P}}$) such that $v \to \sigma(v)$ whenever $v \to w$ for some w.*

Positional winning strategies are known to exist for *parity games*, which are ω-regular games where C is of the form $\{0, \dots, k\}$ for some $k \in \mathbb{N}$, and where $\mathcal{W} \subseteq V^\omega$ consists of those $(v_n)_n$ such that the least number occurring infinitely often in $(\mathsf{c}(v_n))_n$ is even.

Theorem 2.2.3 ([EJ91])**.** *If G is a parity game, then either P or O has a positional winning strategy.*

Acceptance games are often defined as graph games (see e.g. [Tho97, GTW02, PP04]).

Example 2.2.4 (Acceptance Games)**.** *Consider an alternating automaton $\mathcal{A} : \Sigma$ and a tree $T : \mathfrak{D}^* \to \Sigma$. The* acceptance game *$G(\mathcal{A}, T)$ is the rooted graph game*

$$G(\mathcal{A}, T) \quad := \quad \big(V_{\mathsf{P}},\, V_{\mathsf{O}},\, E,\, (\varepsilon, q^\imath_{\mathcal{A}}),\, \mathcal{W}\big)$$

whose positions are

$$V_{\mathsf{P}} \quad := \quad \mathfrak{D}^* \times Q_{\mathcal{A}} \qquad \textit{and} \qquad V_{\mathsf{O}} \quad := \quad \mathfrak{D}^* \times Q_{\mathcal{A}} \times \mathcal{P}(Q_{\mathcal{A}} \times \mathfrak{D})$$

whose edges are given by

$$\begin{array}{lrcll} \textit{from } V_{\mathsf{P}} \textit{ to } V_{\mathsf{O}} : & (p,\, q) & \xrightarrow{\mathsf{P}} & (p,\, q,\, \gamma) & \textit{iff} \quad \gamma \in \partial_{\mathcal{A}}(q, T(p)) \\ \textit{from } V_{\mathsf{O}} \textit{ to } V_{\mathsf{P}} : & (p,\, q,\, \gamma) & \xrightarrow{\mathsf{O}} & (p.d,\, q') & \textit{iff} \quad (q', d) \in \gamma \end{array}$$

and whose winning condition $\mathcal{W}$ is given by

$$(\varepsilon, q_0) \cdot (\varepsilon, \gamma_0) \cdot (p_1, q_1) \cdot \ldots \cdot (p_n, q_n) \cdot (p_n, \gamma_n) \cdot \ldots \quad \in \mathcal{W} \qquad \textit{iff} \qquad (q_i)_{i \in \mathbb{N}} \in \Omega$$

If $\mathcal{A}$ is a parity automaton, with acceptance condition $\Omega_{\mathcal{A}} \subseteq Q_{\mathcal{A}}^{\omega}$ generated from a coloring $\mathsf{c} : Q_{\mathcal{A}} \to \{0, \ldots, k\}$, then $G(\mathcal{A}, T)$ is a parity game for the coloring $\tilde{\mathsf{c}}$ defined as

$$\begin{array}{rcl} \tilde{\mathsf{c}}(p, q) & := & \mathsf{c}(q) \\ \tilde{\mathsf{c}}(p, q, \gamma) & := & \mathsf{c}(q) \end{array}$$

The case of $G(\mathcal{A}, T)$ for $\mathcal{A}$ a non-deterministic automaton in the sense of §1.3 is similar and omitted.

Remark 2.2.5 (On Ex. 2.2.4 vs §1.4)**.** *The simple game $A(G(\mathcal{A}, T))$ actually differs from the acceptance game $\mathcal{G}(\mathcal{A}, T)$ sketched in §1.4. We have*

$$A(G(\mathcal{A}, T)) \quad = \quad \big(\mathfrak{D}^* \times \mathcal{P}(Q_{\mathcal{A}} \times \mathfrak{D}),\ \mathfrak{D}^* \times Q_{\mathcal{A}},\ L,\ \mathcal{W}\big)$$

where L consists of sequences of the form

$$\begin{array}{rrcl} & s & = & (\varepsilon, \gamma_0) \cdot (d_1, q_1) \cdot (d_1, \gamma_1) \cdot \ldots \cdot (d_1 \ldots d_n, q_n) \\ \textit{or} & s & = & (\varepsilon, \gamma_0) \cdot (d_1, q_1) \cdot (d_1, \gamma_1) \cdot \ldots \cdot (d_1 \ldots d_n, q_n) \cdot (d_1 \ldots d_n, \gamma_n) \end{array}$$

where $n \geq 0$, $(q_{k+1}, d_{k+1}) \in \gamma_k$ and $\gamma_k \in \partial_{\mathcal{A}}(q_k, T(d_1 \cdot \ldots \cdot d_k))$ with $q_0 := q^{\imath}_{\mathcal{A}}$.

But there is some redundancy in L since sequences of tree positions $d_1, \ldots, d_k$ do not need to be recorded in moves. This leads us to often prefer (as in §1.4) the positive simple game

$$\mathcal{G}(\mathcal{A}, T) \quad := \quad \big(\mathcal{P}(Q_{\mathcal{A}} \times \mathfrak{D}),\ Q_{\mathcal{A}} \times \mathfrak{D},\ L_{\mathcal{A}(T)},\ \mathcal{W}_{\mathcal{A}(T)}\big)$$

whose legal plays $s \in L_{\mathcal{A}(T)}$ are sequences of the form

$$\begin{array}{rrcl} & s & = & \gamma_0 \cdot (q_1, d_1) \cdot \gamma_1 \cdot \ldots \cdot (q_n, d_n) \\ \textit{or} & s & = & \gamma_0 \cdot (q_1, d_1) \cdot \gamma_1 \cdot \ldots \cdot (q_n, d_n) \cdot \gamma_n \end{array}$$

where $n \geq 0$, $(q_{k+1}, d_{k+1}) \in \gamma_k$ and $\gamma_k \in \partial_{\mathcal{A}}(q_k, T(d_1 \cdot \ldots \cdot d_k))$ with $q_0 := q^{\imath}_{\mathcal{A}}$. The winning plays $\chi \in \mathcal{W}_{\mathcal{A}(T)}$ are generated from the acceptance condition $\Omega_{\mathcal{A}}$ in the expected way: $\mathcal{W}_{\mathcal{A}(T)} \subseteq (\mathcal{P}(Q_{\mathcal{A}} \times \mathfrak{D}) \cdot (Q_{\mathcal{A}} \times \mathfrak{D}))^{\omega}$ consists of those infinite sequences

$$\chi \quad = \quad \gamma_0 \cdot (q_1, d_1) \cdot \gamma_1 \cdot \ldots \cdot (q_n, d_n) \cdot \ldots$$

such that $(q_k)_{k \in \mathbb{N}} \in \Omega_{\mathcal{A}}$ (where $q_0 := q^{\imath}_{\mathcal{A}}$).

In some circumstances (in particular when positionality matters), we see the rooted graph $G(\mathcal{A}, T)$ as a quotient of the tree $L_{\mathcal{A}(T)}$ for the obvious graph morphism $\mathsf{pos} : L_{\mathcal{A}(T)} \to G(\mathcal{A}, T)$.

Remark 2.2.6 (On Ex. 2.2.4 vs [Wal02])**.** *In the acceptance games of [Wal02], the set of O positions is $\mathfrak{D}^* \times \mathcal{P}(Q_{\mathcal{A}} \times \mathfrak{D})$. Hence O positions do not contain state information, contrary the formulation of Ex. 2.2.4 (which follows [DR19]). This difference is due to different positionality assumptions when proving the correctness of the complementation operation $(-)^{\perp\!\!\!\perp}$, i.e. that for a labeled tree T, we have $T \in \mathcal{L}(\mathcal{A}^{\perp\!\!\!\perp})$ iff $T \notin \mathcal{L}(\mathcal{A})$.*

The point occurs more precisely in the left-to-right implication, namely that a P strategy in $G(\mathcal{A}^{\perp\!\!\!\perp}, T)$ induces an O strategy in $G(\mathcal{A}, T)$. The setting of [Wal02] allows for non positional strategies, so that the O strategy in $G(\mathcal{A}, T)$ knows the current state of $\mathcal{A}$ from its history. On the other hand, the formulation of Ex. 2.2.4 is tailored toward [DR19] and as such should restrict to positional strategies. But a positional O strategy in $G(\mathcal{A}, T)$ cannot know the current state of $\mathcal{A}$ from its history, and thus should get it from its current position.

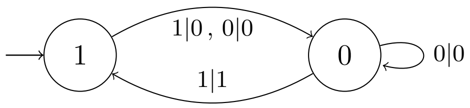


Figure 2.1.: A Mealy machine.

2.3. Causal and Eager Functions. A stream function $F : \Sigma^\omega \to \Gamma^\omega$ is causal if it can produce a prefix of length n of its output from a prefix of length n of its input. Hence F is causal if it is induced by a map $f : \Sigma^+ \to \Gamma$ as follows:

$$F(B)(n) \quad = \quad f(B(0) \cdot \ldots \cdot B(n)) \qquad \text{(for all } B \in \Sigma^\omega \text{ and all } n \in \mathbb{N}\text{)}$$

The finite-state (f.s.) causal functions are those induced by Mealy machines. A Mealy machine $\mathcal{M} : \Sigma \to \Gamma$ is a DFA over input alphabet Σ, which is moreover equipped with an output function $\lambda : Q_\mathcal{M} \times \Sigma \to \Gamma$ (where $Q_\mathcal{M}$ is the state set of $\mathcal{M}$). Writing $\partial^* : \Sigma^* \to Q_\mathcal{M}$ for the iteration of the transition function ∂ of $\mathcal{M}$ from its initial state, $\mathcal{M}$ induces a causal function via the map $\Sigma^+ \to \Gamma$, $\bar{\mathtt{a}}.\mathtt{a} \mapsto \lambda(\partial^*(\bar{\mathtt{a}}), \mathtt{a})$.

We write $F : \Sigma \to_{\mathbf{S}} \Gamma$ (resp. $F : \Sigma \to_{\mathbf{M}} \Gamma$) to mean that F is a causal (resp. f.s. causal) function from $\Sigma^\omega \to \Gamma^\omega$.

Example 2.3.1. *(a) Usual functions $\Sigma \to \Gamma$ lift to (pointwise, one-state) maps $\Sigma \to_{\mathbf{M}} \Gamma$. For instance, the identity $\Sigma \to_{\mathbf{M}} \Sigma$ is induced by the Mealy machine with $\langle \partial, \lambda \rangle : (-, \mathtt{a}) \mapsto (-, \mathtt{a})$.*

(b) Causal functions $\mathbf{1} \to_{\mathbf{S}} \Sigma$ correspond exactly to ω-words $B \in \Sigma^\omega$.

(c) The machine $\mathcal{M} : \mathbf{2} \to \mathbf{2}$ displayed on Fig. 2.1 (where a transition $\mathtt{a}|\mathtt{b}$ outputs $\mathtt{b}$ from input $\mathtt{a}$), taken from [Tho08], realizes the causal function $F : \mathbf{2} \to_{\mathbf{M}} \mathbf{2}$ such that

$$F(B)(n) \quad = \quad \begin{cases} 0 & \text{if } n = 0 \text{ or } F(B)(n-1) = 1 \\ B(n) & \text{otherwise} \end{cases}$$

A causal function $\Sigma \to_{\mathbf{S}} \Gamma$ is *eager* if it can produce a prefix of length $n+1$ of its output from a prefix of length n of its input. More precisely, an eager $F : \Sigma \to_{\mathbf{S}} \Gamma$ is induced by a map $f : \Sigma^* \to \Gamma$ as

$$F(B)(n) \quad = \quad f(B(0) \cdot \ldots \cdot B(n-1)) \qquad \text{(for all } B \in \Sigma^\omega \text{ and all } n \in \mathbb{N}\text{)}$$

Isolating eager functions allows a proper treatment of strategies in games, as a P-strategy σ in a full positive game (U, X) (see Ex. A.0.2) is a function $X^* \to U$, and thus can be seen as an eager function from X to U.

Finite-state eager functions are those induced by eager (Moore) machines (see also [FJR11]). An eager machine $\mathcal{E} : \Sigma \to \Gamma$ is a Mealy machine $\Sigma \to \Gamma$ whose output function $\lambda : Q_\mathcal{E} \to \Gamma$ does not depend on the current input letter. An eager machine $\mathcal{E} : \Sigma \to \Gamma$ induces a f.s. eager function via the map $\Sigma^* \to \Gamma$, $\bar{\mathtt{a}} \mapsto (\lambda_\mathcal{E}(\partial^*_\mathcal{E}(\bar{\mathtt{a}}))$.

We write $F : \Sigma \to_{\mathbf{E}} \Gamma$ when $F : \Sigma \to_{\mathbf{S}} \Gamma$ is eager and $F : \Sigma \to_{\mathbf{EM}} \Gamma$ when F is f.s. eager. All functions $F : \Sigma \to_{\mathbf{M}} \mathbf{1}$, and more generally, constants functions $F : \Sigma \to_{\mathbf{S}} \Gamma$ are eager. Note also that a f.s. causal function which is additionally eager is a f.s. eager function. On the other hand, if $F : \Sigma \to_{\mathbf{EM}} \Gamma$ is induced by an eager machine $\mathcal{E}$ then F is finite-state causal as being induced by the Mealy machine with same states and transitions as $\mathcal{E}$, and with output function $(q, \mathtt{a}) \mapsto \lambda_\mathcal{E}(q)$.

2.4. The Büchi-Landweber Theorem and Finite-State Strategies. The Büchi-Landweber Theorem states that ω-regular games played on finite graphs are *effectively* determined, in the sense that one can decide who is the winner, and moreover that the winner always has a *finite state* winning strategy.

We rely on the Büchi-Landweber Theorem for two reasons. First, it gives *Rabin's Basis Theorem*, namely the decidability of non-emptiness for tree automata, witnessed by finite-state accepting strategies and regular accepted trees. Second, Büchi-Landweber Theorem gives the general theoretical solutions to *Church's Synthesis* (see §8.2.2).

Consider a graph game $G = (V_\mathsf{P}, V_\mathsf{O}, E, v^\imath, \mathcal{W})$ with V_P and V_O finite. We assume w.l.o.g. G to have no dead-end and $v^\imath \in V_\mathsf{P}$. A P (resp. O) strategy in G is defined to be a P (resp. O) strategy in $A(G)$, and can be represented as a function

$$V_\mathsf{P}^* \longrightarrow V_\mathsf{O} \qquad \text{resp.} \qquad V_\mathsf{O}^+ \longrightarrow V_\mathsf{P}$$

Hence, P-strategies in G can be seen as eager functions $V_\mathsf{P} \to_{\mathbf{E}} V_\mathsf{O}$, while O-strategies can be seen as causal functions $V_\mathsf{O} \to_{\mathbf{M}} V_\mathsf{P}$. As a consequence, it is natural to say that a *finite-state* P-strategy (resp. O-strategy) on G is a *finite-state* eager function $V_\mathsf{P} \to_{\mathbf{EM}} V_\mathsf{O}$ (resp. causal function) $V_\mathsf{O} \to_{\mathbf{M}} V_\mathsf{P}$.

Theorem 2.4.1 (Büchi-Landweber [BL69])**.** *Let G be an ω-regular game as above. It is decidable whether P or O has a winning strategy in G. Moreover, the winner always has a finite-state winning strategy.*

Part I.

Axiomatizations

3. Axiomatizations

Part I of this document, which actually amounts to the present chapter, covers material concerning axiomatizations of MSO.

This line of works was initiated by Büchi and Siefkes, who gave axiomatizations of MSO on various classes of linear orders, including ω-words (see e.g. [Sie70, BS73]). These works essentially rely on formalizations of automata in the logic. A major result in the axiomatic treatment of logics over infinite structures is Walukiewicz's proof of completeness of Kozen's axiomatization of the modal μ-calculus [Wal00] (see also [AL17] for an alternative recent proof of this result and [Kai95, Dou17] for the case of ω-words). Another trend relies on model-theoretic techniques. For instance [tCF10, GtC12] give complete axiomatizations of MSO and the modal μ-calculus over finite trees, and [SV10] gives a model-theoretic completeness proof for the flat fragment of the modal μ-calculus. An attractive feature of model-theoretic completeness proofs for the aforementioned logics is that they allow elegant reformulations of algebraic approaches to these logics. Unfortunately, in the case of MSO over infinite trees, the only known algebraic approach [Blu13] seems yet too complex to be easily formalized, so that one has to directly formalize a translation of formulae to automata in the axiomatic theory.

In the case of ω-words, the axiom system essentially consists of Second-Order Peano Arithmetic (PA2) restricted the language of MSO [Sie70]. Thanks to the model theoretic setting used in [GtC09] for the case of finite trees, we proposed in [Rib12] a much shorter and (we believe) simpler argument than [Sie70] for the completeness of MSO on ω-words.

We attacked the case of infinite trees with Anupam Das in [DR15], which unfortunately contains an important (and we believe irreparable) flaw in the positional determinacy argument. Our analysis of the cause of the mistake is the following. The fact that MSO is decidable implies that it does not allow for the usual primitive recursive codings, which make formal reasoning in PA2 humanly feasible (see e.g. [Sim10]). To circumvent this, we proposed in [DR15] a syntactic sugar, essentially allowing for the uniform manipulation of (hereditarily) finite sets, but the version used in [DR15] was too rough to allow for proper checks of the statements written in it. We hope to have resolved this problem with the system FSO (for *Functional Second-Order Logic*) of [DR19]. While the results of [DR19] are deceptive in terms of axiomatizations (we basically assume the positional determinacy of parity games as an axiom), we nevertheless believe that the theory of games and automata formalized there could serve as a solid ground for further works on this question. We do not know yet whether the axiomatization of MSO proposed in [DR15] is complete or not.

This Chapter is organized as follows. We begin in §3.1 with MSO($\mathfrak{D}$), an axiomatic version of MSO inspired from PA2, which, as mentioned above, happens to be complete on ω-words (*i.e.* when $\mathfrak{D} = \mathbf{1}$). We then survey the system FSO($\mathfrak{D}$) of [DR19], as well as the key points of the formalization (§3.2). The material of §3.3 and §3.4 is unpublished and completely prospective: §3.3 covers a failed attempt to prove the *incompleteness* of MSO($\mathbf{2}$) (*i.e.* MSO over the infinite binary tree), while §3.4 sketches some ongoing work aimed at improving the current complete axiom system for MSO($\mathfrak{D}$).

Excepted in §3.3 and §3.4, this Chapter contains few details. We also do not cover the approach of [Rib13]. The idea, there, was to propose a forcing based approach namely to McNaughton's

Theorem [McN66] seen as the reduction of MSO on ω-words to *weak* MSO (where quantifications are restricted to finite sets). The hope was, in the line of [Kri11, Miq11], to then obtain a Curry-Howard approach to MSO. We do not discuss further this work here, since in our (current) view, it is largely superseded, w.r.t. its original Curry-Howard motivation, by the approach developed in Part II. We also do not detail the model-theoretic approach of [Rib12] because it is used nowhere else in this document. As for [DR19], we just give a mere outline, since the work is too long and technical to be presented here.

3.1. The Logic $\mathsf{MSO}(\mathfrak{D})$

We present here a formulation of MSO as a two-sorted logic, with, as mentioned in §1.2, one sort of individuals (intended to range over $\mathfrak{D}^*$) and one sort of monadic predicates (intended to range over $\mathcal{P}(\mathfrak{D}^*)$). The language of the resulting system $\mathsf{MSO}(\mathfrak{D})$ is defined in §3.1.1, while §3.1.2 gives a basic set of axioms for $\mathsf{MSO}(\mathfrak{D})$ corresponding to those of [DR15]. The corresponding notion of Henkin model is briefly presented in §3.1.3 (we do not technically require them, but they provide the right setting for some discussions). Then §3.1.4 deals with the specialization of $\mathsf{MSO}(\mathfrak{D})$ to ω-words, which is known to be complete.

3.1.1. The Language of $\mathsf{MSO}(\mathfrak{D})$.

The language of MSO given in §1.2 is well adapted when discussing translations to automata, but is less convenient when devising axiomatic systems. We think such axiomatic systems as subsystems or variants of Second-Order Peano Arithmetic (see e.g. [Sim10]), and as such, it is customary to have a primitive notion of individuals (intended to range over tree positions $p \in \mathfrak{D}^*$). Also, as common when discussing axiomatic second-order systems, we see second-order logic as a two-sorted first-order logic (see e.g. [Sha91, Sim10, Rib12] for comments on this). So formally $\mathsf{MSO}(\mathfrak{D})$ has one sort of *individuals*, with variables x, y, z, etc., and one sort of *monadic predicates*, with variables X, Y, Z, etc.

$\mathsf{MSO}(\mathfrak{D})$ should also be equipped with means of speaking, for $d \in \mathfrak{D}$, of the dth successor $p.d$ of a tree position $p \in \mathfrak{D}^*$. This can be done in two ways, either with unary function symbols $\mathsf{S}_d(-)$ on individuals, or with binary relations symbols $\mathsf{S}_d(-,-)$. The former, which is in line with the tradition of second-order arithmetic, is clearly more convenient when writing down concrete formulae, and is indeed the format adopted in [Sie70, DR19]. On the other hand, it induces a supplementary step when translating formulae to automata, and makes more difficult the connection with the versions of MSO used in Chap. 8 We assume this latter option here, while whenever convenient we shall use formulae containing terms with function symbols as defined formulae. We follow the same policy for the root position $\varepsilon \in \mathfrak{D}^*$, and assume a primitive unary predicate $\mathsf{R}(-)$ (for "*root*"). Using monadic second-order quantifications, we can have equality on individuals and the (strict) prefix order as defined formulae. For the sake of convenience, we take them as primitive connectives (this contrasts with [Rib12] for equality).

The choice of primitive logical connectives for $\mathsf{MSO}(\mathfrak{D})$ is not crucial here, and we shall simply assume a convenient set of connectives.

Definition 3.1.1 (The Language of $\mathsf{MSO}(\mathfrak{D})$)**.** *The formulae of* $\mathsf{MSO}(\mathfrak{D})$ *are given by*

$$\begin{array}{rcl} \varphi, \psi & ::= & \top \mid \bot \mid x \mathrel{\dot\in} X \mid x \mathrel{\dot=} y \mid x \mathrel{\dot<} y \mid \mathsf{R}(x) \mid \mathsf{S}_d(x,y) \qquad \textit{(for } d \in \mathfrak{D}\textit{)} \\ & \mid & \varphi \wedge \psi \mid \varphi \vee \psi \mid \psi \to \varphi \\ & \mid & (\exists x)\varphi \mid (\forall x)\varphi \mid (\exists X)\varphi \mid (\forall X)\varphi \end{array}$$

$$(\forall x)(\forall y)(\forall z) \textstyle\bigwedge_{d \in \mathfrak{D}} \big(\mathsf{S}_d(x,y) \;\to\; \mathsf{S}_d(x,z) \;\to\; y \doteq z\big)$$

$$(\forall x)(\forall y)(\forall z) \textstyle\bigwedge_{d \neq d'} \big(\mathsf{S}_d(x,y) \;\to\; \mathsf{S}_{d'}(x,z) \;\to\; \neg(y \doteq z)\big)$$

$$(\forall x)\neg(x \mathrel{\dot{<}} x) \qquad\qquad (\forall x)(\forall y)(\forall z)\big(x \mathrel{\dot{<}} y \;\to\; y \mathrel{\dot{<}} z \;\to\; x \mathrel{\dot{<}} z\big)$$

$$(\forall x)(\forall z)(\mathsf{R}(z) \;\to\; z \mathrel{\dot{\leq}} x) \qquad\qquad (\forall x)(\forall y)(\forall z) \textstyle\bigwedge_{d \in \mathfrak{D}} \big(\big[\mathsf{S}_d(z,y) \;\wedge\; x \mathrel{\dot{<}} y\big] \;\leftrightarrow\; x \mathrel{\dot{\leq}} z\big)$$

Figure 3.1.: Tree Axioms of $\mathsf{MSO}(\mathfrak{D})$.

We use the following notations and defined formulae:

$$\begin{array}{rcl} X(y) & := & (y \mathrel{\dot{\in}} X) \\ (x \mathrel{\dot{\leq}} y) & := & x \mathrel{\dot{<}} y \;\vee\; x \doteq y \\ \varphi \leftrightarrow \psi & := & (\varphi \to \psi) \;\wedge\; (\psi \to \varphi) \end{array}$$

Notation 3.1.2. *We shall use individual terms*

$$\mathtt{t}, \mathtt{u} \;::=\; x \;\mid\; \dot{\varepsilon} \;\mid\; \mathsf{S}_d(x) \qquad\qquad (\textit{for } d \in \mathfrak{D})$$

whenever convenient. Formulae written with such terms are translated to $\mathsf{MSO}(\mathfrak{D})$ *formulae in the obvious way. For instance, the formula* $\mathtt{t} \doteq x$ *is defined by induction on* $\mathtt{t}$ *as follows:*

$$\begin{array}{rcl} (\dot{\varepsilon} \doteq x) & := & (\exists z)(\mathsf{R}(z) \wedge z \doteq x) \\ (\mathsf{S}_d(\mathtt{t}) \doteq x) & := & (\exists z)(\mathtt{t} \doteq z \;\wedge\; \mathsf{S}_d(z,x)) \end{array}$$

3.1.2. The Theory $\mathsf{MSO}(\mathfrak{D})$. We assume for $\mathsf{MSO}(\mathfrak{D})$ deduction for two-sorted classical first-order logic. The axioms of $\mathsf{MSO}(\mathfrak{D})$ consist of the following:

- *Equality on Individuals:*

$$(\forall x)(x \doteq x) \qquad \text{and} \qquad (\forall x)(\forall y)\big(x \doteq y \;\longrightarrow\; \varphi[x/z] \;\longrightarrow\; \varphi[y/z]\big) \qquad (\text{for each } \varphi)$$

- The *Tree Axioms* of Fig. 3.1.

- *Comprehension Scheme:*

$$(\exists X)(\forall y)\big[y \mathrel{\dot{\in}} X \;\longleftrightarrow\; \varphi\big] \qquad\qquad (\text{for each } \varphi, \text{ with } X \text{ not free in } \varphi)$$

- *Induction Axiom:*

$$(\forall X)\left((\forall y)\big[\mathsf{R}(y) \;\to\; X(y)\big] \;\to\; \bigwedge_{d \in \mathfrak{D}} (\forall y)(\forall z)\big[\mathsf{S}_d(y,z) \;\to\; X(y) \;\to\; X(z)\big] \;\to\; (\forall y)X(y)\right)$$

We write $\vdash_{\mathsf{MSO}(\mathfrak{D})}$ for deduction in $\mathsf{MSO}(\mathfrak{D})$.

3.1.3. Henkin Models. We recall here Henkin models and completeness in the context of $\mathsf{MSO}(\mathfrak{D})$. An Henkin structure for the language of $\mathsf{MSO}(\mathfrak{D})$ has the form

$$\mathcal{M} \quad = \quad (\mathcal{M}^{\iota}, \mathcal{M}^{o}, \mathsf{R}_{\mathcal{M}}, (\mathsf{S}_{\mathcal{M},d})_{d\in\mathfrak{D}}, <_{\mathcal{M}})$$

where $\mathcal{M}^{\iota}$ is the set of *individuals*, $\mathcal{M}^{o} \subseteq \mathcal{P}(\mathcal{M}^{\iota})$ is the set of (monadic) *predicates*, $\mathsf{R}_{\mathcal{M}}$ is an element of $\mathcal{M}^{\iota}$, and $\mathsf{S}_{\mathcal{M},d}$ and $<_{\mathcal{M}}$ are binary relations on $\mathcal{M}^{\iota}$.

In particular, *the standard model* of $\mathsf{MSO}(\mathfrak{D})$ is

$$\mathfrak{T} \quad := \quad (\mathfrak{D}^*, \mathcal{P}(\mathfrak{D}^*), \mathsf{R}, (\mathsf{S}_d)_{d\in\mathfrak{D}}, <)$$

where R holds on $p \in \mathfrak{D}^*$ iff p is the empty sequence $\varepsilon \in \mathfrak{D}^*$, the relation $\mathsf{S}_d(p, q)$ (resp. $p < q$) holds iff $q = p.d$ (resp. iff p is a strict prefix of q).

The formulae of $\mathsf{MSO}(\mathfrak{D})$ are interpreted in Henkin models $\mathcal{M}$ as expected: individuals range over $\mathcal{M}^{\iota}$, (monadic) predicates range over $\mathcal{M}^{o}$, equality on individuals is interpreted as equality in $\mathcal{M}^{\iota}$, membership $\dot{\in}$ is interpreted as usual set membership $\in$, and S_d (resp. $\dot{<}$) is interpreted as $\mathsf{S}_{\mathcal{M},d}$ (resp. $<_{\mathcal{M}}$).

An Henkin structure $\mathcal{M}$ is a *model of* $\mathsf{MSO}(\mathfrak{D})$ if all the axioms of $\mathsf{MSO}(\mathfrak{D})$ hold in $\mathcal{M}$. As usual (see e.g. [Sha91]), $\mathsf{MSO}(\mathfrak{D})$ is complete w.r.t. its Henkin models:

Theorem 3.1.3 (Henkin Completeness)**.** *Given a closed* $\mathsf{MSO}(\mathfrak{D})$ *formula* φ*, if* φ *holds in all models of* $\mathsf{MSO}(\mathfrak{D})$*, then* $\vdash_{\mathsf{MSO}(\mathfrak{D})} \varphi$*.*

3.1.4. The Case of ω-Words. When $\mathfrak{D} = \mathbf{1}$, we write MSO^{ω} for $\mathsf{MSO}(\mathfrak{D})$ and $\mathfrak{N}$ for the standard model $\mathfrak{T}$. Note that $\mathfrak{N}^{\iota} = \mathbf{1}^* \simeq \mathbb{N}$ and $\mathfrak{N}^{o} \simeq \mathcal{P}(\mathbb{N})$; the atomic predicate R now specifies the natural number 0, and we write it Z. It is known since [Sie70] that MSO^{ω} completely axiomatizes $\mathfrak{N}$. In other words, all Henkin models of MSO^{ω} are equivalent and $\vdash_{\mathsf{MSO}^{\omega}}$ is a complete theory.

Theorem 3.1.4 (Siefkes [Sie70])**.** MSO^{ω} *is complete.*

The original proof of [Sie70] goes via a formalization of Büchi's Theorem [Büc62] (Thm. 2.1.1) in the formal system MSO^{ω}. While this approach is necessarily quite laborious, we noticed in [Rib12] that by adapting the method of [GtC09, GtC12] (based on tools from finite-model theory), it was possible to mimic the usual algebraic proof of decidability of MSO (see e.g. [PP04]), resulting in a much shorter (and we believe simpler) argument than [Sie70]. In both cases, the proofs follow from the fact that an infinite Ramsey's Theorem for pairs is provable in MSO^{ω} (for colorings defined by MSO^{ω} formulae). We refer to [KMPS16] for a calibration in second-order arithmetic of the proof-theoretical strength of the decidability of MSO^{ω}.

3.2. A Functional Extension of $\mathsf{MSO}(\mathfrak{D})$ on Infinite Trees

Devising a complete axiomatization of MSO on infinite trees is much more complex than for ω-words. We developed a first approach in [DR19], that we outline here.

First, since the only known algebraic approach to MSO on infinite trees [Blu13] seems unfortunately too complex to be easily formalized, it seems currently not reasonable to try to extend [GtC12, Rib12] to the case of infinite trees. As a consequence, we chose in [DR19] to proceed similarly as [Sie70] did for ω-word, namely to formalize in $\mathsf{MSO}(\mathfrak{D})$ a theory of automata on infinite trees. Such a formalization, which is necessarily much heavier than [Sie70], would be quite painful to be carried out in the language of $\mathsf{MSO}(\mathfrak{D})$. This is why [DR19] introduced

$\mathsf{FSO}(\mathfrak{D})$ (for *Functional Second-Order* Logic), an extension of $\mathsf{MSO}(\mathfrak{D})$ with built in facilities to manipulate finite sets and labeled trees. Essentially, $\mathsf{FSO}(\mathfrak{D})$ extends $\mathsf{MSO}(\mathfrak{D})$ with a sort of (hereditary) finite sets equipped with bounded quantification, and allows for the direct manipulation of $\mathfrak{D}$-ary trees labeled over (hereditary) finite sets. The formal definition of $\mathsf{FSO}(\mathfrak{D})$ in [DR19] is actually the result of a compromise between the flexibility of the system and the transparency of its translation to $\mathsf{MSO}(\mathfrak{D})$. This compromise leads us to some syntactic subtleties for which we refer to [DR19], and we shall only give here an outline of the system in the setting of §3.1.1.

The system $\mathsf{FSO}(\mathfrak{D})$ is sketched in §3.2.1, while §3.2.2 outlines the formalization accomplished in [DR19]. Recall that the set V_ω of *hereditarily finite sets* (or HF *sets*) is defined as

$$V_\omega \quad := \quad \bigcup_{n \in \mathbb{N}} V_n$$

where $V_0 := \emptyset$ and $V_{n+1} := \mathcal{P}(V_n)$.

3.2.1. The Logic $\mathsf{FSO}(\mathfrak{D})$. The language of $\mathsf{FSO}(\mathfrak{D})$ has the following three sorts.

- The same sort of individuals as $\mathsf{MSO}(\mathfrak{D})$ (§3.1.1).
- A sort of *hereditarily finite sets* with HF variables k, ℓ, etc., and with one constant $\dot{\kappa}$ for each hereditarily finite set κ.
- A sort of "*functions*" with variables F, G, H, etc.

In addition, $\mathsf{FSO}(\mathfrak{D})$ has the following terms for HF-sets, called the HF-*terms*:

$$K, L \quad ::= \quad k \quad | \quad \dot{\kappa} \quad | \quad F(x)$$

The formulae of $\mathsf{FSO}(\mathfrak{D})$ are given by the grammar

$$\begin{array}{rcl} \varphi, \psi & ::= & \top \quad | \quad \bot \quad | \quad x \doteq y \quad | \quad x \mathrel{\dot{<}} y \quad | \quad \mathsf{R}(x) \quad | \quad \mathsf{S}_d(x, y) \qquad\qquad (\text{for } d \in \mathfrak{D}) \\ & | & K \doteq L \quad | \quad K \mathrel{\dot{\in}} L \\ & | & \varphi \wedge \psi \quad | \quad \varphi \vee \psi \quad | \quad \psi \to \varphi \\ & | & (\exists x)\varphi \quad | \quad (\forall x)\varphi \quad | \quad (\exists \ell \mathrel{\dot{\in}} K)\varphi \quad | \quad (\forall \ell \mathrel{\dot{\in}} K)\varphi \quad | \quad (\exists F : K)\varphi \quad | \quad (\forall F : K)\varphi \end{array}$$

The above language for $\mathsf{FSO}(\mathfrak{D})$ is actually not the official language of FSO defined in [DR19], but it is interpretable in it.

The language of $\mathsf{FSO}(\mathfrak{D})$ has a standard interpretation, in which HF variables range over hereditarily finite sets and function variables range over suitable labeled trees. The key is of course that quantifications over HF sets and over functions are bounded. In a closed formula $(\exists \ell \mathrel{\dot{\in}} K)\varphi$ or $(\forall \ell \mathrel{\dot{\in}} K)\varphi$, the term K is interpreted as an HF set, say κ, and the HF variable ℓ ranges over the elements of κ. Similarly, in a closed formula $(\exists F : K)\varphi$ or $(\forall F : K)\varphi$, the term K is interpreted as an HF set κ, and the function variable F ranges over κ-labeled $\mathfrak{D}$-ary trees (*i.e.* over $\mathfrak{D}^* \to \kappa$). Under this interpretation, closed $\mathsf{FSO}(\mathfrak{D})$ formulae are interpretable as closed $\mathsf{MSO}(\mathfrak{D})$ formulae, essentially by expanding quantifications over HS sets using propositional connectives and by representing quantifications over functions by suitable quantifications over tuples of monadic variables. This translation is noted $\langle - \rangle$.

The philosophy underlying the axioms of $\mathsf{FSO}(\mathfrak{D})$ is to incorporate as a much as possible of the set theory of hereditarily finite sets.[1] We refer to [DR19] for details on this point. In addition, $\mathsf{FSO}(\mathfrak{D})$ has the following adaptation of the axioms of $\mathsf{MSO}(\mathfrak{D})$ (in the setting of §3.1.2):

[1] Recall that V_ω is a model of all the axioms of ZFC excepted the axiom of infinity.

- *Equality.* $\mathsf{FSO}(\mathfrak{D})$ has the same equality axiom on individuals as $\mathsf{MSO}(\mathfrak{D})$ (of course taken for the language of $\mathsf{FSO}(\mathfrak{D})$). In addition, $\mathsf{FSO}(\mathfrak{D})$ has the following equality axioms on HF sets:
$$K \doteq K \qquad \text{and} \qquad (K \doteq L \quad \longrightarrow \quad \varphi[K/k] \quad \longrightarrow \quad \varphi[L/k])$$
- $\mathsf{FSO}(\mathfrak{D})$ has the Tree Axioms of Fig. 3.1.
- *Induction.* Instead of a single induction axiom, $\mathsf{FSO}(\mathfrak{D})$ has an induction scheme (for each formula of its language).
- *Functional Choice Axioms.* Instead of the comprehension scheme, $\mathsf{FSO}(\mathfrak{D})$ has a series of *functional choice axioms*. We refer to [DR19, §3.4.5] for details, and just give as an example the following axiom scheme of HF-*bounded choice for function variables*:
$$(\forall x)(\exists \ell \mathrel{\dot\in} K)\varphi(x,\ell) \quad \longrightarrow \quad (\exists F : K)(\forall x)\varphi(x, F(x))$$

Remark 3.2.1. *We insist that functional choice axioms as above actually amount to comprehension in* $\mathsf{MSO}(\mathfrak{D})$*. Such axioms do not create choice predicates for individuals, which are known to be undefinable in* MSO*, and moreover to break decidability when added to the language of* MSO *(see [GS83, BG00, CL07]).*

The above mentioned translation $\langle - \rangle$ of $\mathsf{FSO}(\mathfrak{D})$ formulae to $\mathsf{MSO}(\mathfrak{D})$ formulae extends to provability. But because quantifications over HF sets are required to be bounded, $\mathsf{FSO}(\mathfrak{D})$ has axioms which necessarily involve formulae with free HF variables (e.g. the equality axiom $K \doteq K$). For a formula $\varphi(k_1, \dots, k_n)$ with free variables as displayed, the soundness of the translation $\langle - \rangle$ is stated as follows: if $\mathsf{FSO}(\mathfrak{D})$ proves $\varphi(k_1, \dots, k_n)$, then $\mathsf{MSO}(\mathfrak{D})$ proves $\langle \varphi(\dot\kappa_1, \dots, \dot\kappa_n) \rangle$ for each HF sets $\kappa_1, \dots, \kappa_n \in V_\omega$. Of course, one also has to handle free function variables: for a formula $\varphi(F_1, \dots, F_n)$ with free variables as displayed, if $\mathsf{FSO}(\mathfrak{D})$ proves $\varphi(F_1, \dots, F_n)$, then $\mathsf{MSO}(\mathfrak{D})$ proves $\langle (\forall F_1 : \dot\kappa_1) \dots (\forall F_n : \dot\kappa_n)\varphi(F_1, \dots, F_n) \rangle$ for each HF sets $\kappa_1, \dots, \kappa_n \in V_\omega$.

3.2.2. Games and Automata in $\mathsf{FSO}(\mathfrak{D})$. Most of [DR19] consists in developing a basic setting of games and automata in $\mathsf{FSO}(\mathfrak{D})$. The main constraint imposed by the language of $\mathsf{FSO}(\mathfrak{D})$ (and eventually by that of MSO) is that strategies should be representable as labeled $\mathfrak{D}$-ary trees. In view of Ex. 2.2.4, this means that one can only manipulate *positional* strategies (in the sense of §2.2). By way of illustration, we give the corresponding *external* notion of game in the setting of §2.2.

Definition 3.2.2. *An* MSO-Game *is a graph game G of the form*
$$G \quad = \quad \big((\mathfrak{D}^* \times V{\restriction}\mathsf{P}),\, (\mathfrak{D}^* \times V{\restriction}\mathsf{O}),\, E,\, (\varepsilon, v^\imath),\, \mathcal{W}\big)$$
where $V{\restriction}\mathsf{P}$ and $V{\restriction}\mathsf{O}$ are HF *sets with $v^\imath \in V{\restriction}\mathsf{P}$, and where the edge relation E is induced from labeled trees*
$$\begin{array}{rrcccccc} & E_{\mathsf{P}} & : & \mathfrak{D}^* & \longrightarrow & \big(V{\restriction}\mathsf{P} & \longrightarrow & \mathcal{P}(V{\restriction}\mathsf{O})\big) \\ \text{and} & E_{\mathsf{O}} & : & \mathfrak{D}^* & \longrightarrow & \big(V{\restriction}\mathsf{O} & \longrightarrow & \mathcal{P}(V{\restriction}\mathsf{P} \times \mathfrak{D})\big) \end{array}$$
as $(p, u) \longrightarrow_E (p', v)$ iff
$$\begin{array}{rl} \textit{either} & p' = p \text{ and } v \in E_{\mathsf{P}}(p)(u) \\ \textit{or} & p' = p.d \text{ with } (v, d) \in E_{\mathsf{O}}(p)(u) \end{array}$$

Note that a positional P-strategy in an MSO-game G as above is given by a labeled $\mathfrak{D}$-ary tree

$$\sigma_{\mathsf{P}} \quad : \quad \mathfrak{D}^* \quad \longrightarrow \quad (V{\restriction}\mathsf{P} \quad \longrightarrow \quad V{\restriction}\mathsf{O})$$

while a positional O-strategy is given by a labeled $\mathfrak{D}$-ary tree

$$\sigma_{\mathsf{O}} \quad : \quad \mathfrak{D}^* \quad \longrightarrow \quad \big(V{\restriction}\mathsf{O} \quad \longrightarrow \quad (V{\restriction}\mathsf{P} \times \mathfrak{D})\big)$$

MSO-games are simply a generalization of Ex. 2.2.4 specifically tailored to fit in the language of $\mathsf{FSO}(\mathfrak{D})$. In this setting, the difficult operations on automata are complementation and the *Simulation Theorem*.

For complementation, the construction of $\mathcal{A}^{\perp\!\!\!\perp}$ from $\mathcal{A}$, as well as the usual correspondence between O strategies in acceptance games for $\mathcal{A}$ and P strategies in acceptance games for $\mathcal{A}^{\perp\!\!\!\perp}$ are unproblematic. However, the general correctness statement of complementation, namely that $\mathcal{A}^{\perp\!\!\!\perp}$ accepts exactly the trees rejected by $\mathcal{A}$, relies on the determinacy of acceptance games, which in the language of $\mathsf{FSO}(\mathfrak{D})$ boils down the the positional determinacy of parity MSO-games. We were unfortunately not able to prove this statement from the axioms of $\mathsf{FSO}(\mathfrak{D})$. We thus extended the axiomatization of $\mathsf{FSO}(\mathfrak{D})$ with an axiom scheme (PosDet) stating the positional determinacy of each parity MSO-game.

For the Simulation Theorem we formalized the construction of [Wal02] (which contrary to e.g. [MS95] can be restricted to strategies represented as $\mathfrak{D}$-ary trees). The key point is that the formulation of the parity acceptance condition of the automaton $!\mathcal{A}$ relies on McNaughton Determinization Theorem [McN66] (Thm. 2.1.2). Each instance of this result can actually be formulated in MSO over ω-words, and thanks to Siefkes' Completeness Theorem [Sie70] (Thm. 3.1.4), is provable in MSO^ω. In order to import the relevant constructions into $\mathsf{FSO}(\mathfrak{D})$, we then use the fact that for a fixed MSO-game G, the MSO^ω formulae over the (rooted) infinite paths of G are interpretable in $\mathsf{FSO}(\mathfrak{D})$, and moreover that via this interpretation, the MSO^ω formulae which are true on the (rooted) infinite paths of G are provable in $\mathsf{FSO}(\mathfrak{D})$.

Having formalized usual constructions on automata makes it possible to prove the correctness of a translation of formulae to automata. The completeness of $\mathsf{FSO}(\mathfrak{D}) + (\mathsf{PosDet})$ is then obtained essentially by formalizing in $\mathsf{FSO}(\mathfrak{D})$ the usual proof of Rabin's Basis Theorem [Rab72] (see e.g. [Tho97, Thm. 6.18]). For each closed formula φ of $\mathsf{FSO}(\mathfrak{D})$ we have an automaton $\mathcal{A}(\varphi) : \mathbf{1}$ such that

$$\vdash_{\mathsf{FSO}(\mathfrak{D})+(\mathsf{PosDet})} \quad \varphi \quad \longleftrightarrow \quad (\exists F : \mathbf{1})(F \in \mathcal{L}(\mathcal{A}(\varphi)))$$

Here, $F \in \mathcal{L}(\mathcal{A}(\varphi))$ is an $\mathsf{FSO}(\mathfrak{D})$ formula stating the existence of a winning (positional) P strategy in the acceptance game of $\mathcal{A}(\varphi)$ over the (unique) labeled tree $F : \mathfrak{D}^* \to \mathbf{1}$. Similarly as in the usual (external) setting (see e.g. [Tho97, Ex. 6.12]), this acceptance game can be proved in $\mathsf{FSO}(\mathfrak{D})$ to be equivalent to a game on a finite graph. For such games, the Büchi-Landweber Theorem [BL69] (Thm. 2.4.1) says that the winner always has a *finite state* (positional) strategy. But finite state strategies on finite graphs can be represented by HF sets, so that infinite plays (*i.e.* ω-words) are actually the only infinite objects we need when speaking about winning in this case. As a consequence, the result of each instance of Büchi-Landweber Theorem can be formulated in MSO^ω and thus (via Siefkes' Theorem) be lifted to a provable $\mathsf{FSO}(\mathfrak{D})$ statement. It follows that $\mathsf{FSO}(\mathfrak{D})$ proves either

$$(\exists F : \mathbf{1})(F \in \mathcal{L}(\mathcal{A}(\varphi))) \qquad \text{or} \qquad \neg(\exists F : \mathbf{1})(F \in \mathcal{L}(\mathcal{A}(\varphi)))$$

from which we obtain the completeness of $\mathsf{FSO}(\mathfrak{D}) + (\mathsf{PosDet})$:

Theorem 3.2.3. *For each closed* $\mathsf{FSO}(\mathfrak{D})$ *formula* φ*,* $\mathsf{FSO}(\mathfrak{D}) + (\mathsf{PosDet})$ *proves either* φ *or* $\neg\varphi$*.*

Writing $\langle\mathsf{PosDet}\rangle$ for the axiom consisting of the $\langle-\rangle$ translation of each closed instance of (PosDet), we obtain the corresponding result for $\mathsf{MSO}(\mathfrak{D})$.

Corollary 3.2.4. *For each closed* $\mathsf{MSO}(\mathfrak{D})$ *formula* φ*,* $\mathsf{MSO}(\mathfrak{D}) + \langle\mathsf{PosDet}\rangle$ *proves either* φ *or* $\neg\varphi$*.*

Even if obtained completeness by formally extending $\mathsf{FSO}(\mathfrak{D})$ with the axiom (PosDet), we nevertheless can say that the formal development of a theory of games and automata in $\mathsf{FSO}(\mathfrak{D})$ was already quite demanding.[2] On the positive side, we can nevertheless say that Cor. 3.2.4 provides a theoretical algorithm for Rabin's Tree Theorem [Rab69] (Thm. 1.2.1) and thus to some extent "has" to be non trivial. On the other hand, as stated in the conclusion of [DR19], our main motivation for this axiomatization was (and still is) the Curry-Howard approach to MSO outlined in Chap. 1 and presented in Part. II of this document.

3.3. Is $\mathsf{MSO}(\mathfrak{D})$ Complete?

We were not able to prove the positional determinacy of (parity) MSO-games in $\mathsf{FSO}(\mathfrak{D})$. While we strongly suspect $\mathsf{FSO}(\mathfrak{D})$ (and thus $\mathsf{MSO}(\mathfrak{D})$) to be incomplete, neither were we able to prove the incompleteness of $\mathsf{MSO}(\mathfrak{D})$. We nevertheless find it interesting to report here an inconclusive attempt.

The idea would have been to give a model-theoretic counterpart to the proof of [CL07, Thm. 6] (see also [CLNW10, Thm. 3.7]), which, based on a pumping argument showing the undefinability of choice over individuals in $\mathsf{MSO}(\mathfrak{D})$, presents a game with decidable MSO-theory but with no definable winning strategy. The proof of [CL07, Thm. 6] is based on a non-regular tree P which lines up along its right branch a specific family of regular trees $(V_n)_n$. This family is designed so that for each automaton $\mathcal{A} : \mathbf{2} \times \mathbf{2}$, the language V_n for n large enough is counterexample for $\mathcal{A}$ implementing a choice predicate. The proof of [CL07, Thm. 6] consists in assuming a definable winning strategy σ in a game on P from which one can build a choice predicate φ for each of the V_n's (via their representation in P), and then to "diagonalize" w.r.t. the automaton $\mathcal{A}(\varphi)$.

Then our plan would have been to generate an Henkin model $\mathcal{M}(P)$ from the parameter P, so that assuming the completeness of $\mathsf{MSO}(\mathfrak{D})$ implies a definability property which leads to a contradiction similarly as in the proof of [CL07, Thm. 6]. While the assumption we make for the definition of $\mathcal{M}(P)$ may seem unreasonable, we find the adaptation of the argument of [CL07, Thm. 6] worth the trouble of this Section, because it may give ideas and reasoning principles for our context.

We assume in this section that $\mathfrak{D} = \mathbf{2} = \{0, 1\}$.

3.3.1. On the Undefinability of Choice in MSO. It is known that choice over individuals is not definable in MSO, in the sense that there is no MSO formula $\varphi(x, X)$ which selects a unique element $x \in X$ provided X is non-empty, *i.e.* such that

$$\mathfrak{T} \models \quad (\forall X)\Big((\exists x)(x \mathrel{\dot\in} X) \quad \longrightarrow \quad (\exists! x)\big(x \mathrel{\dot\in} X \ \wedge \ \varphi(x, X)\big)\Big)$$

This result, originally proved in [GS83], has numerous "useful" consequences e.g. the undefinability of winning strategies in some games, or the undefinability of any well-order on tree positions.

[2] The courageous reader is warmly encouraged to look at [DR19].

Such results, among others, can be found in [CL07], which also provides a direct pumping argument for the undefinability of choice.

The basis of the pumping argument of [CL07] (see also [CLNW10]) is the following family of regular languages, where $N, M \in \mathbb{N}$:

$$U_{M,N} \quad := \quad (0+1)^*(0^N 0^* 1)^M (0+1)^*$$

Undefinability of choice is a direct consequence of the following.

Theorem 3.3.1 ([CL07, Lem. 2 & Thm. 3])**.** *Fix a non-deterministic automaton $\mathcal{A} : \mathbf{2} \times \mathbf{2}$ and let $M \geq 2^{|Q_\mathcal{A}|} + 1$ and $N \geq |Q_\mathcal{A}| + 1$, where $|Q_\mathcal{A}|$ is the number of states of $\mathcal{A}$. If $\mathcal{A}$ accepts $\langle U_{M,N}, S\rangle$ for some singleton set $S \subseteq U_{M,N}$, then $\mathcal{A}$ also accepts $\langle U_{M,N}, S'\rangle$ for some singleton $S' \subseteq U_{M,N}$ different from S.*

3.3.2. An Inconclusive Result. Similarly as in the proof of [CL07, Thm. 6], consider the set of tree positions

$$P \quad := \quad \{1^n \cdot 0 \cdot p \mid n \in \mathbb{N} \text{ and } p \in U_{n,n}\}$$

So P contains each set $U_{n,n}$, but located at "address" $1^n \cdot 0$.

We can prove the following.

Proposition 3.3.2. *Let $P \subseteq \mathbf{2}^*$ be defined as above. Assume that there exists a collection of sets $\mathcal{M}(P)^o \in \mathcal{P}(\mathcal{P}(\mathbf{2}^*))$ such that for all $A \in \mathcal{P}(\mathbf{2}^*)$ we have $A \in \mathcal{M}(P)^o$ if and only if there is an $\mathsf{MSO}(\mathbf{2})$ formula $\varphi(X, x)$ (with free variables as displayed) such that*

$$A \quad = \quad \{p \in \mathbf{2}^* \mid \mathcal{M}(P) \models \varphi(P, p)\}$$

where $\mathcal{M}(P)$ is the Henkin model

$$\mathcal{M}(P) \quad := \quad (\mathbf{2}^*, \mathcal{M}(P)^o, \mathsf{R}, (\mathsf{S}_d)_{d\in\mathbf{2}}, <)$$

Then $\mathsf{MSO}(\mathbf{2})$ (and thus $\mathsf{FSO}(\mathbf{2})$) is incomplete.

The existence of model $\mathcal{M}(P)$ as required by Prop. 3.3.2 may seem quite dubious. Note that it would have been trivial to define $\mathcal{M}(P)^o$ if only first-order definability had been required, *i.e.* if the formulae $\varphi(x, X)$ would contain no quantification over monadic predicates. We further discuss this point in §3.3.3.

The rest of this §3.3.2 is devoted to a proof of Prop. 3.3.2. The argument is an adaptation to our context of constructions and ideas of [CL07, Thm. 6] (see also [CLNW10, Thm. 3.7]).

Assume given an Henkin model $\mathcal{M}(P)$ as indicated in Prop. 3.3.2. Note that $\mathcal{M}(P)$ is a model of $\mathsf{MSO}(\mathbf{2})$. Note also that $\mathcal{M}(P)$ and $\mathfrak{T}$ have the same individuals, so that they satisfy the same first order sentences. Since the individuals of $\mathcal{M}(P)$ and $\mathfrak{T}$ are definable by first order formulae, it moreover follows that $\mathcal{M}(P)$ and $\mathfrak{T}$ satisfy the same closed formulae with individual parameters.

Our strategy is to derive a contradiction from Thm. 3.3.1 under the assumption that $\mathsf{FSO}(\mathbf{2})$ (and thus $\mathsf{MSO}(\mathbf{2})$) is complete. In the following, relying on the provable translations between MSO and FSO (see [DR19, §3.6]), we shall make no syntactic difference between $\mathsf{MSO}(\mathbf{2})$ and $\mathsf{FSO}(\mathbf{2})$ formulae.

We now assume toward a contradiction that $\mathsf{FSO}(\mathbf{2})$ (and thus $\mathsf{MSO}(\mathbf{2})$) is complete.

Let $B := 1^*0$. Note that B is definable (both in $\mathcal{M}(P)$ and $\mathfrak{T}$) by a first-order formula (since $p \in B$ iff every strict prefix of p is either the root or a 0-successor and if moreover p itself is a 1-successor). Consider the following deterministic tree automaton $\mathcal{B} : \mathbf{2} \times \mathbf{2}$, with state set $\{q^\imath, q_{\mathsf{P}}, q_{\mathtt{tt}}\}$. When evaluated on a pair $\langle S, T\rangle$, $\mathcal{B}$ begins by staying in state $q^\imath$ as long as O chooses the tree direction 1, and goes to state q_{P} whenever O chooses direction 0. Then $\mathcal{B}$ follows the directions given by $T : \mathfrak{D}^* \to \mathbf{2}$ and goes (and stays forever) in state $q_{\mathtt{tt}}$ whenever it reaches a tree position which belongs to S. The acceptance condition of $\mathcal{B}$ consists of those sequences of states which contain $q_{\mathtt{tt}}$ whenever they contain q_{P}. Of course, if every element of B has an extension in S, then there is some T such that $\mathcal{B}$ accepts $\langle S, T\rangle$. In symbols:

$$\mathfrak{T} \models \quad \underbrace{(\forall X)\Big((\forall x \mathrel{\dot\in} B)(\exists y \mathrel{\dot\in} X)(x \mathrel{\dot\leq} y) \quad\longrightarrow\quad (\exists Y)\big(\langle X, Y\rangle \in \mathcal{L}(\mathcal{B})\big)\Big)}_{\varphi_{\mathcal{B}}}$$

It is tedious but easy to write down a formula $\varphi_{\mathrm{ev}}(X, Y, y)$ which holds in the standard model if and only if, provided y has a (unique) prefix x in B, y is (whenever it exists) the first element of X reached by evaluating $\mathcal{B}$ on $\langle X, Y\rangle$ while letting O play x. Then the following closed formulae hold in the standard model:

$$\begin{aligned} \varphi_! &:= (\forall X)(\forall Y)\Big((\forall x \mathrel{\dot\in} B)(\exists z \mathrel{\dot\in} X)(x \mathrel{\dot\leq} z) \quad\longrightarrow\quad \langle X, Y\rangle \in \mathcal{L}(\mathcal{B}) \quad\longrightarrow\quad (\exists! y)\varphi_{\mathrm{ev}}(X, Y, y)\Big) \\ \varphi_{\mathsf{cor}} &:= (\forall X)(\forall Y)(\forall y)\big(\varphi_{\mathrm{ev}}(X, Y, y) \quad\longrightarrow\quad y \mathrel{\dot\in} X\big) \end{aligned}$$

We now switch to the model $\mathcal{M}(P)$. By completeness the formulae $\varphi_{\mathcal{B}}$, $\varphi_!$ and φ_{cor} hold in $\mathcal{M}(P)$. In particular

$$\mathcal{M}(P) \models (\exists Y)\big(\langle P, Y\rangle \in \mathcal{L}(\mathcal{B})\big)$$

Let $T \in \mathcal{M}(P)^o$ such that $\mathcal{M}(P) \models \langle P, T\rangle \in \mathcal{L}(\mathcal{B})$. Note that we have

$$\mathcal{M}(P) \models \quad (\exists! y)\varphi_{\mathrm{ev}}(P, T, y)$$

Moreover, since $T \in \mathcal{M}(P)^o$, it follows from our assumption on $\mathcal{M}(P)$ that T is definable in $\mathcal{M}(P)$ by a formula, say $\psi(Y, x)$:

$$\mathcal{M}(P) \models \quad (\forall x)\big(x \mathrel{\dot\in} T \quad\longleftrightarrow\quad \psi(P, x)\big)$$

Consider finally the formula

$$\varphi(X, y) \quad := \quad (\exists Y)\Big((\forall x)\big(x \mathrel{\dot\in} Y \;\leftrightarrow\; \psi(X, x)\big) \quad\wedge\quad \varphi_{\mathrm{ev}}(X, Y, y)\Big)$$

We have

$$\mathcal{M}(P) \models \quad (\exists! y)\varphi(P, y) \quad\wedge\quad (\forall y)\big(\varphi(P, y) \;\longrightarrow\; y \mathrel{\dot\in} P\big)$$

Furthermore we can replace the individual argument y of φ by a singleton set Y, to the effect that $\varphi(X, y)$ is equivalent to a formula $\tilde\varphi(X, Y)$ in the sense that

$$(\forall X)(\forall Y)(\forall y)\Big(\mathsf{Sing}(Y) \quad\longrightarrow\quad y \mathrel{\dot\in} Y \quad\longrightarrow\quad \big(\varphi(X, y) \;\longleftrightarrow\; \tilde\varphi(X, Y)\big)\Big)$$

(where $\mathsf{Sing}(-)$ is a first-order definition of "being a singleton") holds both in $\mathcal{M}(P)$ and $\mathfrak{T}$.

We momentarily switch back to the standard model $\mathfrak{T}$. Using Rabin's Theorem [Rab69], let $\mathcal{A} : \mathbf{2} \times \mathbf{2}$ be a total non-deterministic (parity) automaton equivalent to $\tilde\varphi(X, Y)$, *i.e.* such that

$$\mathfrak{T} \models \quad (\forall X)(\forall Y)\big(\tilde\varphi(X, Y) \quad\longleftrightarrow\quad \langle X, Y\rangle \in \mathcal{L}(\mathcal{A})\big)$$

By our completeness assumption, the automaton $\mathcal{A}$ is also correct for $\tilde{\varphi}$ in the model $\mathcal{M}(P)$, in the sense that

$$\mathcal{M}(P) \models \quad (\forall X)(\forall Y)\big(\tilde{\varphi}(X,Y) \quad \longleftrightarrow \quad \langle X, Y\rangle \in \mathcal{L}(\mathcal{A})\big)$$

We thus get

$$\mathcal{M}(P) \models \quad (\forall Y)(\forall y)\Big(\mathsf{Sing}(Y) \quad \longrightarrow \quad y \mathrel{\dot\in} Y \quad \longrightarrow \quad \big(\varphi(P,y) \ \longleftrightarrow \ \langle P, Y\rangle \in \mathcal{L}(\mathcal{A})\big)\Big)$$

The rest of the argument goes as follows. Take $p := 1^n0$ with $n \geq M, N$ for M, N as required by Thm. 3.3.1 applied to $\mathcal{A}$. Since $p \in B$, there is a unique element of P "above p", say $p.r$, such that *in the sense of* $\mathcal{M}(P)$, the automaton $\mathcal{A}$ accepts the pair $\langle P, \{p.r\}\rangle$. We would like to transfer this to the standard model, in order to reach a contradiction from Thm. 3.3.1. But this requires to get rid of the undefinable parameter P.

The first step is to restrict P to $P^{p\dot\leq}$, where $X^{x\dot\leq}$ is the predicate defined from X as

$$(\forall y)\Big(y \mathrel{\dot\in} X^{x\dot\leq} \quad \longleftrightarrow \quad \big(y \mathrel{\dot\in} X \ \wedge \ x \mathrel{\dot\leq} y\big)\Big)$$

Fix a well order on the states of $\mathcal{A}$. Recalling that individuals (*i.e.* tree positions) are definable, the following is provable in FSO. Fix an arbitrary predicate X. Let q be the least state of $\mathcal{A}$ such that for some extension p' of p, there is a winning P-strategy in $G(\mathcal{A}, \langle X, \{p'\}\rangle)$ which reaches position (p, q). Then there is a P strategy in the acceptance game $G(\mathcal{A}, \langle X^{p\dot\leq}, \{p'\}\rangle)$ which is winning from position (p, q). Moreover, for each extension p'' of p, there is a winning P strategy in the acceptance game $G(\mathcal{A}, \langle X, \{p''\}\rangle)$ whenever there is a P strategy in the acceptance game $G(\mathcal{A}, \langle X^{p\dot\leq}, \{p''\}\rangle)$ which is winning from position (p, q). As a consequence, now in the sense of $\mathcal{M}(P)$, there is a state q of $\mathcal{A}$ such that there is a unique extension p' of p for which P has a winning strategy in $G(\mathcal{A}, \langle P^{p\dot\leq}, \{p'\}\rangle)$ from position (p, q).

It remains to get rid of $P^{p\dot\leq}$, and we are done if we show that $P^{p\dot\leq}$ is definable in $\mathcal{M}(P)$ by a formula $\delta(x)$ of one free individual variable x. But $P^{p\dot\leq} = 1^n0 \cdot U_{n,n}$ (with n as above), which is a regular subset of $\mathbf{2}^*$. It follows that *in the standard model* $\mathfrak{T}$, $P^{p\dot\leq}$ is definable by a formula $\delta(x)$:

$$\mathfrak{N} \models \quad (\forall x)\big(x \mathrel{\dot\in} P^{p\dot\leq} \quad \longleftrightarrow \quad \delta(x)\big)$$

It is easy to see that $P^{p\dot\leq}$ is also defined by $\delta(x)$ in $\mathcal{M}(P)$. Indeed, note that for each $s \in \mathbf{2}^*$, it follows from our completeness assumption that

$$\mathfrak{N} \models \delta(s) \quad \text{if and only if} \quad \mathcal{M}(P) \models \delta(s)$$

We thus obtain

$$\mathcal{M}(P) \models \quad (\forall x)\big(x \mathrel{\dot\in} P^{p\dot\leq} \quad \longleftrightarrow \quad \delta(x)\big)$$

Thanks to the formula $\delta(x)$, there is a *closed* formula β which expresses, both in $\mathcal{M}(P)$ and $\mathfrak{T}$, that there is a unique extension p' of p such that there is a P strategy in $G(\mathcal{A}, \langle P^{p\dot\leq}, \{p'\}\rangle)$ which is winning from position (p, q). By completeness, β holds in the standard model $\mathfrak{T}$. Now, writing extensions of p as $p.r$, this obviously implies that there is unique $r \in U_{n,n}$ such that there is a P strategy in $G(\mathcal{A}, \langle U_{n,n}, \{r\}\rangle)$ which is winning from position (ε, q). We thus have reached a contradiction from Thm. 3.3.1 since $n \geq M, N$ with M, N independent from the initial state of $\mathcal{A}$.

3.3.3. Discussion. What makes the existence of $\mathcal{M}(P)$ look problematic is the apparent circularity in the requirement that $\mathcal{M}(P)^o$ consists exactly of the sets definable by formulae with monadic quantifiers evaluated in $\mathcal{M}(P)^o$.

Consider for instance the usual definition of Gödel's *constructible sets* from set theory (see e.g. [Jec06]). Adapted to our context, it would amount to define a family $(L_\mathfrak{a})_\mathfrak{a}$ of subsets of $\mathcal{P}(\mathbf{2}^*)$ by induction on ordinals $\mathfrak{a}$ as follows:

- $L_\emptyset := \{P\}$;
- $L_\mathfrak{a} := \bigcup_{\mathfrak{b}<\mathfrak{a}} L_\mathfrak{b}$ for a limit ordinal $\mathfrak{a}$;
- $L_{\mathfrak{a}+1}$ is the set of all $A \in \mathcal{P}(\mathbf{2}^*)$ such that for some $\mathsf{MSO}(\mathbf{2})$ formula $\varphi(X_1,\dots,X_n,x)$ (with free variables as displayed) and some $A_1,\dots,A_n \in L_\mathfrak{a}$, we have

$$A \quad = \quad \{p \in \mathbf{2}^* \mid \mathfrak{N}{\restriction}L_\mathfrak{a} \models \varphi(A_1,\dots,A_n,p)\}$$

Here $\mathfrak{T}{\restriction}\mathcal{S}^o$ is the Henkin model with standard individuals and with monadic predicates $\mathcal{S}^o \subseteq \mathcal{P}(\mathbf{2}^*)$. As usual, $L_\mathfrak{a} \subseteq L_\mathfrak{b}$ whenever $\mathfrak{a} < \mathfrak{b}$. Also, we have $L_{\mathfrak{a}_\ell+1} = L_{\mathfrak{a}_\ell}$ for some limit ordinal $\mathfrak{a}_\ell$, so that $L_\mathfrak{a} = L_{\mathfrak{a}_\ell}$ for each $\mathfrak{a} > \mathfrak{a}_\ell$. Let $\mathcal{L}(P) := \mathfrak{N}{\restriction}L_{\mathfrak{a}_\ell}$ (the choice of $\mathfrak{a}_\ell$ is irrelevant), so that $\mathcal{L}(P)^o = L_{\mathfrak{a}_\ell}$. The fact that $\mathcal{L}(P)^o$ is a model of $\mathsf{MSO}(\mathbf{2})$ is trivial for the individual part, and follows from the fact that $L_{\mathfrak{a}_\ell+1} = L_{\mathfrak{a}_\ell}$ for the comprehension scheme. However, $\mathcal{L}(P)$ seems to have no reason to satisfy the requirement of Prop. 3.3.2. The point is that each $A \in \mathcal{L}(P)^o$ is definable, of course from parameters in $\mathcal{L}(P)^o$, but relative to some $L_\mathfrak{a}$ with possibly $\mathfrak{a} < \mathfrak{a}_\ell$, while there seems to be no reason for the theories of $\mathfrak{T}{\restriction}L_\mathfrak{a}$ and $\mathfrak{T}{\restriction}L_{\mathfrak{a}_\ell}$ to coincide.

On the other hand, in the setting of (subsystems of) second-order arithmetic, for each set $P \subseteq \mathbb{N}$, there exists a minimal β-model of comprehension containing P ([Sim10, Thm. VII.5.17]).[3] This follows from the fact that the counterpart of "$X \in \mathcal{L}^o(P)$" is definable in the language of arithmetic (using transfinite recursion), so that thanks to *absoluteness*, all β-models of comprehension coincide on their constructible part. All this ultimately relies on the availability of Gödel's codings in the language of arithmetic, while in our setting nothing seems to ensure that all "β-models" of $\mathsf{MSO}(\mathbf{2})$-comprehension contain $\mathcal{L}(P)$.[4]

An other possibility, that we did not investigate yet, is to use the fact that the set P of §3.3.2 belongs to the fourth level of the higher-order pushdown hierarchy (see [CL07, CLNW10] and references therein). One might then consider Henkin models of $\mathsf{MSO}(\mathbf{2})$ whose predicates belong to some (or all) levels of the higher-order pushdown hierarchy. Of course, the argument of Prop. 3.3.2 is not likely to extend, since the set T would now be defined from P and other pushdown parameters as well. On the other hand, the argument may lift to some extension of $\mathsf{MSO}(\mathbf{2})$ with atomic predicates from the pushdown hierarchy.

3.4. An Axiom of Definition by Cumulative Unions

As requiring (PosDet) for the completeness of $\mathsf{FSO}(\mathfrak{D})$ is not very satisfactory, together with Anupam Das we investigated other axioms which may more closely target the aspects of the language of $\mathsf{MSO}(\mathfrak{D})$ which made us stuck when trying to prove positional determinacy. While this is still work in progress, we give an informal account of it here.

[3] A β model is a model of arithmetic with standard individuals, and which validates all true Σ^1_1 sentences (with parameters) of second-order arithmetic.

[4] At least if these "β-models" are defined relative to the language of $\mathsf{MSO}(\mathbf{2})$.

Let us first briefly discuss the general case of positional determinacy of parity games in the setting of Thm. 2.2.3. The original proof of [EJ91], as well as that of [Kla94], relies on ordinal notations. While allowing for constructive arguments (see e.g. [Zie98]) ordinal notations for uncountable ordinals seem unavailable within the language of $\mathsf{MSO}(\mathfrak{D})$. Following [Zie98, Tho97] one can avoid ordinals via the following *Uniformization Lemma*. We state it in the setting of §2.2.

Lemma 3.4.1 (Uniformization)**.** *Fix a graph game G such that $\overline{v}.\chi \in \mathcal{W}$ whenever $\chi \in \mathcal{W}$. For $\mathsf{J} \in \{\mathsf{P}, \mathsf{O}\}$, let W_{J} be the set of positions $v \in V$ such that Player J has a winning positional strategy in G from v. Then there is a positional strategy σ for Player J such that σ is winning from every $v \in W_{\mathsf{J}}$.*

In symbols, Lem. 3.4.1 can be presented as the following inversion of quantifiers:

$$\big(\exists\sigma \text{ positional } \mathsf{J}\text{-strategy}\big)(\forall v)\Big[\big(\exists\tau \text{ positional } \mathsf{J}\text{-strategy win. } G_v\big) \implies \sigma \text{ wins } G_v\Big]$$

Uniformization easily follows if one has access a well order on positional strategies (see e.g. [Zie98, Tho97]). For the sake of completeness, we recall a well-known argument.

Proof sketch of Lem. 3.4.1. Fix a well order on the set of all positional J-strategies which are winning in G_v for v ranging over W_{J}. Define a positional J-strategy σ by following, for each $v \in W_{\mathsf{J}}$, the move taken by the least positional winning J-strategy from v. Then, in an infinite play χ of σ from some $v \in W_{\mathsf{J}}$, after a while the strategy followed by χ stabilizes. Hence a suffix of χ is a winning play, which implies that χ is itself winning by our assumption on $\mathcal{W}$. □

It case the graph game G in Lem. 3.4.1 is finite, then there are only finitely many positional strategies on G, and well ordering them comes at no axiomatic cost. On the other hand, for infinite games (e.g. the acceptance games required for the complementing tree automata), a well order on positional strategies is obtained via the Axiom of Choice, applied to a set of the same cardinality as $\mathcal{P}(V)$.

It may be worth recalling here some known fact about determinacy in the setting of second-order arithmetic. First, recall that ω-regular sets belong to the finite levels of the difference hierarchy on $\mathbf{\Pi}^0_2$ sets, while $\mathsf{PA2}$ proves determinacy of all games of the finite levels of the difference hierarchy on $\mathbf{\Pi}^0_3$ sets [MS11].[5] Second, it has been shown in [KM16] that positional determinacy of parity games can be proved in arithmetic using Π^1_2 comprehension.

Actually, determinacy proofs usually rely on some form of choice, and the result of [MS11] does appeal to the $\mathbf{\Pi}^1_4$-conservativity of the axiom of dependent choices (formulated in the language of $\mathsf{PA2}$) over $\mathsf{PA2}$.[6] As this conservativity result relies on absoluteness and thus ultimately on coded constructible sets (see e.g. [Sim10]), it is likely that trying to prove a determinacy result for ω-regular games in the setting of $\mathsf{MSO}(\mathfrak{D})$ may lead to complications.

3.4.1. The Axiom (Def) of Definition by Cumulative Unions.

Together with Anupam Das, we are currently investigating an approach to Uniformization which amounts to a form of dependent choices expressible in the language of $\mathsf{MSO}(\mathfrak{D})$.

Working in the language of $\mathsf{FSO}(\mathfrak{D})$, fix HF-sets K and L. We assume notations (see [DR19]) allowing us to manipulate function variables $F : K^L$ as if they were uncurried to functions $\mathfrak{D}^* \times L \to K$. We similarly assume notations to quantify over $\mathfrak{D}^* \times L$ with variables u, v, etc

[5]See [Tan91] for the proof theoretic strength of Σ^0_2 determinacy and [MT07] for Δ^0_3 determinacy.
[6]This is phenomenon also occurs for Borel determinacy in set theory (see e.g. [Mar75, Mos09]).

$$
\begin{array}{rcl}
\mathrm{Lin}(X) & := & (\forall x \mathrel{\dot\in} X)(\forall y \mathrel{\dot\in} X)\big(x \mathrel{\dot\leq} y \quad\vee\quad y \mathrel{\dot\leq} x\big) \\
\mathsf{S}(x,y) & := & \bigvee_{d\in\mathfrak{D}} y \doteq \mathsf{S}_d(x) \\
\mathrm{Path}(X) & := & \mathrm{Lin}(X) \quad\wedge\quad (\forall x \mathrel{\dot\in} X)(\exists y \mathrel{\dot\in} X)\mathsf{S}(x,y) \\
\mathrm{Tree}(T) & := & (\forall x)(\forall y)\big(x \mathrel{\dot\in} T \quad\longrightarrow\quad y \mathrel{\dot\leq} x \quad\longrightarrow\quad y \mathrel{\dot\in} T\big) \\
\mathrm{FT}(T) & := & \mathrm{Tree}(T) \quad\wedge\quad (\forall X : \mathbf{2})\big(\mathrm{Path}(X) \quad\longrightarrow\quad (\exists x \mathrel{\dot\in} X)\neg(x \mathrel{\dot\in} T)\big) \\
\mathsf{S}_{\mathrm{FT}}(T,U) & := & \mathrm{FT}(T) \quad\wedge\quad \mathrm{FT}(U) \quad\wedge\quad T \mathrel{\dot\subseteq} U \quad\wedge\quad (\exists! y)\big(y \mathrel{\dot\in} U \;\wedge\; \neg(y \mathrel{\dot\in} T)\big)
\end{array}
$$

Figure 3.2.: Some defined formulae used in the axiom (Def).

(which of course are expanded as pairs of quantifications over individuals and L). With these notations, the axiom (Def), schematic in K, L, $\varphi(s,T)$ and $\mathrm{MC}(s,T,u)$, is the following:

$$
\begin{array}{l}
\left[\begin{array}{ll}
 & (\forall s, s' : K^L)(\forall^{\mathrm{FT}} T)\,\big(s =_{\mathrm{MC}(s,T)} s' \quad\longrightarrow\quad \varphi(s,T) \quad\longrightarrow\quad \varphi(s',T)\big) \\
\wedge & (\forall s : K^L)(\forall^{\mathrm{FT}} T, U)\,\big(\varphi(s,T) \quad\longrightarrow\quad U \mathrel{\dot\subseteq} T \quad\longrightarrow\quad \varphi(s,U)\big) \\
\wedge & (\exists s : K^L)(\forall^{\mathrm{FT}} T)\big((\forall x)\big(x \mathrel{\dot\in} T \;\leftrightarrow\; x \doteq \mathsf{R}\big) \quad\longrightarrow\quad \varphi(s,T)\big) \\
\wedge & (\forall s : K^L)(\forall^{\mathrm{FT}} T, T')\big(\varphi(s,T) \quad\longrightarrow\quad \mathsf{S}_{\mathrm{FT}}(T,T') \quad\longrightarrow \\
 & \qquad (\exists s' : K^L)\big(\varphi(s',T') \quad\wedge\quad \mathrm{MC}(s,T) \subseteq \mathrm{MC}(s',T') \quad\wedge\quad s =_{\mathrm{MC}(s,T)} s'\big)\big)
\end{array}\right] \\
\longrightarrow\quad (\exists s : K^L)(\forall^{\mathrm{FT}} T)\varphi(s,T)
\end{array}
$$

The relativized quantification $(\forall^{\mathrm{FT}} T)(-)$ stands for $(\forall T : \mathbf{2})(\mathrm{FT}(T) \to (-))$ and the formulae FT and S_{FT} are defined in Fig. 3.2. Note that in the standard model, $\mathrm{FT}(T)$ holds iff $T : \mathfrak{D}^* \to \mathbf{2}$ is the characteristic function of a finite subtree of $\mathfrak{D}^*$, and that for finite trees T, U, the formula $\mathsf{S}_{\mathrm{FT}}(T,U)$ holds iff U extends T with exactly one node. The axiom (Def) uses in addition the following defined formulae:

$$
\begin{array}{rcl}
s =_{\mathrm{MC}(t,T)} s' & := & (\forall u \in \mathfrak{D}^* \times L)\big(\mathrm{MC}(t,T,u) \quad\longrightarrow\quad s(u) \doteq s'(u)\big)\big) \\
\mathrm{MC}(s,T) \subseteq \mathrm{MC}(s',T') & := & (\forall u \in \mathfrak{D}^* \times L)\big(\mathrm{MC}(s,T,v) \quad\longrightarrow\quad \mathrm{MC}(s',T',v)\big)
\end{array}
$$

The intuition is that $\mathrm{MC}(s,T)$ stands for a subset of $\mathfrak{D}^* \times L$ (namely the set of all $v \in \mathfrak{D}^* \times L$ such that $\mathrm{MC}(s,T)$), and that $s =_{\mathrm{MC}(t,T)} s'$ means that s, s', seen as functions $\mathfrak{D}^* \times L \to K$, agree on $\mathrm{MS}(t,T)$.

We read the axiom (Def) as follows:

(a) The first premise means that $\varphi(-,-)$ is "continuous" in its first argument with "modulus of continuity" MC.

(b) The second premise means that $\varphi(-,-)$ is contravariant in its second argument.

(c) The third premise means that φ is satisfiable at the root.

(d) The fourth premise means that if φ holds at (s,T) and if T' is a one-step extension of T then φ holds at (s',T') for some $s' : \mathfrak{D}^* \times L \to K$ which agrees with s on (s,T).

In the following, we refer to the premises of (Def) as the above items (a)–(d)

It is easy (while a bit tedious) to check that (Def) holds in the standard model. We defer the proof to App. B.

Proposition 3.4.2. *All instances of* (Def) *hold in the standard model.*

Remark 3.4.3. *We believe* (Def) *to be provable in* PA2 *augmented with the axiom* AC_0^k *(see [Sim10, Def. VII.6.1]) for* φ *a* Σ_k^1 *formula and* MC *a* Δ_k^1 *formula.*

3.4.2. Uniformization in $(\mathsf{FSO}(\mathfrak{D}) + \mathsf{Def})$. It is a tedious ongoing work (in collaboration with Anupam Das) to show that $\mathsf{FSO}(\mathfrak{D}) + (\mathsf{Def})$ proves (PosDet). The proof is split in two parts:

(1) Show that $\mathsf{FSO}(\mathfrak{D})$ augmented with Uniformization for MSO-Games (see Def. 3.2.2) proves (PosDet) by formalizing the argument of [Tho97].

(2) Show Uniformization from suitable instances of (Def).

In order to illustrate the axiom (Def), we give an argument for (2) in the standard model . We see for now no reason for this argument not be formalizable in $\mathsf{FSO}(\mathfrak{D})$, but this tedious task has yet to be completed.

Consider an $\mathsf{MSO}(\mathfrak{D})$-game $G = (V{\restriction}\mathsf{P}, V{\restriction}\mathsf{O}, E, q^\imath, \mathcal{W})$ such that $\overline{v}.\chi \in \mathcal{W}$ whenever $\chi \in \mathcal{W}$. Write $V := (\mathfrak{D}^* \times V{\restriction}\mathsf{P}) + (\mathfrak{D}^* \times V{\restriction}\mathsf{O})$ and $V{\restriction}\mathsf{PO} := V{\restriction}\mathsf{P} + V{\restriction}\mathsf{O}$. In the following, we reason modulo

$$V \quad \simeq \quad \mathfrak{D}^* \times V{\restriction}\mathsf{PO}$$

For simplicity, we only consider the case of player P. So let W be the set of all $v \in V$ such that P has a (positional) winning strategy in

$$G_v \quad := \quad (V, E, v, \mathcal{W})$$

We are going to show that P has a (positional) strategy σ which is winning in each of the G_v for $v \in W$. To this end we will apply (Def) with the following MC and φ.

- For a finite tree $T \subseteq \mathfrak{D}^*$ and a function $\sigma : \mathfrak{D}^* \longrightarrow (V{\restriction}\mathsf{PO} \to V{\restriction}\mathsf{PO})$, we let $\mathrm{MC}(\sigma, T, v)$ hold on $v \in V$ if there is some $(p, k) \in W$ with $p \in T$ and such that v is reachable from (p, k) by σ seen as a partial strategy $\mathfrak{D}^* \longrightarrow (V{\restriction}\mathsf{P} \rightharpoonup V{\restriction}\mathsf{O})$.
- For a finite tree $T \subseteq \mathfrak{D}^*$ and a function $\sigma : \mathfrak{D}^* \longrightarrow (V{\restriction}\mathsf{PO} \to V{\restriction}\mathsf{PO})$, we let $\varphi(s, T)$ hold if and only if σ induce a *total* strategy $\mathfrak{D}^* \longrightarrow (V{\restriction}\mathsf{P} \to V{\restriction}\mathsf{O})$ which is winning in each $G_{(p,k)}$ with $(p, k) \in W$ and $p \in T$.

Remark 3.4.4. *Note that* in the language of second-order arithmetic*,* MC *is representable by an arithmetical formula while* φ *is representable by a* Π^1_1 *formula.*

In the following, we assume granted the following *Finite Merging* property:

- Fix $p \in \mathfrak{D}^*$ and an HF-set $L \subseteq V$. If for all $\ell \in L$, P has a winning positional strategy in $G_{(p,\ell)}$, then P has a positional strategy which is winning in each $G_{(p,\ell)}$ for $\ell \in L$.

We expect Finite Merging to be provable in $\mathsf{FSO}(\mathfrak{D})$. The reason is that $\mathsf{FSO}(\mathfrak{D})$ is actually equipped with axioms providing a well order for each HF-set (see [DR19, Rem. 3.17, §3.4.4]).

Note that if there is some P-strategy σ such that $\varphi(\sigma, T)$ holds for each finite tree $T \subseteq \mathfrak{D}^*$, we easily get that σ wins each of the G_v for $v \in W$. It thus remains to check the premises of (Def) with our MC and φ.

(a) Consider functions σ, σ' and a finite tree $T \subseteq \mathfrak{D}^*$. Assume that $\sigma =_{\mathrm{MC}(\sigma,T)} \sigma'$ and that $\varphi(\sigma, T)$ holds. Hence σ induces a (total) strategy $\mathfrak{D}^* \longrightarrow (V{\restriction}\mathsf{P} \to V{\restriction}\mathsf{O})$, and moreover we have $\sigma(v) = \sigma'(v)$ whenever there is some $(p, k) \in W$ with $p \in T$ such that v is reachable from (p, k) by σ. Note that this implies $\sigma(p, k) = \sigma'(p, k)$ for each $(p, k) \in W$ with $p \in T$, and (by induction and since σ is assumed to induce a total strategy) that $\mathrm{MC}(\sigma, T) = \mathrm{MC}(\sigma', T)$. But now, assuming $\varphi(\sigma, T)$, we immediately get $\varphi(\sigma', T)$, as a play of σ' from some $(p, k) \in W$ with $p \in T$ only consists of positions in $\mathrm{MC}(\sigma', T) = \mathrm{MC}(\sigma, T)$.

(b) Trivial.

(c) Trivial.

(d) Let σ and T such that $\varphi(\sigma, T)$ holds, and consider some tree T' of the form $T + \{p\}$ with $p \in \mathfrak{D}^*$. Let K_0 be the HF-set of all $k \in V{\restriction}\mathsf{PO}$ such that $(p, k) \in W$, and write $K_0 = K + L$, were K is the set of all $k \in K_0$ such that (the strategy induced by) σ wins $G_{(p,k)}$. Now, by *Finite Merging*, there is some strategy τ which is winning in $G_{(p,\ell)}$ for each $\ell \in L$. We define the strategy σ' as playing as σ on $\mathrm{MC}(\sigma, T)$ and as τ everywhere else. We have $\mathrm{MC}(\sigma, T) \subseteq \mathrm{MC}(\sigma', T')$ and $\sigma =_{\mathrm{MC}(\sigma,T)} \sigma'$ by construction of σ'. We moreover have $\varphi(\sigma', T')$ since, as σ is winning from each $(\tilde{p}, k) \in W$ with $\tilde{p} \in T$, no (p, ℓ) with $\ell \in L$ is reachable by σ from some $(\tilde{p}, k) \in W$ with $\tilde{p} \in T$, so that $\{p\} \times L$ is disjoint from $\mathrm{MC}(\sigma, T)$.

Part II.

Toward Curry-Howard Approaches

4. Introduction to Part II

This part surveys published and unpublished works developing the Curry-Howard approach to MSO outlined in Chap. 1. Much of this material (but for §8.1 and §8.3) was already written in [Rib15, Rib18, PR17, PR18b, PR19]. The linear variants LMSO of MSO are discussed in Chap. 8, while Chapters 5–7 concern the corresponding automata-based realizability model.

The automata-based realizability model comes in two variants. The first one, described in [Rib15] and surveyed in Chap. 5, is based on usual alternating automata. While this model has some categorical limitations (see §5.5), it is nevertheless sufficient to sketch the connection between linear logic and the interpretation of MSO in tree automata mentioned in §1.5. The second variant, which was developed in [Rib18] (and reused in [PR17, PR18b, PR19]), comes from a simplification of the underlying game model, namely to (total) *zig-zag* strategies. Chapter 6 gathers (mostly unpublished) simple but technical basic material on a setting of zig-zag games, while Chap. 7 presents its automata side, *uniform automata*, and how it inherits some categorical structure of zig-zag games, leading to the situation described in §1.5.

Both variants of the automata-based realizability model follow the guidelines and axiomatizations provided by categorical logic and categorical approaches to the Curry-Howard correspondence, for which we refer to [Jac01, LS86] and [AC98]. We moreover refer to [Mel09] for a comprehensive presentation of categorical axiomatizations of models of (subsystems of) linear logic. Technically, the realizability models are presented as monoidal categories *indexed* (or *fibred*) over a base category $\mathbf{T}$ of trees, whose objects are alphabets and whose morphisms from Σ to Γ induce functions from Σ-labeled trees to Γ-labeled trees (see §4.2 and §6.4.2).

The present chapter gives some background to Part II. First, in §4.1–4.3, we expose some ingredients and methodology of our approach based on categorical logic. We state in §4.1 the minimal requirements imposed by categorical semantics of proofs, and §4.2 presents some basic ideas and motivations on indexed categories for modeling free variables and quantifications. Finally, in §4.3 we explain why it seems difficult to obtain a suitable categorical semantics of implications using usual connectives on automata. Second, building on [Rib15], in §4.4 & §4.5 we sketch the connection between linear logic and the interpretation of MSO in tree automata mentioned in §1.5. Third, we present some material on game semantics in §4.6-4.7. While §4.6 aims at presenting what can make this technology a pertinent solution to the composition problem raised in §4.3, we expose in §4.7 some basic well-known facts on the representation of strategies as relations which underlie both realizability models.

It turns out to be conceptually convenient to think in terms of a deduction system *at the level of automata*. For a start, let us think that such a deduction system manipulates sequents of the form

$$T \ ; \ \mathcal{A}_1, \dots, \mathcal{A}_n \vdash \mathcal{B} \tag{4.1}$$

where T is an infinite tree labeled over (say) the alphabet Σ, and $\mathcal{A}_1, \dots, \mathcal{A}_n, \mathcal{B}$ are tree automata over Σ. Extending the terminology of §1.6, the *provability interpretation* of such a deduction system is that if the sequent (4.1) is provable, then the automaton $\mathcal{B}$ accepts the tree T as soon as the automata $\mathcal{A}_1, \dots, \mathcal{A}_n$ all accept T.

4.1. Compositionality and Categorical Semantics. The method of categorical semantics of proofs (see e.g. [LS86, AC98, Jac01, Mel09]) is to interpret *proofs* in a deduction system as *morphisms* in a category $\mathbb{C}$, such that $\mathbb{C}$ is equipped with some structure corresponding to the connectives and rules of the deduction system. For the moment, let us step back from acceptance games and consider run trees. Our task is thus to devise categories whose objects include all sets of the form $\mathcal{A}(T)$, for an automaton $\mathcal{A}$ and a tree T, and such that the proofs of a sequent $T\ ;\ \mathcal{A} \vdash \mathcal{B}$ can be interpreted as morphisms from $\mathcal{A}(T)$ to $\mathcal{B}(T)$.

The first requirement of categorical semantics is that the very notion of category already imposes interpretations to be *compositional*. Recall that the sets of morphisms of a (locally small) category $\mathbb{C}$ come with associative *composition* operations

$$(-)\circ(-) \quad : \quad \mathbb{C}[B,C]\times\mathbb{C}[A,B] \quad\longrightarrow\quad \mathbb{C}[A,C] \qquad \text{(for each } \mathbb{C}\text{-objects } A,B,C\text{)}$$

and with identity morphisms $\mathrm{id}_A \in \mathbb{C}[A,A]$ which are neutral for composition:

$$f\circ \mathrm{id}_A \quad=\quad f \quad=\quad \mathrm{id}_B\circ f \qquad \text{for every } f\in\mathbb{C}[A,B] \tag{4.2}$$

Composition and identities provide the interpretations respectively of the following instances of the usual *cut* and *axiom* rules:

$$(\textsc{Cut}_0)\ \frac{T\ ;\ \mathcal{A}\vdash\mathcal{B} \qquad T\ ;\ \mathcal{B}\vdash\mathcal{C}}{T\ ;\ \mathcal{A}\vdash\mathcal{C}} \qquad\qquad \frac{}{T\ ;\ \mathcal{A}\vdash\mathcal{A}}\ (\textsc{Axiom})$$

The identity laws (4.2) imply for instance that the three derivations below must be interpreted by the same morphism:

$$\frac{\dfrac{}{T\ ;\ \mathcal{A}\vdash\mathcal{A}} \qquad \dfrac{\mathscr{D}}{T\ ;\ \mathcal{A}\vdash\mathcal{B}}}{T\ ;\ \mathcal{A}\vdash\mathcal{B}} \qquad\qquad \frac{\mathscr{D}}{T\ ;\ \mathcal{A}\vdash\mathcal{B}} \qquad\qquad \frac{\dfrac{\mathscr{D}}{T\ ;\ \mathcal{A}\vdash\mathcal{B}} \qquad \dfrac{}{T\ ;\ \mathcal{B}\vdash\mathcal{B}}}{T\ ;\ \mathcal{A}\vdash\mathcal{B}} \tag{4.3}$$

4.2. Indexed Structure: Substitution and Quantification Rules. Our categories actually involve a slight generalization of the usual notion of acceptance (either with run trees or games) of automata. This generalization is induced by the axiomatization of quantification and substitution in categorical logic (see e.g. [Jac01, LS86]).

Let us briefly discuss the usual setting of first-order logic over a manysorted individual language. The categorical semantics of existential quantifications is given by an adjunction which is usually represented as

$$\frac{(\exists x)\varphi(x)\vdash\psi}{\varphi(x)\vdash\psi}\quad (x \text{ not free in } \psi) \tag{4.4}$$

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

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

In general, the variable x occurs in φ. As a consequence, in order to properly formulate (4.4) 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 (but 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 $\iota = \iota_1 \times \cdots \times \iota_n$ of $\mathbb{B}$, one devises a category $\mathbb{E}_\iota$ whose objects represent formulae with free variables among $x_{\iota_1}, \ldots, x_{\iota_n}$, 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 with free variables 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 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 (4.4) 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 $\pi^\star$:

$$\frac{(\exists x_\iota)\varphi(x_\iota, y_o) \vdash \psi(y_o)}{\varphi(x_\iota, y_o) \vdash \pi^\star(\psi)(x_\iota, y_o)} \tag{4.5}$$

(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.

Returning to automata and infinite trees, we take as base category the following category $\mathbf{T}$ of trees.

Definition 4.2.1 (The Base Category $\mathbf{T}$). *The objects of* $\mathbf{T}$ *are alphabets, and its morphisms from* Σ *to* Γ*, denoted* $M, N, L, \ldots$*, are functions of the form*

$$\bigcup_{n>0} (\Sigma^n \times \mathfrak{D}^{n-1}) \quad \longrightarrow \quad \Gamma$$

A $\mathbf{T}$-morphism $M \in \mathbf{T}[\Sigma, \Gamma]$ thus takes for each $n \in \mathbb{N}$ a sequence of input letters $\bar{\mathsf{a}} = \mathsf{a}_0 \cdot \ldots \cdot \mathsf{a}_n \in \Sigma^{n+1}$ and a sequence of tree directions $p = d_1 \cdot \ldots \cdot d_n \in \mathfrak{D}^n$ to an output letter $M(\bar{\mathsf{a}}, p) \in \Gamma$. In particular, we have $\mathbf{T}[\mathbf{1}, \Sigma] \simeq (\mathfrak{D}^* \to \Sigma)$, so each Σ-labeled $\mathfrak{D}$-ary tree T corresponds to a morphism $\dot{T} \in \mathbf{T}[\mathbf{1}, \Sigma]$. Moreover, $(\Sigma \to \Gamma)$-labeled trees $M : \mathfrak{D}^* \to (\Sigma \to \Gamma)$ induce

$\mathbf{T}$-morphisms from Σ to Γ.[1] $\mathbf{T}$-morphisms are composed in the expected way (see §6.4 and Prop. 6.4.3 for details and justification of this choice).

We therefore do not devise a single category $\mathbb{C}$, but a $\mathbf{T}$-indexed collection of categories $\mathbb{E}_\Sigma$, one for each alphabet Σ. Let us sketch the general idea with runs of non-deterministic automata. Given a non-deterministic automaton $\mathcal{A}$ over Γ and a morphism $M \in \mathbf{T}[\Sigma, \Gamma]$, a Σ-*run* of $\mathcal{A}$ on M is a tree

$$\rho \quad : \quad \mathfrak{D}^* \quad \longrightarrow \quad \Sigma \times Q_\mathcal{A}$$

such that $\rho(\varepsilon) = (\mathtt{a}_0, q^\imath_\mathcal{A})$ for some $\mathtt{a}_0 \in \Sigma$, and which respects the transition function

$$\partial_\mathcal{A} \quad : \quad Q_\mathcal{A} \times \Gamma \quad \longrightarrow \quad \mathcal{P}(\mathfrak{D} \to Q_\mathcal{A})$$

supplied with input letters $\mathtt{b} \in \Gamma$ computed by M from tree positions $p = d_1 \cdot \ldots \cdot d_n$ and sequences of input letters $\overline{\mathtt{a}} = \mathtt{a}_0 \cdot \ldots \cdot \mathtt{a}_n$ where $\mathtt{a}_k$ is given by the Σ-component of $\rho(d_1 \cdot \ldots \cdot d_k) \in \Sigma \times Q_\mathcal{A}$ (so $\mathtt{a}_0$ is given by $R(\varepsilon)$ and $\mathtt{a}_n$ is given by $R(p)$). Explicitly, ρ is a Σ-run tree when for p and $\overline{\mathtt{a}}$ as above, if $\rho(p)$ is labeled with state $q \in Q_\mathcal{A}$, then there exists a $\mathfrak{D}$-tuple $(q_d)_{d \in \mathfrak{D}} \in \partial_\mathcal{A}(q, \mathtt{b})$ with $\mathtt{b} = M(\overline{\mathtt{a}}, p)$ and such that for all $d \in \mathfrak{D}$, $\rho(p \cdot d)$ is labeled with state q_d. Such a Σ-run ρ is *accepting* if the $Q_\mathcal{A}$-labeled tree

$$p \in \mathfrak{D}^* \quad \longmapsto \quad \pi(\rho(p)) \in Q_\mathcal{A}$$

is accepting in the usual sense (where $\pi : \Sigma \times Q_\mathcal{A} \to Q_\mathcal{A}$ is the second projection), that is if all its infinite paths belong to $\Omega_\mathcal{A}$. We let $\Sigma \vdash \mathcal{A}(M)$ be the set of accepting Σ-run trees of $\mathcal{A}$ on M, and simply write $\mathcal{A}(M)$ for $\Sigma \vdash \mathcal{A}(M)$ when Σ is clear from the context.

Roughly speaking, for each Σ, the objects of the category $\mathbb{E}_\Sigma$ includes all sets of the form $\Sigma \vdash \mathcal{A}(M)$. Moreover, given $L \in \mathbf{T}[\Delta, \Sigma]$, the substitution functor

$$L^\star \quad : \quad \mathbb{E}_\Sigma \quad \longrightarrow \quad \mathbb{E}_\Delta$$

takes an $\mathbb{E}_\Sigma$-object $\Sigma \vdash \mathcal{A}(M)$ to the $\mathbb{E}_\Delta$-object $\Delta \vdash \mathcal{A}(M \circ L)$, where the $\mathbf{T}$-map $M \circ L \in \mathbf{T}[\Delta, \Gamma]$ is the $\mathbf{T}$-composition of L and M (assuming $M \in \mathbf{T}[\Sigma, \Gamma]$ as above).

This induces sequents generalizing (4.1). For instance, given $M \in \mathbf{T}[\Sigma, \Gamma]$, we have sequents of the form

$$M \ ; \ \mathcal{A}_1, \ldots, \mathcal{A}_n \vdash \mathcal{B} \tag{4.6}$$

where $\mathcal{A}_1, \ldots, \mathcal{A}_n$ and $\mathcal{B}$ are automata over Γ. Such sequents are to be thought about as our version of "*open* sequents" or "sequents with free variables" (here of sort Σ), with the usual implicit prenex universal quantification over these, and are to be interpreted as a morphism in the category $\mathbb{E}_\Sigma$ (the *fibre* over Σ). Substitution functors such as $L^\star : \mathbb{E}_\Sigma \to \mathbb{E}_\Delta$ above act in the deduction system via a substitution rule

$$(\textsc{Subst}) \ \frac{M \ ; \ \mathcal{A}_1, \ldots \mathcal{A}_n \vdash \mathcal{B}}{M \circ L \ ; \ \mathcal{A}_1, \ldots \mathcal{A}_n \vdash \mathcal{B}} \qquad (\text{where } M \in \mathbf{T}[\Sigma, \Gamma] \text{ and } L \in \mathbf{T}[\Delta, \Sigma]) \tag{4.7}$$

Let us briefly sketch the most important instances of this construction.

(a) Consider a $\mathbf{T}$-map $\dot{T} : \mathbf{T}[\mathbf{1}, \Sigma]$ representing a tree $T : \mathfrak{D}^* \to \Sigma$. Then the accepting runs of $\mathcal{A}$ on T are in bijection with the accepting $\mathbf{1}$-run trees of $\mathcal{A}$ on $\dot{T}$:

$$(\mathbf{1} \vdash \mathcal{A}(\dot{T})) \quad \simeq \quad \mathcal{A}(T)$$

[1]The morphisms from Σ to Γ of the base category of [Rib15] are restricted to $(\Sigma \to \Gamma)$-labeled trees.

Sequents of the form (4.6) thus indeed generalize sequents of the form

$$T \ ; \ \mathcal{A}_1 \, , \ \dots \ , \ \mathcal{A}_n \vdash \mathcal{B}$$

with $T : \mathfrak{D}^* \to \Sigma$ (as depicted in (4.1)), which are to be interpreted in the category $\mathbb{E}_{\mathbf{1}}$ (the fibre over $\mathbf{1}$), and are to be thought about as representing *closed* statements.

(b) Given a non-deterministic automaton $\mathcal{A}$ over Σ, we write $\Sigma \vdash \mathcal{A}$ (or even just $\mathcal{A}$ when no ambiguity arises) for $\Sigma \vdash \mathcal{A}(\mathrm{Id}_\Sigma)$ where the $\mathbf{T}$-identity $\mathrm{Id}_\Sigma \in \mathbf{T}[\Sigma, \Sigma]$ is given by

$$\mathrm{Id}_\Sigma(\overline{\mathtt{a}} \cdot \mathtt{a}, p) \quad := \quad \mathtt{a}$$

Consider now another automaton $\mathcal{B}$, also over Σ. Then we write

$$\Sigma \ ; \ \mathcal{A} \vdash \mathcal{B} \tag{4.8}$$

(or even $\mathcal{A} \vdash \mathcal{B}$) for the sequent $\mathrm{Id}_\Sigma \ ; \ \mathcal{A} \vdash \mathcal{B}$. The provability interpretation of (4.8) is that if (4.8) is provable, then $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$. The *computational interpretation* of (4.8) consists of a uniform simulation of $\mathcal{A}$ by $\mathcal{B}$ (generalizing the notion used with the *guidable automata* of [CL08]). Moreover, given a Σ-labeled tree T seen as a morphism $\dot{T} \in \mathbf{T}[\mathbf{1}, \Sigma]$, the interpretation of the substitution rule

$$\frac{\Sigma \ ; \ \mathcal{A} \vdash \mathcal{B}}{\dot{T} \ ; \ \mathcal{A} \vdash \mathcal{B}}$$

takes a morphism $\sigma \in \mathbb{E}_\Sigma[\mathcal{A}, \mathcal{B}]$ to a function $\dot{T}^\star(\sigma) : \mathcal{A}(T) \to \mathcal{B}(T)$.

(c) Any ordinary function $\mathtt{f} : \Sigma \to \Gamma$ induces a morphism $[\mathtt{f}] \in \mathbf{T}[\Sigma, \Gamma]$ defined as

$$[\mathtt{f}] \quad : \quad (\overline{\mathtt{a}} \cdot \mathtt{a}, p) \quad \longmapsto \quad \mathtt{f}(\mathtt{a})$$

The action of the substitution functor $[\mathtt{f}]^\star : \mathbb{E}_\Gamma \to \mathbb{E}_\Sigma$ on $\mathbb{E}_\Gamma$-objects of the form $\Gamma \vdash \mathcal{A}$ can be internalized in automata. We indeed have

$$[\mathtt{f}]^\star(\Gamma \vdash \mathcal{A}) \quad = \quad \Sigma \vdash \mathcal{A}([\mathtt{f}]) \quad = \quad \Sigma \vdash \mathcal{A}[\mathtt{f}]$$

where the automaton $\mathcal{A}[\mathtt{f}]$ over Σ is defined as $\mathcal{A}$ but with transition function:

$$\begin{array}{rcrcl} \partial_{\mathcal{A}[\mathtt{f}]} & : & Q_\mathcal{A} \times \Gamma & \longrightarrow & \mathcal{P}(\mathfrak{D} \to Q_\mathcal{A}) \\ & & (q, \mathtt{b}) & \longmapsto & \partial_\mathcal{A}(q, \mathtt{f}(\mathtt{b})) \end{array}$$

In particular:

(i) $\mathbf{T}$-maps from $\Sigma \times \Gamma$ to Σ indeed include projections $[\pi] : \mathfrak{D}^* \to (\Sigma \times \Gamma \to \Sigma)$ induced by $\mathbf{Set}$-projections $\pi : \Sigma \times \Gamma \to \Sigma$.

(ii) Consider automata $\mathcal{A}_1, \dots, \mathcal{A}_n$ and $\mathcal{B}$, with $\mathcal{A}_i$ over Σ_i and $\mathcal{B}$ over Γ. Consider furthermore $\mathbf{T}$-morphisms $M_i \in \mathbf{T}[\Delta, \Sigma_i]$ and $L \in \mathbf{T}[\Delta, \Gamma]$. Then we write

$$\Delta \ ; \ \mathcal{A}_1(M_1) \, , \ \dots \ , \ \mathcal{A}_n(M_n) \vdash \mathcal{B}(L)$$

for the sequent

$$\langle M_1, \dots, M_n, L \rangle \ ; \ \mathcal{A}_1[\pi_1] \, , \ \dots \ , \ \mathcal{A}_n[\pi_n] \vdash \mathcal{B}[\pi]$$

where

$$\langle M_1, \ldots, M_n, L\rangle \ \in \ \mathbf{T}[\Delta, \Sigma_1 \times \cdots \times \Sigma_n \times \Gamma]$$

is the **T**-tupling of $M_1, \ldots, M_n, L$ (see §6.4.2) and where the π_i's and π are suitable projections:

$$\begin{array}{lcllcl} \pi_i & : & \Sigma_1 \times \cdots \times \Sigma_n \times \Gamma & \longrightarrow & \Sigma_i \\ \pi & : & \Sigma_1 \times \cdots \times \Sigma_n \times \Gamma & \longrightarrow & \Gamma \end{array}$$

Unless otherwise stated, all the sequents seen up to now must from now on be thought about as being of the more general form (4.8), that is a with a **T**-map M (of appropriate type) instead of the labeled tree T.

4.3. Toward a Semantics for Implications. The *provability interpretation* of sequents tells us that in sequents of the form

$$M \ ; \ \mathcal{A} \vdash \mathcal{B} \tag{4.9}$$

the symbol $\vdash$ is a form of implication. We shall see later on that this implication can be internalized in automata, but this would lead us outside of non-deterministic automata (see Chap. 7). For the moment let us sketch some salient consequences this imposes to the interpretation of the symbol $\vdash$ in sequents of the form (4.9).

Assume that proofs of our deduction system are interpreted in categories $\mathbb{E}_{(-)}$ indexed over **T**. Then, internalizing $\vdash$ in automata will imply that given automata $\mathcal{A}$ and $\mathcal{B}$ over Σ there is an automaton $(\mathcal{A} \multimap \mathcal{B})$ over Σ such that for each tree $T : \mathfrak{D}^* \to \Sigma$ there is a bijection

$$\mathbb{E}_{\mathbf{1}}[\mathcal{A}(\dot{T}), \mathcal{B}(\dot{T})] \quad \simeq \quad \mathbf{1} \vdash (\mathcal{A} \multimap \mathcal{B})(\dot{T})$$

that we informally write as

$$\dot{T} \ ; \ \mathcal{A} \vdash \mathcal{B} \qquad \simeq \qquad \mathbf{1} \vdash (\mathcal{A} \multimap \mathcal{B})(\dot{T})$$

In other words, morphisms in the interpretation of $\dot{T} \ ; \ \mathcal{A} \vdash \mathcal{B}$ will correspond to the runs of an automaton $(\mathcal{A} \multimap \mathcal{B})$ on T. This could suggest to interpret $\dot{T} \ ; \ \mathcal{A} \vdash \mathcal{B}$ as the runs of an automaton of the form ${\sim}\mathcal{A} \vee \mathcal{B}$ over T, where ${\sim}\mathcal{A}$ is the complement of $\mathcal{A}$ (in the sense of §1.3) and $(-) \vee (-)$ is a disjunction on automata. Let us rule out this possibility, at least for the natural implementation of $(-) \vee (-)$ with an *additive* disjunction $(-) \oplus (-)$. Given automata $\mathcal{A}_1$ and $\mathcal{A}_2$, both over Σ and with $\mathcal{A}_i = (Q_{\mathcal{A}_i}, q^\imath_{\mathcal{A}_i}, \partial_{\mathcal{A}_i}, \Omega_{\mathcal{A}_i})$, the non-deterministic automaton $\mathcal{A}_1 \oplus \mathcal{A}_2$ over Σ is

$$\mathcal{A}_1 \oplus \mathcal{A}_2 \quad := \quad (Q_{\mathcal{A}_1} + Q_{\mathcal{A}_2} + \mathbf{1} \,,\, \bullet \,,\, \partial_{\mathcal{A}_1 \oplus \mathcal{A}_2} \,,\, \Omega_{\mathcal{A}_1 \oplus \mathcal{A}_2})$$

where, via the embedding of $Q_{\mathcal{A}_1}^{\mathfrak{D}} + Q_{\mathcal{A}_2}^{\mathfrak{D}}$ into $(Q_{\mathcal{A}_1} + Q_{\mathcal{A}_2})^{\mathfrak{D}}$, we let

$$\partial_{\mathcal{A}_1 \oplus \mathcal{A}_2}(q, \mathtt{a}) \quad := \quad \begin{cases} \partial_{\mathcal{A}_1}(q^\imath_{\mathcal{A}_1}, \mathtt{a}) + \partial_{\mathcal{A}_2}(q^\imath_{\mathcal{A}_2}, \mathtt{a}) & \text{if } q = \bullet \in \mathbf{1} \\ \partial_{\mathcal{A}_i}(q, \mathtt{a}) & \text{if } q \in Q_{\mathcal{A}_i} \end{cases}$$

and where $\bullet, q_1, q_2, \ldots \in \Omega_{\mathcal{A}_1 \oplus \mathcal{A}_2}$ iff either $q^\imath_{\mathcal{A}_1}, q_1, q_2, \ldots \in \Omega_{\mathcal{A}_1}$ or $q^\imath_{\mathcal{A}_2}, q_1, q_2, \ldots \in \Omega_{\mathcal{A}_2}$.

Note that in **Set**, for every $M : \mathfrak{D}^* \to (\Gamma \to \Sigma)$ we have

$$(\mathcal{A}_1 \oplus \mathcal{A}_2)(M) \quad \simeq \quad \mathcal{A}_1(M) + \mathcal{A}_2(M)$$

so in particular

$$\mathcal{L}(\mathcal{A}_1 \oplus \mathcal{A}_2) \quad = \quad \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$$

Assume now that we take for $\mathbb{E}_{\mathbf{1}}[\mathcal{A}(\dot{T}), \mathcal{B}(\dot{T})]$ the set of runs of $({\sim}\mathcal{A} \oplus \mathcal{B})$ on T, that is the disjoint union ${\sim}\mathcal{A}(T) + \mathcal{B}(T)$. Then one faces the following difficulties.

- We have to devise identity morphisms, say

$$\mathrm{id}_{\mathcal{A}(\dot{T})} \quad \in \quad {\sim}\mathcal{A}(T) + \mathcal{A}(T)$$

One may take for $\mathrm{id}_{\mathcal{A}(\dot{T})}$ either an accepting run of $\mathcal{A}$ on T or an accepting run of ${\sim}\mathcal{A}$ on T. But this raises two problems. First, it may be undecidable whether a possibly non-recursive tree is accepted or rejected by a given automaton. So this precludes any *general and effective* computational interpretation of the deduction system. Second, even if we restrict to trees T for which acceptance is known to be decidable (e.g. trees generated by *higher-order recursion schemes* [Ong06]), there seem to be no *canonical choice* of an actual accepting run $\mathrm{id}_{\mathcal{A}(\dot{T})} \in {\sim}\mathcal{A}(T) + \mathcal{A}(T)$.

- It is not clear how to define composition, say

$$(-) \circ (-) \quad : \quad ({\sim}\mathcal{B}(T) + \mathcal{C}(T)) \times ({\sim}\mathcal{A}(T) + \mathcal{B}(T)) \quad \longrightarrow \quad {\sim}\mathcal{A}(T) + \mathcal{C}(T)$$

Given run trees, say

$$\rho_{\mathcal{C}(T)} \in \mathcal{C}(T) \subseteq {\sim}\mathcal{B}(T) + \mathcal{C}(T) \qquad \text{and} \qquad \rho_{\sim\mathcal{A}(T)} \in {\sim}\mathcal{A}(T) \subseteq {\sim}\mathcal{A}(T) + \mathcal{B}(T)$$

there seems to be no obvious choice for $\rho_{\mathcal{C}(T)} \circ \rho_{\sim\mathcal{A}(T)} \in {\sim}\mathcal{A}(T) + \mathcal{C}(T)$. Both

$$\rho_{\mathcal{C}(T)} \circ \rho_{\sim\mathcal{A}(T)} \quad := \quad \rho_{\mathcal{C}(T)} \qquad \text{and} \qquad \rho_{\mathcal{C}(T)} \circ \rho_{\sim\mathcal{A}(T)} \quad := \quad \rho_{\sim\mathcal{A}(T)}$$

may seem reasonable. But each of them breaks one of the equalities between the interpretations of the derivations depicted in (4.3).

The methodology of linear logic may suggest here to devise a linear implication of the form

$$\mathcal{A} \multimap \mathcal{B} \quad := \quad \mathcal{A}^{\perp\!\!\!\perp} \parr \mathcal{B}$$

where $\parr$ is a dual of the direct product $\otimes$ (see §4.4 below and §1.5). relying on a Cartesian product of states and evaluating its arguments in parallel, with acceptance given by a disjunction. However, in contrast with ω-word automata [PR18b] (see §7.5), such a connective does not seem to exist on tree automata. The reason is that the universal quantification on paths (in the definition of acceptance) does not commute with disjunction.

4.4. The (Synchronous) Direct Product of (Non-Deterministic) Automata. Returning to sequents of the form

$$M \ ; \ \mathcal{A}_1, \ldots, \mathcal{A}_n \vdash \mathcal{B} \tag{4.10}$$

the provability interpretation tells us that the left commas in $\mathcal{A}_1, \ldots, \mathcal{A}_n$ correspond to a form of conjunction. We now sketch how to interpret these commas with a direct product of non-deterministic automata. Our aim here is to pave the way to §4.5, in which we discuss how Linear Logic [Gir87] enters the picture.

The *direct product* $\mathcal{A}_1 \otimes \mathcal{A}_2$ of the non-deterministic automata $\mathcal{A}_i = (Q_{\mathcal{A}_i}, q^{\imath}_{\mathcal{A}_i}, \partial_{\mathcal{A}_i}, \Omega_{\mathcal{A}_i})$, both over Σ, is the non-deterministic automaton over Σ

$$\mathcal{A}_1 \otimes \mathcal{A}_2 \quad := \quad \left(Q_{\mathcal{A}_1} \times Q_{\mathcal{A}_2}\, , \, (q^{\imath}_{\mathcal{A}_1}, q^{\imath}_{\mathcal{A}_2})\, , \, \partial_{\mathcal{A}_1 \otimes \mathcal{A}_2}\, , \, \Omega_{\mathcal{A}_1 \otimes \mathcal{A}_2}\right)$$

with

$$\partial_{\mathcal{A}_1\otimes\mathcal{A}_2}((q_1,q_2),\mathtt{a}) \quad := \quad \{\langle \mathtt{g}_1,\mathtt{g}_2\rangle : \mathfrak{D} \to Q_{\mathcal{A}_1} \times Q_{\mathcal{A}_2} \mid \mathtt{g}_i \in \partial_{\mathcal{A}_i}(q_i,\mathtt{a}) \text{ for } i = 1,2\}$$

and where $\Omega_{\mathcal{A}_1\otimes\mathcal{A}_2}$ is $\Omega_{\mathcal{A}_1} \times \Omega_{\mathcal{A}_2}$ modulo $(Q_{\mathcal{A}_1} \times Q_{\mathcal{A}_2})^\omega \simeq Q^\omega_{\mathcal{A}_1} \times Q^\omega_{\mathcal{A}_2}$. For every tree T, the (accepting) runs of $\mathcal{A}_1 \otimes \mathcal{A}_2$ on T are exactly[2] the pairs $\langle \rho_1, \rho_2\rangle : \mathfrak{D}^* \to Q_{\mathcal{A}_1} \times Q_{\mathcal{A}_2}$ of (accepting) runs of $\mathcal{A}_1$ and $\mathcal{A}_2$ over T. We therefore have, in the category **Set**

$$(\mathcal{A}_1 \otimes \mathcal{A}_2)(T) \quad \simeq \quad \mathcal{A}_1(T) \times \mathcal{A}_2(T) \tag{4.11}$$

from which we immediately get

$$\mathcal{L}(\mathcal{A}_1 \otimes \mathcal{A}_2) \quad = \quad \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$$

For similar reasons, this direct product $(-) \otimes (-)$ on is Cartesian in the categories of [Rib15] provided one restricts to *total* ND automata.[3] This implies that we can equip total ND automata with the deduction rules of a Cartesian product, such as the following (where $\mathbf{I}$ is a unit automaton similar to that of Ex. 7.0.2.(i)):

$$\begin{array}{cc} (\textsc{Left } \otimes)\ \dfrac{M \,;\, \overline{\mathcal{A}},\, \mathcal{A},\, \mathcal{B},\, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \,;\, \overline{\mathcal{A}},\, \mathcal{A} \otimes \mathcal{B},\, \overline{\mathcal{B}} \vdash \mathcal{C}} & \dfrac{M \,;\, \overline{\mathcal{A}} \vdash \mathcal{A} \qquad M \,;\, \overline{\mathcal{B}} \vdash \mathcal{B}}{M \,;\, \overline{\mathcal{A}},\, \overline{\mathcal{B}} \vdash \mathcal{A} \otimes \mathcal{B}}\ (\textsc{Right } \otimes) \\[2ex] (\textsc{Left } \mathbf{I})\ \dfrac{M \,;\, \overline{\mathcal{A}},\, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \,;\, \overline{\mathcal{A}},\, \mathbf{I},\, \overline{\mathcal{B}} \vdash \mathcal{C}} & \dfrac{}{M \,;\, \vdash \mathbf{I}}\ (\textsc{Right } \mathbf{I}) \end{array} \tag{4.12}$$

together with the structural *exchange rule*:

$$(\textsc{Exchange})\ \dfrac{M \,;\, \overline{\mathcal{A}},\, \mathcal{A},\, \mathcal{B},\, \overline{\mathcal{C}} \vdash \mathcal{C}}{M \,;\, \overline{\mathcal{A}},\, \mathcal{B},\, \mathcal{A},\, \overline{\mathcal{C}} \vdash \mathcal{C}} \tag{4.13}$$

as well as the structural *weakening* and *contraction* rules:

$$(\textsc{Weak})\ \dfrac{M \,;\, \overline{\mathcal{A}},\, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \,;\, \overline{\mathcal{A}},\, \mathcal{A},\, \overline{\mathcal{B}} \vdash \mathcal{C}} \qquad \dfrac{M \,;\, \overline{\mathcal{A}},\, \mathcal{A},\, \mathcal{A},\, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \,;\, \overline{\mathcal{A}},\, \mathcal{A},\, \overline{\mathcal{B}} \vdash \mathcal{C}}\ (\textsc{Contr}) \tag{4.14}$$

and the following general (multiplicative) cut rule:

$$(\textsc{Cut})\ \dfrac{M \,;\, \overline{\mathcal{A}} \vdash \mathcal{A} \qquad M \,;\, \overline{\mathcal{B}},\, \mathcal{A},\, \overline{\mathcal{C}} \vdash \mathcal{C}}{M \,;\, \overline{\mathcal{B}},\, \overline{\mathcal{A}},\, \overline{\mathcal{C}} \vdash \mathcal{C}} \tag{4.15}$$

To summarize, the left commas in sequents of the form (4.10) can be internalized as a product $(\otimes, \mathbf{I})$. This product is Cartesian on total ND automata, and its deduction rules are induced by this structure.

4.5. Alternating Automata and Linear Logic. With respect to our context, the basic insight of Linear Logic [Gir87], is that having an explicit control on the weakening and contraction structural rules (4.14) gives rise to a decomposition of the usual intuitionistic connectives $\wedge, \to$ into more refined ones (usually denoted $\otimes, \&, !, \multimap$). In a lot of cases this allows, thanks to the Curry-Howard correspondence, for refined constructions of models of programming languages

[2]Because universal quantifications commute over conjunctions!

[3]A *non-deterministic* automaton is *total* if the empty set is not in the range of its transition function.

based on (typed) λ-calculi (see e.g. [AC98]). Let us briefly discuss what this amounts to for the conjunction $\otimes$.

First, when suppressing the structural rules (Weak) and (Contr), the rules (4.12) and (4.13) only specify a *symmetric monoidal structure* (see e.g. [Mel09] for definitions), a notion weaker than Cartesian products. This is pertinent in our context because the product $(-)\otimes(-)$ defined in §4.4 on non-deterministic automata extends to (total[4]) alternating automata, but induces a symmetric monoidal product which is not Cartesian.

Given alternating automata $\mathcal{A}$ and $\mathcal{B}$ over Σ, the automaton $\mathcal{A}\otimes\mathcal{B}$ over Σ has state set $Q_{\mathcal{A}} \times Q_{\mathcal{B}}$, and evaluates $\mathcal{A}$ and $\mathcal{B}$ along common paths $p \in \mathfrak{D}^*$ (see [Rib15] for details). Now, recall that with alternating automata, O can choose states in addition to tree directions. Hence, given a P-strategy on $(\mathcal{A}\otimes\mathcal{B})(T)$, and given a branch of this strategy following a given path $p \in \mathfrak{D}^*$, it is possible for P to make different choices according to previous O-moves. In particular, some choice of P in component $\mathcal{A}$ may depend on previous O-moves in $\mathcal{B}$. So a P-strategy on $(\mathcal{A}\otimes\mathcal{B})(T)$ may not uniquely determine a pair of strategies in $\mathcal{A}(T)\times\mathcal{B}(T)$. Note that this was not possible with non-deterministic automata, since $p \in \mathfrak{D}^*$ uniquely determines the previous O-moves.

Second, the structural rules (Weak) and (Contr) are restored in Linear Logic for an *exponential modality* $!(-)$:

$$\frac{M \;;\; \mathcal{A}_1, \dots, \dots, \mathcal{A}_n \vdash \mathcal{B}}{M \;;\; \mathcal{A}_1, \dots, !\mathcal{A}, \dots, \mathcal{A}_n \vdash \mathcal{B}} \qquad \frac{M \;;\; \mathcal{A}_1, \dots, !\mathcal{A}_i, !\mathcal{A}_i, \dots \mathcal{A}_n \vdash \mathcal{B}}{M \;;\; \mathcal{A}_1, \dots, !\mathcal{A}_i, \dots, \mathcal{A}_n \vdash \mathcal{B}} \tag{4.16}$$

The modality $!(-)$ is itself subject to specific introduction rules, called *dereliction* and *promotion*:

$$\frac{M \;;\; \mathcal{A}_1, \dots, \mathcal{A}_i, \dots, \mathcal{A}_n \vdash \mathcal{B}}{M \;;\; \mathcal{A}_1, \dots, !\mathcal{A}_i, \dots, \mathcal{A}_n \vdash \mathcal{B}} \qquad \frac{M \;;\; !\mathcal{A}_1, \dots, !\mathcal{A}_n \vdash \mathcal{B}}{M \;;\; !\mathcal{A}_1, \dots, !\mathcal{A}_n \vdash !\mathcal{B}} \tag{4.17}$$

Then (but see also [Gir87, AC98, Mel09] for details), the categorical interpretation of proofs implies that the monoidal product $(\otimes, \mathbf{I})$ is Cartesian on objects of the form $!\mathcal{A}$.[5] This indicates that non-deterministic automata behave as objects of the form $!\mathcal{A}$, and it turns out that to some extent, the powerset construction translating an alternating automaton to an equivalent non-deterministic one (the *Simulation Theorem* [MS87, EJ91, MS95]), corresponds to an $!(-)$-modality of intuitionistic linear logic. In particular, all the $!(-)$-rules (4.16) and (4.17) can be interpreted in our categories.[6] (But unfortunately, this interpretation is not compatible with usual cut-elimination, because the operation $!(-)$ fails to be a functor.)

This implies that an intuitionistic sequent

$$M \;;\; \mathcal{A}_1, \dots, \mathcal{A}_n \vdash \mathcal{B}$$

where the left commas behave as a Cartesian product, corresponds to the linear sequent

$$M \;;\; !\mathcal{A}_1, \dots, !\mathcal{A}_n \vdash \mathcal{B}$$

where the left commas behave as a symmetric monoidal product $(-)\otimes(-)$.

[4] *Total* alternating automata were called *complete* in [Rib15].
[5] Technically, objects under a $!(-)$ are *commutative comonoids* (see e.g. [Mel09]).
[6] (Weak) actually holds (in a non-canonical way) for total alternating automata (*i.e.* the ! is not strictly necessary in the conclusion).

4.6. Game Semantics: Linear Arrow Games and Copy-Cat. Our solution to the composition problem discussed in §4.3 is based on the technology of *Game Semantics*, which provides by now well understood ways of building categories of games, with strategies as morphisms. We present here the usual notion of morphism of a basic game semantics called *Simple Games* (see e.g. [Abr97, Hyl97]). Recall from §4.3 that a sequent $M\ ;\ \mathcal{A} \vdash \mathcal{B}$ (for $\mathcal{A}, \mathcal{B}$ usual non-deterministic automata) should be thought as a form of implication, but that the runs of the automaton $\sim\mathcal{A} \oplus \mathcal{B}$ seemed not to convey the right information. The first encountered difficulty concerned the existence of *canonical* identities $\mathrm{id}_{\mathcal{A}(M)} \in \mathbb{E}_\Sigma[\mathcal{A}(M), \mathcal{A}(M)]$ if the homset $\mathbb{E}_\Sigma[\mathcal{A}(M), \mathcal{A}(M)]$ were to be the set of accepting runs or winning P-strategies $(\sim\mathcal{A})(M) + \mathcal{A}(M)$. The solution of game semantics is to devise, from component games A and B of the same polarity, an implication game $A \multimap_{\mathbf{SG}} B$ in which the game B is interleaved with a copy of A of reversed polarity.[7] More precisely:

Definition 4.6.1 (Linear Arrow Games)**.** *Given polarized simple games A and B of the same* polarity*, the* linear arrow game *$A \multimap_{\mathbf{SG}} B$ is the negative game*

$$A \multimap_{\mathbf{SG}} B \quad := \quad (A_{\mathsf{O}} + B_{\mathsf{P}},\ A_{\mathsf{P}} + B_{\mathsf{O}},\ L_{A \multimap_{\mathbf{SG}} B})$$

where $L_{A \multimap_{\mathbf{SG}} B}$ consists of those negative plays s such that $s_{\restriction A} \in L_A$ and $s_{\restriction B} \in L_B$, where $s_{\restriction A}$ is the restriction of s to $A_{\mathsf{P}} + A_{\mathsf{O}}$, and similarly for $s_{\restriction B}$.

Hence, O always begins in $A \multimap_{\mathbf{SG}} B$, and then plays alternate between P and O. Note that the roles of P and O are reversed in component A and are preserved in component B (*i.e.* P in $A \multimap_{\mathbf{SG}} B$ plays as O in A and as P in B).

An important basic property of $A \multimap_{\mathbf{SG}} B$, known as the *switching condition*, is that O must stay in the same component as the previous move so that only P can switch between components A and B:

- *Switching Condition:* Given a legal O-play $s = t \cdot n \cdot m$, either n, m are both in component A, or they are both in component B.

Indeed, note that since $A \multimap_{\mathbf{SG}} B$ is negative, its legal O-plays are of odd-length. So if s is a legal O-play, then the lengths of $s_{\restriction A}$ and $s_{\restriction B}$ cannot have the same parity. Assume now that $s = t \cdot n \cdot m$ with n and m in different components. Since A and B are assumed to be of the same polarity, the moves n and m are of different polarities w.r.t. A and B, so they are of the same polarity as moves of $A \multimap_{\mathbf{SG}} B$ (as $s_{\restriction A}$ and $s_{\restriction B}$ have lengths of opposite parity), contradicting the legality of s.

Simple games and (winning) strategies form a category $\mathbf{SG}^{(\mathrm{W})}$, whose objects are simple games (with winning), and whose morphisms are (winning) P-strategies $\sigma : A \multimap_{\mathbf{SG}} B$. We refer to [Abr97, Hyl97, AC98] for full treatments, and in particular to [Abr97, Hyl97] for totality and winning. Actually, the general notion of winning in games of the form $A \multimap_{\mathbf{SG}} B$ is a bit technical. Fortunately, we only need to consider the case of infinite plays on $A \multimap_{\mathbf{SG}} B$ whose projections on A and B are both infinite. We say that such a play is winning for P in $A \multimap_{\mathbf{SG}} B$ iff its projection on B is winning for P whenever its projection on A is winning for P (with the original polarity of A).

Consider now the definition of the identity strategy id_A in $A \multimap_{\mathbf{SG}} A$ for $A = (U, X)$ a full positive game (see Ex. A.0.2). Since O must begin in $A \multimap_{\mathbf{SG}} A$, but it is P who begins in the right copy of A, it follows that O must begin in the left copy of A (taking the role of P in

[7] Noted $\overline{A}$ in App. A.

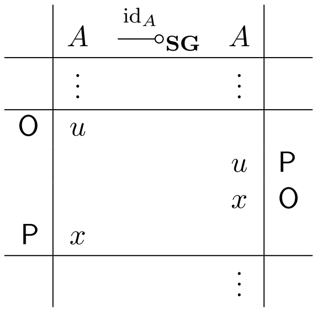


Figure 4.1.: A play of the *copy-cat* identity strategy id.

that component). It is then easy to define an identity "*copy-cat*" strategy for P, which always switches component and copies the previous O-move from the other component. A play of this strategy is depicted in Fig. 4.1 (where plays grow from top to bottom). Formally, id_A is the unique strategy in $A \multimap_{\mathbf{SG}} A$ such that

$$\mathrm{id}_A \quad = \quad \{s \in L_{A^0 \multimap_{\mathbf{SG}} A^1} \mid s_{\restriction A^0} = s_{\restriction A^1}\} \tag{4.18}$$

(where we have written $A \multimap_{\mathbf{SG}} A$ as $A^0 \multimap_{\mathbf{SG}} A^1$ in order to distinguish the two copies of A).

In particular, the same (infinite) sequences of moves are produced by id_A in both copies of A. Assuming that A is equipped with a winning condition $\mathcal{W}_A$, such sequences are either winning for P in A or are winning for O in A. So they are winning for P in $A \multimap_{\mathbf{SG}} A$.

Let us finally say a few words on composition of strategies, referring to e.g. [Abr97, Hyl97] for details. The idea is that given strategies $\sigma : A \multimap_{\mathbf{SG}} B$ and $\tau : B \multimap_{\mathbf{SG}} C$, their composite

$$\tau \circ \sigma \quad : \quad A \quad \multimap_{\mathbf{DZ}} \quad C$$

is obtained by letting σ and τ interact in their common component B. The crucial observation is that in an interaction of σ and τ in component B, all the P-moves are played by σ and all the O-moves are played by τ. It follows that the interactions of σ and τ in component B are completely determined by σ and τ and the O-moves in $A \multimap_{\mathbf{SG}} C$. The composite strategy $\tau \circ \sigma$ is then obtained by hiding the interaction of σ and τ in their common component B.

In the particular case of full positive games $A = (U, X)$, $B = (V, Y)$ and $C = (W, Z)$, and strategies $\sigma : A \multimap_{\mathbf{SG}} B$ and $\tau : B \multimap_{\mathbf{SG}} C$ playing in the *zig-zag* way depicted in Fig. 4.2 (top), an interaction of σ and τ is depicted in Fig. 4.2 (middle), and the composite strategy $\tau \circ \sigma$ plays as in Fig. 4.2 (bottom).

4.7. The Hyland & Schalk Functor. Hyland & Schalk have presented in [HS99] a faithful functor, that we denote HS, from simple games to the category **Rel** of sets and relations:[8]

$$\mathrm{HS} \quad : \quad \mathbf{SG}^{(\mathrm{W})} \quad \longrightarrow \quad \mathbf{Rel}$$

The functor HS maps a simple game to its set of legal plays, and a strategy $\sigma : A \multimap_{\mathbf{SG}} B$ to

$$\mathrm{HS}(\sigma) \quad := \quad \{(s_{\restriction A}, s_{\restriction B}) \mid s \in \sigma\} \quad \subseteq \quad L_A \times L_B$$

[8]See also App. 4 in the full version of [Rib15].

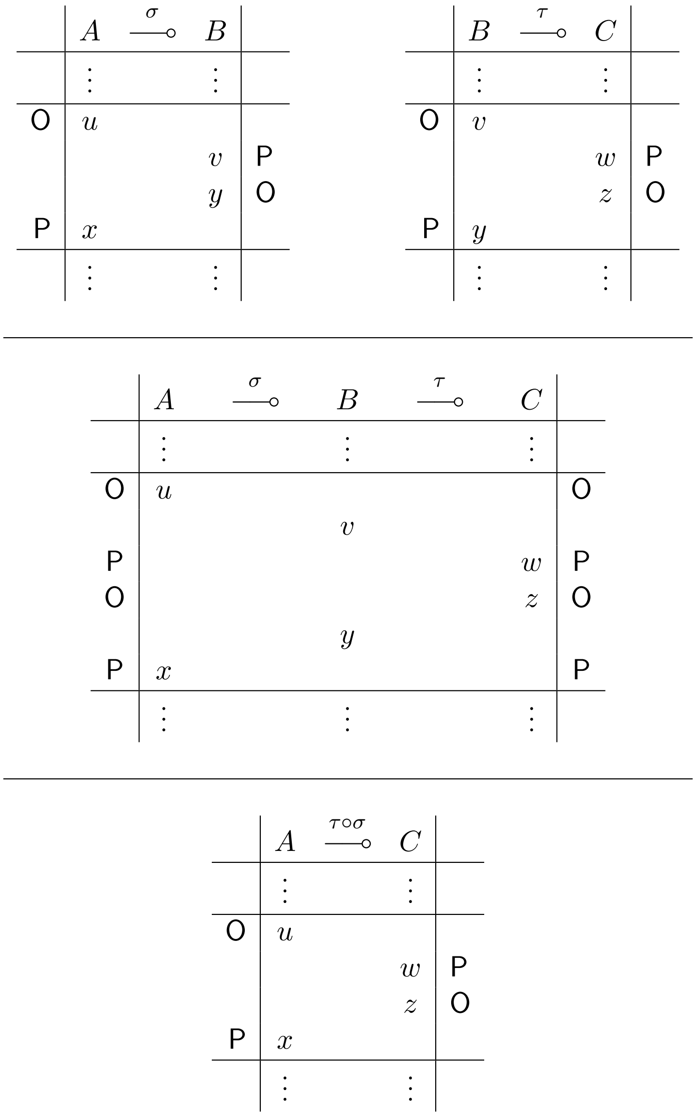


Figure 4.2.: An interaction of strategies on full positive games.

Hence strategies $\sigma : A \multimap_{\mathbf{SG}} B$ can be faithfully represented as spans of sets (where the arrow are the obvious projections)

$$L_A \longleftarrow \mathrm{HS}(\sigma) \longrightarrow L_B$$

and moreover, composition and identities in $\mathbf{SG}^{(\mathrm{W})}$ can be recovered from composition and identities in **Rel**. Indeed, the identity strategy id_A is the unique strategy such that $\mathrm{HS}(\mathrm{id}_A)$ is the identity relation on L_A, namely

$$\mathrm{HS}(\mathrm{id}_A) \quad = \quad \{(s, s) \mid s \in L_A\}$$

while strategies can be composed as relations:

$$\mathrm{HS}(\tau \circ \sigma) \quad = \quad \mathrm{HS}(\tau) \circ \mathrm{HS}(\sigma)$$

Moreover, it is easy to check (and folklore) that identity strategies and composition of strategies, when seen as relations, are given by pullbacks. First, $\mathrm{HS}(\mathrm{id}_A)$ is the pullback of the **Set**-identity $L_A \to L_A$ with itself in **Set**:

$$\begin{array}{ccc}
\mathrm{HS}(\mathrm{id}_A) & \longrightarrow & L_A \\
\downarrow & \lrcorner & \downarrow \\
L_A & \longrightarrow & L_A
\end{array}$$

Second, given $\sigma : A \multimap_{\mathbf{SG}} B$ and $\tau : B \multimap_{\mathbf{SG}} C$ we have the following pullback in **Set**:

$$\begin{array}{ccccc}
\mathrm{HS}(\tau \circ \sigma) & \longrightarrow & \mathrm{HS}(\tau) & \longrightarrow & L_C \\
\downarrow & \lrcorner & \downarrow & & \\
\mathrm{HS}(\sigma) & \longrightarrow & L_B & & \\
\downarrow & & & & \\
L_A & & & &
\end{array}$$

The description of $\mathrm{HS}(\tau \circ \sigma)$ by the above pullback amounts to the following relational version of the usual *Zipping Lemma* of game semantics, stating that the interactions of $\sigma : A \multimap_{\mathbf{SG}} B$ and $\tau : B \multimap_{\mathbf{SG}} C$ in component B are completely determined by the O-moves in components A and C (with the polarities of $A \multimap_{\mathbf{SG}} C$).

Lemma 4.7.1 (Relational Zipping)**.** *Given total zig-zag $\sigma : A \multimap_{\mathbf{DZ}} B$ and $\tau : B \multimap_{\mathbf{DZ}} C$, and given $(t_A, t_C) \in \mathrm{HS}(\tau) \circ \mathrm{HS}(\sigma)$, there is exactly one legal play $t_B \in L_B$ such that $(t_A, t_B) \in \mathrm{HS}(\sigma)$ and $(t_B, t_C) \in \mathrm{HS}(\tau)$.*

5. A First Model with Usual Automata

We review here the model of [Rib15]. While substantially superseded by [Rib18], this model is nevertheless simpler to define, and uses structures underlying and refined in [Rib18]. It also has the advantage of being available for usual (alternating) tree automata, not requiring the specific presentation of [Rib18].

5.1. Substituted Acceptance Games. Since our solution to the composition problem discussed in §4.3 is based on the linear arrow of simple games presented in §4.6, we shall recast the set $\Sigma \vdash \mathcal{A}(M)$ of Σ-run trees of $\mathcal{A} : \Gamma$ over a $\mathbf{T}$-morphism $M : \Sigma \to \Gamma$ (defined in §4.2 above) as the set of winning P-strategies in what we called a *substituted acceptance game* in [Rib15]. We do not exactly follow the definition of [Rib15], and define substituted acceptance games as simple (tree) games rather than as graph games in the sense of Def. 2.2.1 (see §1.4 and Rem. 2.2.5). We directly give the definition for an alternating automaton $\mathcal{A}$.

Definition 5.1.1 (Substituted Acceptance Game ([Rib15])). *Consider an alternating tree automaton $\mathcal{A} = (Q, q^\imath, \partial, \Omega)$ on Γ and a morphism $M \in \mathbf{T}[\Sigma, \Gamma]$. The* substituted acceptance game *$\Sigma \vdash \mathcal{A}(M)$ is the positive simple game*

$$\big(\Sigma \times \mathcal{P}(Q \times \mathfrak{D}),\ Q \times \mathfrak{D},\ L,\ \mathcal{W}\big)$$

whose legal plays $s \in L$ are sequences of the form

$$\begin{array}{rrcl} & s & = & (\mathsf{a}_0, \gamma_0) \cdot (q_1, d_1) \cdot (\mathsf{a}_1, \gamma_1) \cdot \ldots \cdot (q_n, d_n) \\ \textit{or} & s & = & (\mathsf{a}_0, \gamma_0) \cdot (q_1, d_1) \cdot (\mathsf{a}_1, \gamma_1) \cdot \ldots \cdot (q_n, d_n) \cdot (\mathsf{a}_n, \gamma_n) \end{array}$$

where $n \geq 0$, $(q_{k+1}, d_{k+1}) \in \gamma_k$ and $\gamma_k \in \partial(q_k, M(d_1 \cdot \ldots \cdot d_k)(\mathsf{a}_0 \cdot \ldots \cdot \mathsf{a}_k))$ with $q_0 := q^\imath_{\mathcal{A}}$.

The winning plays $\chi \in \mathcal{W}$ are generated from the acceptance condition Ω in the expected way. We let $\mathcal{W} \subseteq (\mathcal{P}(Q \times \mathfrak{D}) \cdot (Q \times \mathfrak{D}))^\omega$ consist of the infinite sequences

$$\chi \quad = \quad (\mathsf{a}_0, \gamma_0) \cdot (q_1, d_1) \cdot (\mathsf{a}_1, \gamma_1) \cdot \ldots \cdot (q_n, d_n) \cdot \ldots$$

such that $(q_k)_{k \in \mathbb{N}} \in \Omega$ (where $q_0 := q^\imath$).

The input alphabet of $\Gamma \vdash \mathcal{A}$ is Γ, and we use the tree morphism $M \in \mathbf{T}[\Sigma, \Gamma]$ in a contravariant way to obtain a game with "input alphabet" Σ, that we emphasize by writing $\Sigma \vdash \mathcal{A}(M)$. Note that input letters $\mathsf{a} \in \Sigma$ are chosen by P, while directions $d \in \mathfrak{D}$ are chosen by O:

$$\begin{array}{ccccccccccc} \mathsf{P} & & \mathsf{O} & & \mathsf{P} & & \mathsf{O} & & & & \mathsf{P} \\ (\mathsf{a}_0, \gamma_0) & \cdot & (q_0, d_0) & \cdot & (\mathsf{a}_1, \gamma_1) & \cdot & (q_1, d_1) & \cdot & \cdots & \cdot & (\mathsf{a}_{n+1}, \gamma_{n+1}) & \cdot & \cdots \end{array}$$

Write $\Sigma \vdash \sigma \Vdash \mathcal{A}(M)$ if σ is a winning P-strategy on $\Sigma \vdash \mathcal{A}(M)$, and $\Sigma \Vdash \mathcal{A}(M)$ if $\Sigma \vdash \sigma \Vdash \mathcal{A}(M)$ for some σ.

Remark 5.1.2 (Correspondence with usual Acceptance Games). *Substituted acceptance games generalize usual acceptance games (see Ex. 2.2.4 and Rem. 2.2.5). Given a tree $T : \mathfrak{D}^* \to \Sigma$, writing $\dot{T}$ for the corresponding $\mathbf{T}$ morphism $\mathbf{1} \to_{\mathbf{T}} \Sigma$, the set of P-moves of the substituted acceptance game $\mathbf{1} \vdash \mathcal{A}(\dot{T})$ is $\mathbf{1} \times \mathcal{P}(Q_{\mathcal{A}} \times \mathfrak{D}) \simeq \mathcal{P}(Q_{\mathcal{A}} \times \mathfrak{D})$, so that the positive simple games $\mathbf{1} \vdash \mathcal{A}(\dot{T})$ and $\mathcal{G}(\mathcal{A}, T)$ are isomorphic.*

5.2. Linear Synchronous Arrow Games. We now present the notion of *linear synchronous arrow games*, which constitutes the morphisms of [Rib15]. Synchronous arrow games are a restriction of the linear arrow of simple games between substituted acceptance games, in which P has to play the same input letters a and the same tree directions d as proposed by O. Consider substituted acceptance games $\Sigma \vdash \mathcal{A}(M)$ and $\Sigma \vdash \mathcal{B}(N)$ with

$$\begin{array}{rrcl} & \mathcal{A} & = & (Q_{\mathcal{A}}, q^{\imath}_{\mathcal{A}}, \partial_{\mathcal{A}}, \Omega_{\mathcal{A}}) \\ \text{and} & \mathcal{B} & = & (Q_{\mathcal{B}}, q^{\imath}_{\mathcal{B}}, \partial_{\mathcal{B}}, \Omega_{\mathcal{B}}) \end{array}$$

We first define a *synchronous strategy*, notation

$$\Sigma \vdash \sigma : \mathcal{A}(M) \multimap_{\circledast} \mathcal{B}(N)$$

as a strategy

$$\sigma \quad : \quad (\Sigma \vdash \mathcal{A}(M)) \quad \longrightarrow_{\mathbf{SG}} \quad (\Sigma \vdash \mathcal{B}(N))$$

satisfying some specific constraints. We will then informally speak of the *synchronous game*

$$\Sigma \vdash \mathcal{A}(M) \multimap_{\circledast} \mathcal{B}(N)$$

to mean

$$(\Sigma \vdash \mathcal{A}(M)) \quad \longrightarrow_{\mathbf{SG}} \quad (\Sigma \vdash \mathcal{B}(N))$$

restricted to synchronous strategies. In any case, we drop the Σ in the above notations when clear from the context.

Intuitively, synchronous strategies are strategies whose plays are synchronous, in the sense that $\mathcal{A}$ and $\mathcal{B}$ are evaluated along the same path in $\mathfrak{D}^\omega$, while M and N read the same input letters from Σ. The synchronous plays of $\mathcal{A}(M) \multimap_{\mathbf{SG}} \mathcal{B}(N)$ are defined using the following notion of *trace*. Let

$$\mathrm{Tr}_\Sigma \quad := \quad (\Sigma \cdot \mathfrak{D})^* + (\Sigma \cdot \mathfrak{D})^* \cdot \Sigma$$

and define the *trace function* $\mathrm{tr}_{\mathcal{A}(M)} : L_{\mathcal{A}(M)} \longrightarrow \mathrm{Tr}_\Sigma$ inductively as follows

$$\begin{array}{rcl} \mathrm{tr}_{\mathcal{A}(M)}(\varepsilon) & := & \varepsilon \\ \mathrm{tr}_{\mathcal{A}(M)}(s \cdot (\mathsf{a}, \gamma)) & := & \mathrm{tr}_{\mathcal{A}(M)}(s) \cdot \mathsf{a} \\ \mathrm{tr}_{\mathcal{A}(M)}(s \cdot (q, d)) & := & \mathrm{tr}_{\mathcal{A}(M)}(s) \cdot d \end{array}$$

We let the *trace* of a play $s \in L_{\mathcal{A}(M)}$ be the sequence $\mathrm{tr}_{\mathcal{A}(M)}(s)$. The trace function $\mathrm{tr}_{\mathcal{B}(N)} : L_{\mathcal{B}(N)} \longrightarrow \mathrm{Tr}_\Sigma$ is defined similarly. Note that both $\mathrm{tr}_{\mathcal{A}(M)}$ and $\mathrm{tr}_{\mathcal{B}(N)}$ have the same codomain Tr_Σ, which only depends on the input alphabet of M and N.

Definition 5.2.1 (Synchronous (P-)Play)**.** *A legal play $s \in L_{\mathcal{A}(M) \multimap_{\mathbf{SG}} \mathcal{B}(N)}$ is* synchronous *if*

$$\mathrm{tr}_{\mathcal{A}(M)}(s_{\restriction \mathcal{A}(M)}) \quad = \quad \mathrm{tr}_{\mathcal{B}(N)}(s_{\restriction \mathcal{B}(N)})$$

Note that trace functions are length-preserving, so that the trace of a play s always has the same length as s. Hence if s is a synchronous play in $\mathcal{A}(M) \multimap_{\mathbf{SG}} \mathcal{B}(N)$, then $s_{\restriction A(M)}$ and $s_{\restriction B(N)}$ have the same length, so that s is even length. It follows that the synchronous plays of $\mathcal{A}(M) \multimap_{\mathbf{SG}} \mathcal{B}(N)$ must be P-plays.

A typical synchronous play in $\mathcal{A}(M) \multimap_{\mathbf{SG}} \mathcal{B}(N)$ is depicted in Fig. 5.1. Note that synchronous plays must have the same zig-zag shape as the copy-cat plays (see Fig. 4.1), and that O actually chooses both the input letters $\mathsf{a} \in \Sigma$ and the tree directions $d \in \mathfrak{D}$. This follows from the

Σ	$\mathcal{A}(M)$	$\mathrel{-\!\!\circledast}$	$\mathcal{B}(N)$	
	$(\varepsilon, \varepsilon, q^{\imath}_{\mathcal{A}})$		$(\varepsilon, \varepsilon, q^{\imath}_{\mathcal{B}})$	
	$\vdots$		$\vdots$	
	$(p, \bar{\mathsf{a}}, q_{\mathcal{A}})$		$(p, \bar{\mathsf{a}}, q_{\mathcal{B}})$	
O	$(\mathsf{a}, \gamma_{\mathcal{A}})$			
			$(\mathsf{a}, \gamma_{\mathcal{B}})$	P
			$(q'_{\mathcal{B}}, d)$	O
P	$(q'_{\mathcal{A}}, d)$			
	$(p \cdot d,\ \bar{\mathsf{a}} \cdot \mathsf{a},\ q'_{\mathcal{A}})$		$(p \cdot d,\ \bar{\mathsf{a}} \cdot \mathsf{a},\ q'_{\mathcal{B}})$	
	$\vdots$		$\vdots$	

Figure 5.1.: A play of a synchronous strategy

fact that in the game $\mathcal{A}(M) \multimap_{\mathbf{SG}} \mathcal{B}(N)$, O must begin in the component $\mathcal{A}(M)$, choosing in particular some $\mathsf{a} \in \Sigma$. Then, by synchronicity, P must switch to component $\mathcal{B}(N)$ and play a move containing the same $\mathsf{a} \in \Sigma$. Since O cannot switch component, its next move must be in component $\mathcal{B}(N)$, and so in particular contain some $d \in \mathfrak{D}$. But then, again by synchronicity, P must switch to component $\mathcal{A}(M)$ and play a move containing the same $d \in \mathfrak{D}$.

We are now going to formally define what is synchronous strategy. Note the following pullback in **Set**, where $L^{\mathsf{even}}_{\mathcal{A}(M) \mathrel{-\!\!\circledast} \mathcal{B}(N)}$ denotes the set of synchronous plays of $\mathcal{A}(M) \multimap_{\mathbf{SG}} \mathcal{B}(N)$:

$$\begin{array}{ccc} L^{\mathsf{even}}_{\mathcal{A}(M) \mathrel{-\!\!\circledast} \mathcal{B}(N)} & \longrightarrow & L_{\mathcal{B}(N)} \\ \big\downarrow & \lrcorner & \big\downarrow \mathrm{tr} \\ L_{\mathcal{A}(M)} & \xrightarrow[\mathrm{tr}]{} & \mathrm{Tr}_{\Sigma} \end{array} \tag{5.1}$$

Definition 5.2.2 (Synchronous Strategies)**.** *A strategy $\sigma : (\Sigma \vdash \mathcal{A}(M)) \multimap_{\mathbf{SG}} (\Sigma \vdash \mathcal{B}(N))$ is* synchronous *if the following commutes in* **Set***:*

$$\begin{array}{ccc} \mathrm{HS}(\sigma) & \longrightarrow & L_{\mathcal{B}(N)} \\ \big\downarrow & & \big\downarrow \mathrm{tr} \\ L_{\mathcal{A}(M)} & \xrightarrow[\mathrm{tr}]{} & \mathrm{Tr}_{\Sigma} \end{array}$$

It readily follows from §4.7 that the identity copy-cat strategy $\mathrm{id}_{\mathcal{A}(M)} : \mathcal{A}(M) \multimap_{\mathbf{SG}} \mathcal{A}(M)$ is synchronous, and that synchronous strategies are preserved by composition. We can thus define categories of substituted acceptance games and synchronous strategies.

Definition 5.2.3 (The Categories $\mathbf{SAG}_{(-)}$)**.** *For each alphabet Σ, the category $\mathbf{SAG}_{\Sigma}$ is defined as follows:*

- *the objects of $\mathbf{SAG}_{\Sigma}$ are games $\Sigma \vdash \mathcal{A}(M)$,*
- *the morphisms of $\mathbf{SAG}_{\Sigma}$ are synchronous strategies $\Sigma \vdash \sigma : \mathcal{A}(M) \mathrel{-\!\!\circledast} \mathcal{B}(N)$.*

The faithfulness of HS is actually trivial for synchronous strategies (see Lem. 6.1.2).

The game $\Sigma \vdash \mathcal{A}(M) \mathrel{-\!\!\circledast} \mathcal{B}(N)$ can be equipped with the winning condition mentioned in §4.6, using the winning condition of Def. 2.2.1 on the component games $\mathcal{A}(M)$ and $\mathcal{B}(N)$. Hence, given an infinite play χ in $\mathcal{A}(M) \multimap \mathcal{B}(N)$ whose projections on $\mathcal{A}(M)$ and $\mathcal{B}(N)$ are both infinite, we say that χ is *winning for* P if its projection on $\mathcal{B}(N)$ is winning for P whenever its projection on $\mathcal{A}(M)$ is winning for P. We write

$$\Sigma \vdash \sigma \Vdash \mathcal{A}(M) \multimap \mathcal{B}(N)$$

to mean that the synchronous strategy $\sigma : \mathcal{A}(M) \multimap \mathcal{B}(N)$ is winning w.r.t. the winning condition described above. Preservation of winning by composition of synchronous strategies[1], which is simpler than the general case of simple games, follows from the same property for zig-zag strategies, which are discussed in §6.2.

Definition 5.2.4 (The Categories $\mathbf{SAG}^{\mathrm{W}}_{(-)}$)**.** *For each alphabet Σ, the category $\mathbf{SAG}^{\mathrm{W}}_{\Sigma}$ is defined as follows:*

- *the objects of $\mathbf{SAG}^{\mathrm{W}}_{\Sigma}$ are games $\Sigma \vdash \mathcal{A}(M)$,*
- *the morphisms of $\mathbf{SAG}^{\mathrm{W}}_{\Sigma}$ are synchronous winning strategies $\Sigma \vdash \sigma \Vdash \mathcal{A}(M) \multimap \mathcal{B}(N)$.*

Remark 5.2.5 (Correspondence with usual Acceptance Games)**.** *Linear asynchronous arrow games also generalize usual acceptance games. The idea is that given a tree $T : \mathfrak{D}^* \to \Sigma$, the* P*-strategies in the substituted acceptance games $\mathbf{1} \vdash \mathcal{A}(\dot{T})$ are in bijection with the* P*-strategies in the linear synchronous arrow game $\mathbf{1} \vdash \mathbf{I} \multimap \mathcal{A}(\dot{T})$ (where $\mathbf{I}$ is a unit automaton similar to that of Ex. 7.0.2.(i)).*

5.3. Substitution and Fibred Structure. We briefly discuss here the indexed structure of the categories $\mathbf{SAG}_{(-)}$ and $\mathbf{SAG}^{\mathrm{W}}_{(-)}$, as it underlies that of [Rib18]. According to the setting sketched in §4.2, we are looking for (contravariant) *substitution functors*

$$(-)^\star \quad : \quad \mathbf{T}^{\mathrm{op}} \longrightarrow \mathbf{Cat}$$

taking an alphabet Σ to a category $\mathbf{SAG}^{(\mathrm{W})}_{\Sigma}$ and a $\mathbf{T}$-morphism $L : \Sigma \to \Gamma$ to a functor

$$L^\star \quad : \quad \mathbf{SAG}^{(\mathrm{W})}_{\Gamma} \quad \longrightarrow \quad \mathbf{SAG}^{(\mathrm{W})}_{\Sigma}$$

Fix a $\mathbf{T}$-morphism $L : \Sigma \to \Gamma$. On objects, $L^\star$ acts as suggested in §4.2:

$$L^\star(\Gamma \vdash \mathcal{A}(M)) \quad := \quad \Sigma \vdash \mathcal{A}(M \circ L)$$

The action of $L^\star$ on strategies is not difficult but requires more work. The idea is that $L : \Sigma \to_{\mathbf{T}} \Gamma$ extends to a function $\mathrm{Tr}(L) : \mathrm{Tr}_\Sigma \to \mathrm{Tr}_\Gamma$, from which the action of $L^\star$ on objects can be recovered as the following pullback in $\mathbf{Set}$:

$$\begin{array}{ccc} L_{\mathcal{A}(M \circ L)} & \longrightarrow & L_{\mathcal{A}(M)} \\ {\scriptstyle \mathrm{tr}}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{tr}} \\ \mathrm{Tr}_\Sigma & \xrightarrow{\mathrm{Tr}(L)} & \mathrm{Tr}_\Gamma \end{array}$$

In order to define the action of $L^\star$ on strategies, one just has to note that the above pullback extends to synchronous (P-)plays in arrow games:

$$\begin{array}{ccc} L^{\mathsf{even}}_{\mathcal{A}(M \circ L) \multimap \mathcal{B}(N \circ L)} & \longrightarrow & L^{\mathsf{even}}_{\mathcal{A}(M) \multimap \mathcal{B}(N)} \\ {\scriptstyle \mathrm{tr}^{\multimap}}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{tr}^{\multimap}} \\ \mathrm{Tr}_\Sigma & \xrightarrow{\mathrm{Tr}(L)} & \mathrm{Tr}_\Gamma \end{array}$$

[1] Recall that winning strategies are assumed to be total in Def. A.0.3.

(where the trace functions $\mathrm{tr}^{-\circledast}$ are obtained from (5.1) in the obvious way). It is then tedious but easy to define a strategy $L^\star(\sigma)$ so as to get

$$\begin{array}{ccc} \mathrm{HS}(L^\star(\sigma)) & \longrightarrow & \mathrm{HS}(\sigma) \\ {\scriptstyle \mathrm{tr}^{-\circledast}}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{tr}^{-\circledast}} \\ \mathrm{Tr}_\Sigma & \xrightarrow{\mathrm{Tr}(L)} & \mathrm{Tr}_\Gamma \end{array}$$

For each $\mathbf{T}$-morphism $L : \Sigma \to \Gamma$ we thus obtain functors

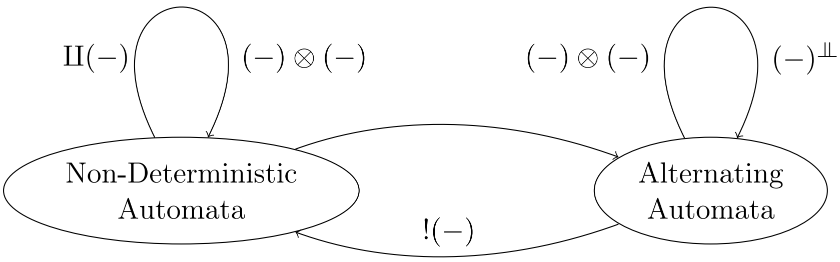
$$L^\star \;:\; \mathbf{SAG}_\Gamma \;\longrightarrow\; \mathbf{SAG}_\Sigma \qquad \text{and} \qquad L^\star \;:\; \mathbf{SAG}^{\mathrm{W}}_\Gamma \;\longrightarrow\; \mathbf{SAG}^{\mathrm{W}}_\Sigma$$

The maps $(-)^\star$ are actually themselves (contravariantly) functorial, in the sense that

$$(\mathrm{Id}_\Sigma)^\star \;=\; \mathrm{Id}_{\mathbf{SAG}^{(\mathrm{W})}_\Sigma} \qquad \text{and} \qquad (L \circ K)^\star \;=\; K^\star \circ L^\star$$

We refer to [Rib15] and its full version for details.

5.4. Some Further Structure. The categories $\mathbf{SAG}_{(-)}$ and $\mathbf{SAG}^{\mathrm{W}}_{(-)}$ are actually equipped with all the structure mentioned in the picture (1.5), which in the setting of usual tree automata could be drawn as:



The (symmetric) monoidal product $\otimes$ is the direct product alluded to in §4.4 and §4.5, and $!(-)$ is given by a usual powerset construction for the *Simulation* Theorem [EJ91, MS95]. As for existential quantification, in order to get the expected categorical properties (mentioned in §4.2) we devised a variant (written $\amalg$) of the usual projection operation, with which at each step, the next state is labeled with the hidden (projected) letter. Complementation of alternating automata $(-)^{\perp\!\!\!\perp}$ is the usual operation (linear in the number of states). All this structure is detailed in [Rib15] and its full version, and we do note repeat it here since it involves some easy but inconvenient technicalities w.r.t. totality of automata.

Our main motivation to switch to the setting of [Rib18] is to obtain a clean (monoidal) *closed* structure on automata. We explain in §5.5 below the functoriality problem we had in the particular case of the usual linear complementation operation $(-)^{\perp\!\!\!\perp}$ on alternating automata.

5.5. On the Functoriality of the Usual Linear Negation of Alternating Automata. We explain here why it seems not obvious to turn the usual linear complementation $(-)^{\perp\!\!\!\perp}$ of alternating automata into a (contravariant) functor. The difficulty resides in the preservation of composition. Recall from §1.4 that we see the transition function

$$\partial_{\mathcal{A}} \;:\; Q_{\mathcal{A}} \times \Sigma \;\longrightarrow\; \mathcal{P}(\mathcal{P}(Q_{\mathcal{A}} \times \mathfrak{D}))$$

of an alternating automaton $\mathcal{A} : \Sigma$ as taking $(q, \mathtt{a}) \in Q_\mathcal{A} \times \Sigma$ to a disjunctive normal form

$$\bigvee_{\gamma \in \partial_\mathcal{A}(q,\mathtt{a})} \ \bigwedge_{(q',d) \in \gamma} (q', d)$$

so that the transition function of $\mathcal{A}^{\perp\!\!\!\perp}$ intuitively has to take $(q, \mathtt{a})$ to a disjunctive normal form representing the dual of $\partial_\mathcal{A}(q, \mathtt{a})$. Following [Wal02], we thus let $\mathcal{A}^{\perp\!\!\!\perp}$ have the same states as $\mathcal{A}$ and $\partial_{\mathcal{A}^{\perp\!\!\!\perp}}(q, \mathtt{a})$ be the set of all $\gamma^{\perp\!\!\!\perp} \subseteq \mathcal{P}(Q_\mathcal{A} \times \mathfrak{D})$ such that $\gamma^{\perp\!\!\!\perp} \cap \gamma \neq \emptyset$ for all $\gamma \in \partial_\mathcal{A}(q, \mathtt{a})$.

Consider a total (winning) P-strategy $\sigma : \mathcal{A} \multimap\!\!\circledast\ \mathcal{B}$ playing as in Fig. 5.1, and let us see how to directly define a total (winning) strategy $\sigma^{\perp\!\!\!\perp} : \mathcal{B}^{\perp\!\!\!\perp} \multimap\!\!\circledast\ \mathcal{A}^{\perp\!\!\!\perp}$. The plays of $\sigma^{\perp\!\!\!\perp}$ should have the following shape:

Σ	$\mathcal{B}^{\perp\!\!\!\perp}$	$\multimap\!\!\circledast$	$\mathcal{A}^{\perp\!\!\!\perp}$	
	$(\varepsilon, q^\imath_\mathcal{B})$		$(\varepsilon, q^\imath_\mathcal{A})$	
	$\vdots$		$\vdots$	
	$(p, q_\mathcal{B})$		$(p, q_\mathcal{A})$	
O	$(\mathtt{a}, \gamma_{\mathcal{B}^{\perp\!\!\!\perp}})$			with $\gamma_{\mathcal{B}^{\perp\!\!\!\perp}} \in \partial_{\mathcal{B}^{\perp\!\!\!\perp}}(q_\mathcal{B}, \mathtt{a})$
P			$(\mathtt{a}, \gamma_{\mathcal{A}^{\perp\!\!\!\perp}})$	with $\gamma_{\mathcal{A}^{\perp\!\!\!\perp}} \in \partial_{\mathcal{A}^{\perp\!\!\!\perp}}(q_\mathcal{A}, \mathtt{a})$
O			$(q'_\mathcal{A}, d)$	with $(q'_\mathcal{A}, d) \in \gamma_{\mathcal{A}^{\perp\!\!\!\perp}}$
P	$(q'_\mathcal{B}, d)$			with $(q'_\mathcal{B}, d) \in \gamma_{\mathcal{B}^{\perp\!\!\!\perp}}$
	$(p.d,\ q'_\mathcal{B})$		$(p.d,\ q'_\mathcal{A})$	
	$\vdots$		$\vdots$	

Let us see how to directly define $\sigma^{\perp\!\!\!\perp}$ from σ. Assume we are in position $((p, q_\mathcal{B}), (p, q_\mathcal{A}))$ as above, and fix $\mathtt{a}$ and $\gamma_{\mathcal{B}^{\perp\!\!\!\perp}}$. We have to choose some $\gamma_{\mathcal{A}^{\perp\!\!\!\perp}}$ such that $\gamma_{\mathcal{A}^{\perp\!\!\!\perp}} \cap \gamma_\mathcal{A} \neq \emptyset$ for all $\gamma_\mathcal{A} \in \partial_\mathcal{A}(q_\mathcal{A}, \mathtt{a})$, and moreover, for each $(q'_\mathcal{A}, d) \in \gamma_{\mathcal{A}^{\perp\!\!\!\perp}}$, we must choose some $(q'_\mathcal{B}, d)$ from $\gamma_{\mathcal{B}^{\perp\!\!\!\perp}}$. The only canonical way to do this seems to use the fact that from position $((p, q_\mathcal{A}), (p, q_\mathcal{B}))$, the strategy σ induces functions

$$\begin{array}{lrcccc} & f & : & \gamma_\mathcal{A} & \longmapsto & \gamma_\mathcal{B} \\ \text{and} & F & : & (\gamma_\mathcal{A}\ ,\ (q'_\mathcal{B}, d) \in f(\gamma_\mathcal{A})) & \longmapsto & (q'_\mathcal{A}, d) \in \gamma_\mathcal{A} \end{array}$$

as in:

Σ	$\mathcal{A}$	$\overset{\sigma}{\multimap\!\!\circledast}$	$\mathcal{B}$	
	$(p, q_\mathcal{A})$		$(p, q_\mathcal{B})$	
O	$(\mathtt{a}, \gamma_\mathcal{A})$			
P			$(\mathtt{a}, \gamma_\mathcal{B})$	$f(\gamma_\mathcal{A}) = \gamma_\mathcal{B}$
O			$(q'_\mathcal{B}, d)$	
P	$(q'_\mathcal{A}, d)$			$F(\gamma_\mathcal{A},\ (q'_\mathcal{B}, d)) = (q'_\mathcal{A}, d)$

Then we can let

$$\gamma_{\mathcal{A}^{\perp\!\!\!\perp}} \quad := \quad \{F(\gamma_\mathcal{A}, (q'_\mathcal{B}, d)) \ \mid\ \gamma_\mathcal{A} \in \partial_\mathcal{A}(q_\mathcal{A}, \mathtt{a}) \quad \text{and} \quad (q'_\mathcal{B}, d) \in \gamma_{\mathcal{B}^{\perp\!\!\!\perp}} \cap f(\gamma_\mathcal{A})\}$$

Moreover, for each $(q'_\mathcal{A}, d) \in \gamma_{\mathcal{A}^{\perp\!\!\!\perp}}$, there are some $\gamma_\mathcal{A}$ and some $(q'_\mathcal{B}, d) \in \gamma_{\mathcal{B}^{\perp\!\!\!\perp}} \cap f(\gamma_\mathcal{A})$ such that $(q'_\mathcal{A}, d) = F(\gamma_\mathcal{A}, (q'_\mathcal{B}, d))$. The difficulty here is that σ may play *the same* $(q'_\mathcal{A}, d)$ from one $\gamma_\mathcal{A}$ but from *distinct* $(q'_\mathcal{B}, d), (q''_\mathcal{B}, d) \in \gamma_{\mathcal{B}^{\perp\!\!\!\perp}} \cap f(\gamma_\mathcal{A})$, and it is not clear how to choose one. A possibility would be to impose a linear order on states, and to always take the least available state. But then it is not clear how to preserve composition, because σ may be precomposed with a strategy $\tau : \mathcal{B} \multimap\!\!\circledast\ \mathcal{C}$ where $\mathcal{C}$ is defined as $\mathcal{B}$ but with a different order on states, and τ plays as the identity, but does not preserve the order of states.

6. Zig-Zag Games

The purpose of this chapter is to give some conceptual and technical justification to the model of [Rib18] (to be presented in an elementary way in Chap. 7). Besides with its improvement over [Rib15] (Chap. 5) in terms of categorical structure, the model of [Rib18] comes from a simplification of the underlying game model, namely to total *zig-zag* strategies. A consequence of this simplification is that strategies now admit a very simple functional presentation. This gives a further description of their basic categorical structure, as well as the basis of the Dialectica-like approach of [PR18b, PR19].

This chapter is mostly technical and unpublished. Sections 6.1 and 6.2 present the category $\mathbf{DZ}$ (resp. $\mathbf{DZ}^{\mathrm{W}}$) of full positive games (in the sense of Ex. A.0.2) and total zig-zag strategies (resp. winning total zig-zag strategies). We show in §6.3 how the category $\mathbf{DZ}$ can be reconstructed by combining, starting from the topos of trees $\mathscr{S}$ (see e.g. [BMSS12]), known recipes for building models of linear logic [HS03] with a specific distributive law. This in particular gives a conceptual approach to the monoidal structure of $\mathbf{DZ}$. We then outline in §6.4 an indexed structure on top of $\mathbf{DZ}$ similar to that of $\mathbf{SAG}_{(-)}$ (Chap. 5), but this time built from a variant of *simple fibrations* (see e.g. [Jac01]) based on *comonoid indexing* [HS99, HS03]. We thus "mechanically" obtain indexed categories with existential and universal quantifications, improving on [Rib15]. Finally, we give in §6.5 a notion of *finite-state* total zig-zag strategy, essentially obtained by instantiating the framework of §2.3 to the functional representation of total zig-zag strategies.

6.1. Zig-Zag Strategies

Consider substituted acceptance games $\Sigma \vdash \mathcal{A}(M), \mathcal{B}(N)$ as in §5.1. Recall that the synchronicity constraint of §5.2 imposes a legal P-play s in $\mathcal{A}(M) \multimap\!\!\circledast\, \mathcal{B}(N)$ to satisfy

$$\mathrm{tr}_{\mathcal{A}(M)}(s_{\restriction \mathcal{A}(M)}) \quad = \quad \mathrm{tr}_{\mathcal{B}(N)}(s_{\restriction \mathcal{B}(N)})$$

Since the functions $\mathrm{tr}_{\mathcal{A}(M)}$ and $\mathrm{tr}_{\mathcal{B}(N)}$ are length-preserving, this imposes in particular $s_{\restriction \mathcal{A}(M)}$ and $s_{\restriction \mathcal{B}(N)}$ to have the same length.

On the other hand, given simple games A and B of the same polarity, and a play s in $A \multimap_{\mathbf{SG}} B$, if

$$\mathrm{length}(s_{\restriction A}) \quad = \quad \mathrm{length}(s_{\restriction B}) \tag{6.1}$$

then in s, each P-move must switch component w.r.t. the previous O-move. Let us discuss the case where (say) $A = (U, X)$ and $B = (V, Y)$ are full positive games (see Ex. A.0.2). Recall that O begins in $A \multimap_{\mathbf{SG}} B$ and must play in component A since A and B are positive. In order to maintain (6.1), P must then switch to component B. After the P-move in B, the switching condition imposes O to stay in B, and then P has to switch to A, again to maintain (6.1). It follows that s must have the *zig-zag* shape depicted in Fig. 6.1.

This leads to the following notion of zig-zag strategies in the setting of simple games. We restrict to the case of full positive games (see Ex. A.0.2).

	A	$\multimap$	B	
	$\vdots$		$\vdots$	
O	u			
			v	P
			y	O
P	x			
	$\vdots$		$\vdots$	

Figure 6.1.: A typical zig-zag play with $A = (U, X)$ and $B = (V, Y)$ full positive games

Definition 6.1.1 (Zig-Zag Plays and Strategies)**.** *Given full positive games A and B, a play s in $A \multimap_{\mathbf{SG}} B$ is a* zig-zag play *if*

$$\text{length}(s_{\restriction A}) \quad = \quad \text{length}(s_{\restriction B})$$

A P-strategy $\sigma : A \multimap_{\mathbf{SG}} B$ is a zig-zag *strategy if all its plays are zig-zag plays.*

We write $A \multimap_{\mathbf{DZ}} B$ for the game obtained by restricting $A \multimap_{\mathbf{SG}} B$ to (prefixes of) its legal zig-zag plays (so the P-strategies on $A \multimap_{\mathbf{DZ}} B$ are exactly the zig-zag P-strategies on $A \multimap_{\mathbf{SG}} B$).

Consider now games with winning A and B. Note that if $\sigma : A \multimap_{\mathbf{DZ}} B$ is total, then for every $\chi \in ((A_{\mathsf{P}} + B_{\mathsf{O}}) \cdot (A_{\mathsf{O}} + B_{\mathsf{P}}))^\omega$, if χ has infinitely many finite prefixes in σ, then $\chi_{\restriction A}$ and $\chi_{\restriction B}$ are both infinite. We therefore let $\mathcal{W}_{A \multimap B} \subseteq ((A_{\mathsf{P}} + B_{\mathsf{O}}) \cdot (A_{\mathsf{O}} + B_{\mathsf{P}}))^\omega$ be the set of infinite sequences χ such that $(\chi_{\restriction A} \in \mathcal{W}_A \;\Rightarrow\; \chi_{\restriction B} \in \mathcal{W}_B)$.

We now briefly discuss the composition of zig-zag strategies, following §5.2 and §4.7. First, note that zig-zag strategies $\sigma : A \multimap_{\mathbf{DZ}} B$ could have been defined similarly as the synchronous ones (Def. 5.2.2) as being those strategies $\sigma : A \multimap_{\mathbf{SG}} B$ such that

$$\begin{array}{ccc} \text{HS}(\sigma) & \longrightarrow & L_B \\ \downarrow & & \downarrow \scriptstyle\text{length} \\ L_A & \xrightarrow[\text{length}]{} & \mathbb{N} \end{array}$$

The faithfulness of $\text{HS} : \mathbf{SG} \to \mathbf{Rel}$ is actually trivial for zig-zag strategies, because HS is already injective on zig-zag *plays*: given $(t, t') \in L_A \times L_B$, there is at most one *zig-zag* play s such that $\text{HS}(s) = (t, t')$.

Lemma 6.1.2. *(i) Given* zig-zag *plays s, t in $A \multimap_{\mathbf{SG}} B$, if $\text{HS}(s) = \text{HS}(t)$ then $s = t$.*

(ii) The map HS is injective on zig-zag *strategies: $\text{HS}(\sigma) = \text{HS}(\tau)$ implies $\sigma = \tau$.*

Similarly as for synchronous strategies in §5.2, it then follows from §4.7 that copy-cat strategies are zig-zag and that zig-zag strategies are preserved by composition.

6.2. The Categories DZ and DZ$^{\text{W}}$

It follows from §6.1 that we could define a category of full positive games and zig-zag strategies. We shall actually additionally require zig-zag strategies to be *total*.

Proposition 6.2.1. *Full positive games (with winning) and (winning) total zig-zag strategies form a category $\mathbf{DZ}^{(\text{W})}$.*

	$A \xrightarrow{(f,F)}$	B	
	(u_0, x_0)	(v_0, y_0)	
	$\vdots$	$\vdots$	
	(u_n, x_n)	(v_n, y_n)	
O	u_{n+1}		
P		v_{n+1}	$v_{n+1} = f_{n+1}(u_0 \cdot \ldots \cdot u_n \cdot u_{n+1}\,,\, y_0 \cdot \ldots \cdot y_n)$
O		y_{n+1}	
P	x_{n+1}		$x_{n+1} = F_{n+1}(u_0 \cdot \ldots \cdot u_n \cdot u_{n+1}\,,\, y_0 \cdot \ldots \cdot y_n \cdot y_{n+1})$
	$\vdots$	$\vdots$	

Figure 6.2.: Representation of total zig-zag strategies according to Prop. 6.2.2 (where $A = (U, X)$ and $B = (V, Y)$).

It must be noted here that simple games and total strategies do *not* form a category, unless some winning conditions are assumed (see e.g. [Abr97]). The special case of **DZ**, to be detailed below, can be thought about as relying in particular on the following very simple functional representation of strategies (which was taken as a definition in [Rib18]).

Proposition 6.2.2. *Consider full positive games $A = (U, X)$ and $B = (V, Y)$.* Total *zig-zag strategies $\sigma : A \multimap_{\mathbf{DZ}} B$ are in bijection with pairs of functions (f, F) where*

$$\begin{array}{lcll} f & : & \bigcup_{n>0} \left(U^n \times Y^{n-1}\right) & \longrightarrow & V \\ F & : & \bigcup_{n>0} \left(U^n \times Y^n\right) & \longrightarrow & X \end{array} \tag{6.2}$$

Given pairs of maps (f, F) as in (6.2), for each $n > 0$, we write f_n and F_n for the induced maps

$$f_n \quad : \quad U^n \times Y^{n-1} \quad \longrightarrow \quad V^n \qquad \text{and} \qquad F_n \quad : \quad U^n \times Y^n \quad \longrightarrow \quad X^n$$

The representation of strategies by Prop. 6.2.2 can be pictured as in Fig. 6.2.

Note that for each full positive game $A = (U, X)$, there is a bijection

$$\beta \;=\; \langle \beta_U, \beta_X \rangle \quad : \quad L_A^{\mathsf{even}} \quad \longrightarrow \quad \cup_{n \in \mathbb{N}} (U^n \times X^n)$$

defined as $\beta(\varepsilon) := (\varepsilon, \varepsilon)$ and $\beta(s.u.x) = (\beta_U(s).u, \beta_X(s).x)$. In the following, we often write $((\overline{u}, \overline{x}), (\overline{v}, \overline{y})) \in \mathrm{HS}(\sigma)$ for $(\beta^{-1}(\overline{u}, \overline{x}), \beta^{-1}(\overline{v}, \overline{y})) \in \mathrm{HS}(\sigma)$.

Proof of Prop. 6.2.2. Fix $A = (U, X)$ and $B = (V, Y)$ and consider a total zig-zag strategy $\sigma : A \multimap_{\mathbf{DZ}} B$. By induction on $n \in \mathbb{N}$, it is easy to see that for all $(\overline{u}, \overline{y}) \in U^n \times Y^n$, there is a unique $(s, t) \in \mathrm{HS}(\sigma)$ such that $\overline{u} = \beta_U(s)$ and $\overline{y} = \beta_Y(t)$. The property vacuously holds for $n = 0$. Assuming it for n, given $(\overline{u}.u, \overline{y}.y) \in U^{n+1} \times Y^{n+1}$, by induction hypothesis, there is a unique $(s, t) \in \mathrm{HS}(\sigma)$ such that $\overline{u} = \beta_U(s)$ and $\overline{y} = \beta_Y(t)$. Now, since σ is total and zig-zag, there is a unique $v \in V$ such that $(s.u, t.v) \in \mathrm{HS}(\sigma)$. Similarly, there is a unique $x \in X$ such that $(s.u.x, t.v.y) \in \mathrm{HS}(\sigma)$, and the property follows. Furthermore, since $\overline{u}.u$ and $\overline{y}$ uniquely determine $\overline{v} = \beta_V(t)$ and v, and since $\overline{u}.u$ and $\overline{y}.y$ uniquely determine $\overline{x} = \beta_X(s)$ and x, we obtain a pair of functions (f, F) as in (6.2) defined as

$$f(\overline{u}.u,\, \overline{y}) \quad := \quad v \qquad \text{and} \qquad F(\overline{y}.y,\, \overline{u}.u) \quad := \quad x$$

Conversely, each pair (f, F) as in (6.2) uniquely determines a total zig-zag strategy σ, with, for all $\overline{u}.u \in U^{n+1}$, and all $\overline{y} \in Y^n$,

$$((\overline{u}, \overline{x}).u \ , \ (\overline{v}, \overline{y}).v) \ \in \ \mathrm{HS}(\sigma)$$

where $\overline{v}.v = f_{n+1}(\overline{u}.u, \overline{y})$ and $\overline{x} = F_n(\overline{u}, \overline{y})$; and moreover for all y,

$$((\overline{u}, \overline{x}).u.x \ , \ (\overline{v}, \overline{y}).v.y) \ \in \ \mathrm{HS}(\sigma)$$

where $x = F(\overline{u}.u, \overline{y}.y)$. □

The representation of strategies by pairs of maps (f, F) as given by Prop. 6.2.2 provides an easy way to show that total zig-zag strategies are preserved by composition. Fix total zig-zag strategies $\sigma : A \multimap_{\mathbf{DZ}} B$ and $\tau : B \multimap_{\mathbf{DZ}} C$. Thanks to Prop. 6.2.2, given pairs of maps (f, F) and (g, G) representing resp. σ and τ, this amounts to provide a pair (h, H) representing $\tau \circ \sigma$. Write $A = (U, X)$, $B = (V, Y)$ and $C = (W, Z)$. The relational composite $\mathrm{HS}(\tau) \circ \mathrm{HS}(\sigma)$ is such that $((\overline{u}, \overline{x}), (\overline{w}, \overline{z})) \in \mathrm{HS}(\tau) \circ \mathrm{HS}(\sigma)$ if and only if there are $(\overline{v}, \overline{y})$ such that

$$((\overline{u}, \overline{x}) \, , \, (\overline{v}, \overline{y})) \in \mathrm{HS}(\sigma) \qquad \text{and} \qquad ((\overline{v}, \overline{y}) \, , \, (\overline{w}, \overline{z})) \in \mathrm{HS}(\tau)$$

But by Prop. 6.2.2 this is possible if and only if the following equations are satisfied:

$$\begin{array}{rcl} \overline{v} & = & f_n\big(\overline{u}, \, \mathsf{next}(\overline{y})\big) \\ \overline{x} & = & F_n(\overline{u}, \, \overline{y}) \end{array} \qquad \begin{array}{rcl} \overline{w} & = & g_n\big(\overline{v}, \, \mathsf{next}(\overline{z})\big) \\ \overline{y} & = & G_n(\overline{v}, \, \overline{z}) \end{array} \tag{6.3}$$

(where $\mathsf{next}(\varepsilon) := \varepsilon$ and $\mathsf{next}(\overline{y}.y) := \overline{y}$). The derived equation

$$\overline{y} \quad = \quad G_n\big(f_n(\overline{u}, \, \mathsf{next}(\overline{y})) \, , \, \overline{z}\big) \tag{6.4}$$

determines $\overline{y} = y(\overline{u}, \overline{z}) = y_1 . \dots . y_n$ uniquely from $\overline{u} = u_1 . \dots . u_n$ and $\overline{z} = z_1 . \dots . z_n$, as

$$y_k \quad = \quad G_k(f_k(u_1 . \dots . u_k, \, y_1 . \dots . y_{k-1}) \, , \, z_1 . \dots . z_k) \tag{6.5}$$

We can thus define a pair of maps

$$\begin{array}{rcccl} h & : & \bigcup_{n>0} \big(U^n \times Z^{n-1}\big) & \longrightarrow & W \\ H & : & \bigcup_{n>0} \big(U^n \times Z^n\big) & \longrightarrow & X \end{array}$$

as follows:

$$\begin{array}{rcl} h_n(\overline{u}u, \, \overline{z}) & := & g_n(f_n(\overline{u}u, \, y(\overline{u}, \overline{z}))) \, , \, \overline{z}) \\ H_n(\overline{u}u, \, \overline{z}z) & := & F_n(\overline{u}u, \, y(\overline{u}u, \overline{z}z)) \end{array}$$

Then, by construction of (h, H), the total strategy $\theta : A \multimap_{\mathbf{DZ}} C$ it represents is such that $\mathrm{HS}(\theta) = \mathrm{HS}(\tau) \circ \mathrm{HS}(\sigma)$. It follows that $\theta = \tau \circ \sigma$, and that $\tau \circ \sigma$ is total zig-zag.

We thus have shown that total zig-zag strategies compose. Hence **DZ** is a category, and we have proved the first part of Prop. 6.2.1.

We now turn to the case of $\mathbf{DZ}^{\mathrm{W}}$, *i.e.* the case of winning total strategies. We rely on the Relational Zipping Lemma 4.7.1. Consider total winning zig-zag strategies $\sigma : A \multimap_{\mathbf{DZ}} B$ and $\tau : B \multimap_{\mathbf{DZ}} C$, where $A = (U, X)$, $B = (V, Y)$ and $C = (W, Z)$. Given an infinite play $\chi \in ((X + W) \cdot (U + Z))^\omega$ of $\tau \circ \sigma$ (*i.e.* such that $\chi(0) \cdot \ldots \cdot \chi(k) \in \tau \circ \sigma$ for infinitely many $k \in \mathbb{N}$), it follows from Lem. 4.7.1 that there are infinite plays χ_σ and χ_τ of resp. σ and τ such that

$$(\chi_\sigma)_{\restriction A} \quad = \quad \chi_{\restriction A} \qquad \text{and} \qquad (\chi_\sigma)_{\restriction B} \quad = \quad (\chi_\tau)_{\restriction B} \qquad \text{and} \qquad (\chi_\tau)_{\restriction C} \quad = \quad \chi_{\restriction C}$$

from which we get

$$(\chi_{\restriction A} \in \mathcal{W}_A) \quad \Rightarrow \quad ((\chi_\sigma)_{\restriction B} \ = \ (\chi_\tau)_{\restriction B} \ \in \ \mathcal{W}_B) \quad \Rightarrow \quad (\chi_{\restriction C} \in \mathcal{W}_C)$$

and we are done.

6.3. A Reconstruction of DZ

We have seen that total zig-zag strategies, originally defined as a restriction of a strategies in simple games, and can be given a direct construction via pairs of functions (Prop. 6.2.2). Our purpose here is to give an algebraic refinement of Prop. 6.2.2. It relies on an instance of Dialectica called *simple self-dualization* in [HS03], that we will perform it in the topos of trees $\mathscr{S}$.

A benefit of this detour is a conceptual description of basic monoidal structure of **DZ** (Cor. 6.3.7), which is instrumental in the model of [Rib18].

6.3.1. Simple Self Dualization. We describe here variants of well-known constructions of Dialectica-like categories, for which we refer to [dP91, Hyl02, HS03]. Given a category $\mathbb{C}$, its *simple self-dualization* is $\mathbf{G}(\mathbb{C}) := \mathbb{C} \times \mathbb{C}^{\mathrm{op}}$ (also written $\mathbb{C}^{\mathrm{d}}$ in [HS03]). Its objects are pairs U, X of objects of $\mathbb{C}$, and a morphism from (U, X) to (V, Y) is given by a pair of maps (f, F) with $f : U \to V$ and $F : Y \to X$, denoted

$$\text{where} \qquad \begin{array}{rcrcl} (f,F) & : & (U,X) & \to\!\!\!\!\!+ & (V,Y) \\ & & U & \xrightarrow{f} & V \\ & & X & \xleftarrow{F} & Y \end{array}$$

Assume now that $\mathbb{C}$ is Cartesian closed. Then $\mathbf{G}(\mathbb{C})$ can be equipped with a comonad (T, ϵ, δ) where T acts on objects as

$$T(U, X) \quad := \quad (U, X^U)$$

We are interested in the Kleisli category $\mathbf{DC}(\mathbb{C}) := \mathbf{Kl}(T)$. Explicitly, its objects are pairs of objects of $\mathbb{C}$, and a map from (U, X) to (V, Y) is a $\mathbf{G}(\mathbb{C})$-morphism (f, F) from $T(U, X)$ to (V, Y), that is

$$\text{where} \qquad \begin{array}{rcrcl} (f,F) & : & (U,X^U) & \to\!\!\!\!\!+ & (V,Y) \\ & & U & \xrightarrow{f} & V \\ & & X & \xleftarrow{F} & U \times Y \end{array}$$

(modulo exponential transpose). The category $\mathbf{DC}(\mathbb{C})$ is symmetric monoidal closed with structure

$$\begin{array}{rcl} (U,X) \otimes (V,Y) & = & (U \times V \ , \ X \times Y) \qquad\qquad \text{with unit } (\mathbf{1}, \mathbf{1}) \\ (U,X) \multimap (V,Y) & = & (V^U \times X^{U \times Y} \ , \ U \times Y) \end{array}$$

6.3.2. The Topos of Trees. The *topos of trees* $\mathscr{S}$ is the presheaf category over the order $(\mathbb{N} \setminus \{0\}, \leq)$ seen as a category. We refer to [BMSS12] for further background on the topos of trees, and to [MLM92] for presheaf categories in general.

Explicitly, an object X of $\mathscr{S}$ is given by a family of sets $(X_n)_{n>0}$ equipped with *restriction maps* $r^X_n : X_{n+1} \to X_n$. A morphism from X to Y is a natural transformation, that is a family of functions $f = (f_n : X_n \to Y_n)_n$ compatible with restriction maps, in the sense that $r^Y_n \circ f_{n+1} = f_n \circ r^X_n$, as in:

$$\begin{array}{ccccccccccc} X & \quad & X_1 & \xleftarrow{r^X_1} & X_2 & \longleftarrow \cdots \longleftarrow & X_n & \xleftarrow{r^X_n} & X_{n+1} & \longleftarrow & \cdots \\ {\scriptstyle f}\downarrow & & {\scriptstyle f_1}\downarrow & & {\scriptstyle f_2}\downarrow & & {\scriptstyle f_n}\downarrow & & {\scriptstyle f_{n+1}}\downarrow & & \\ Y & & Y_1 & \xleftarrow[r^Y_1]{} & Y_2 & \longleftarrow \cdots \longleftarrow & Y_n & \xleftarrow[r^Y_n]{} & Y_{n+1} & \longleftarrow & \cdots \end{array}$$

As a topos, $\mathscr{S}$ is Cartesian closed. The Cartesian product is computed pointwise, so that $(X \times Y)_n := X_n \times Y_n$. Exponentials X^Y are computed with the usual formula

$$(X^Y)_n \quad := \quad \mathsf{N}[\mathbb{N}[-, n] \times Y, X]$$

Explicitly, $(X^Y)_n$ consists of sequences of functions $(f_k : Y_k \to X_k)_{k \leq n}$ which are compatible with restriction. The restriction map of X^Y takes $(f_k)_{k \leq n+1} \in (X^Y)_{n+1}$ to $(f_k)_{k \leq n} \in (X^Y)_n$.

We will use the "*later*" functor $\blacktriangleright : \mathscr{S} \to \mathscr{S}$ of [BMSS12]. This functor shifts indices by 1 and inserts a dummy singleton set $\mathbf{1}$ at the beginning:

$$\begin{array}{lllllllll}
X : & X_1 \xleftarrow{r_1^X} & X_2 \longleftarrow & \cdots \xleftarrow{r_{n-1}^X} & X_n \xleftarrow{r_n^X} & X_{n+1} \longleftarrow & \cdots \\
\blacktriangleright X : & \mathbf{1} \xleftarrow{\mathbf{1}} & X_1 \xleftarrow{r_1^X} & \cdots \longleftarrow & X_{n-1} \xleftarrow{r_{n-1}^X} & X_n \xleftarrow{r_n^X} & \cdots
\end{array}$$

The later functor is moreover equipped with a natural transformation $\mathsf{next} : \mathrm{Id} \Rightarrow \blacktriangleright(-)$, whose component $\mathsf{next}^X : X \to \blacktriangleright X$ can be pictured as:

$$\begin{array}{ccccccccccc}
X & & X_1 & \xleftarrow{r_1^X} & X_2 & \longleftarrow \cdots \longleftarrow & X_n & \xleftarrow{r_n^X} & X_{n+1} & \longleftarrow & \cdots \\
\Big\downarrow {\scriptstyle \mathsf{next}^X} & & \Big\downarrow {\scriptstyle \mathbf{1}} & & \Big\downarrow {\scriptstyle r_1^X} & & \Big\downarrow {\scriptstyle r_{n-1}^X} & & \Big\downarrow {\scriptstyle r_n^X} & & \\
\blacktriangleright X & & \mathbf{1} & \xleftarrow[\mathbf{1}]{} & X_1 & \longleftarrow \cdots \longleftarrow & X_{n-1} & \xleftarrow[r_{n-1}^X]{} & X_n & \longleftarrow & \cdots
\end{array}$$

This structure allows $\mathscr{S}$ to be equipped with fixpoint operators $\mathsf{fix}^X : X^{\blacktriangleright X} \to X$, defined as

$$\mathsf{fix}^X_n((f_k)_{k \leq n}) \quad := \quad (f_n \circ \cdots \circ f_1)(\bullet)$$

where $(f_k : (\blacktriangleright X)_k \to X_k)_{k \leq n}$, as in

$$\begin{array}{l}
\mathbf{1} \xrightarrow{f_1} X_1 = (\blacktriangleright X)_2 \xrightarrow{f_2} X_2 = (\blacktriangleright X)_3 \xrightarrow{f_3} \cdots \\
\cdots \xrightarrow{f_{n-1}} X_{n-1} = (\blacktriangleright X)_n \xrightarrow{f_n} X_n
\end{array}$$

The maps fix^X are natural in X. Moreover, fix induce *unique* fixpoints, in the sense that given $f : \blacktriangleright X \times Y \to X$, writing $f^t : Y \to X^{\blacktriangleright X}$ for the exponential transpose of f, $\mathsf{fix}^X \circ f^t$ is the unique $h : Y \to X$ satisfying $f \circ \langle \mathsf{next}^X \circ h, \mathrm{id}_Y \rangle = h$ (see [BMSS12, Thm. 2.4]).

Given a set M, write $\mathsf{Str}(M)$ for the object of $\mathscr{S}$ with $\mathsf{Str}(M)_n := M^n$ and $r_n(\overline{m}.m) := \overline{m}$. Note that r_n is surjective.

6.3.3. Rebuilding DZ. The topos of trees allows us to treat composition in **DZ** in a more algebraic way than in §6.2. First, Prop. 6.2.2 can be reformulated as follows:

Proposition 6.3.1. *Given positive full games $A = (U, X)$ and $B = (V, Y)$, total zig-zag strategies $\sigma : A \multimap_{\mathbf{DZ}} B$ are in 1-1 correspondence with $\mathbf{G}(\mathscr{S})$-morphisms*

$$(f, F) \quad : \quad (\mathsf{Str}(U), \mathsf{Str}(X)^{\mathsf{Str}(U)}) \quad \longrightarrow\!\!\!\!\!\!+ \quad (\mathsf{Str}(V)^{\blacktriangleright \mathsf{Str}(Y)}, \mathsf{Str}(Y))$$

The interesting point is that the solution (6.5) of the recursive equation (6.4) can be obtained using the fixpoint combinator fix of the topos of trees. Similarly as in §6.2, consider positive full games $A = (U, X)$, $B = (V, Y)$ and $C = (W, Z)$, and $\mathbf{G}(\mathscr{S})$-morphisms

$$\begin{array}{lcl} (f, F) & : & (\mathsf{Str}(U), \mathsf{Str}(X)^{\mathsf{Str}(U)}) & \nrightarrow & (\mathsf{Str}(V)^{\blacktriangleright \mathsf{Str}(Y)}, \mathsf{Str}(Y)) \\ (g, G) & : & (\mathsf{Str}(V), \mathsf{Str}(Y)^{\mathsf{Str}(V)}) & \nrightarrow & (\mathsf{Str}(W)^{\blacktriangleright \mathsf{Str}(Z)}, \mathsf{Str}(Z)) \end{array}$$

Write σ and τ for the total zig-zag strategies corresponding to resp. (f, F) and (g, G). As in §6.2, the relational composite

$$\mathrm{HS}(\tau \circ \sigma) \quad = \quad \mathrm{HS}(\tau) \circ \mathrm{HS}(\sigma)$$

must be such that $((\overline{u}, \overline{x}), (\overline{w}, \overline{z})) \in \mathrm{HS}(\tau) \circ \mathrm{HS}(\sigma)$ if and only if there are $(\overline{v}, \overline{y})$ such that

$$((\overline{u}, \overline{x}), (\overline{v}, \overline{y})) \in \mathrm{HS}(\sigma) \qquad \text{and} \qquad ((\overline{v}, \overline{y}), (\overline{w}, \overline{z})) \in \mathrm{HS}(\tau)$$

This is equivalent to the system of equations (6.3), where next is now the action at index n of the $\mathscr{S}$-morphism next. As a consequence, equation (6.4) now uniquely defines $\overline{y}$ from $\overline{u}$ and $\overline{z}$ as

$$\overline{y} \quad = \quad y(\overline{u}, \overline{z}) \quad = \quad \mathsf{fix}^Y_n \big(\boldsymbol{\lambda} y. G_n(f_n(\overline{u}, y)\, ,\, \overline{z}) \big)$$

(We have here tacitly used the fact that $\xi \in (\mathsf{Str}(M)^{\blacktriangleright \mathsf{Str}(M)})_n$ is completely determined by its last component ξ_n.)

More generally, given $\mathbf{G}(\mathscr{S})$-objects (U, X), (V, Y), (W, Z), and $\mathbf{G}(\mathscr{S})$-morphisms

$$\begin{array}{lcl} (f, F) & : & (U, X^U) & \nrightarrow & (V^{\blacktriangleright Y}, Y) \\ (g, G) & : & (V, Y^V) & \nrightarrow & (W^{\blacktriangleright Z}, Z) \end{array}$$

we can define their composite

$$(g, G) \circ (f, F) \;=\; (h, H) \quad : \quad (U, X^U) \quad \nrightarrow \quad (W^{\blacktriangleright Z}, Z)$$

as follows (using the internal λ-calculus of $\mathscr{S}$):

$$\begin{array}{lrcl} & h(u, z) & := & g\big(f\big(u,\, y(\mathsf{next}(u), z)\big)\, ,\, z\big) \\ & H(z, u) & := & F(u\, ,\, y(u, z)) \\ \text{where} & y(u, z) & := & \mathsf{fix}^Y (\boldsymbol{\lambda} y. G(f(u, y), z)) \end{array}$$

It is possible to directly check that this composition is associative and preserves identities. We can actually do better: the operation

$$(-)^{\blacktriangleright} \quad : \quad (U, X) \quad \longmapsto \quad (U^{\blacktriangleright X}, X) \tag{6.6}$$

is the action on objects of a functor part of a monad, and the composition of $\mathbf{G}(\mathscr{S})$-morphisms

$$\begin{array}{lcl} (f, F) & : & TA & \nrightarrow & B^{\blacktriangleright} \\ (g, G) & : & TB & \nrightarrow & C^{\blacktriangleright} \end{array}$$

can be described by a distributive law of T over $(-)^{\blacktriangleright}$.

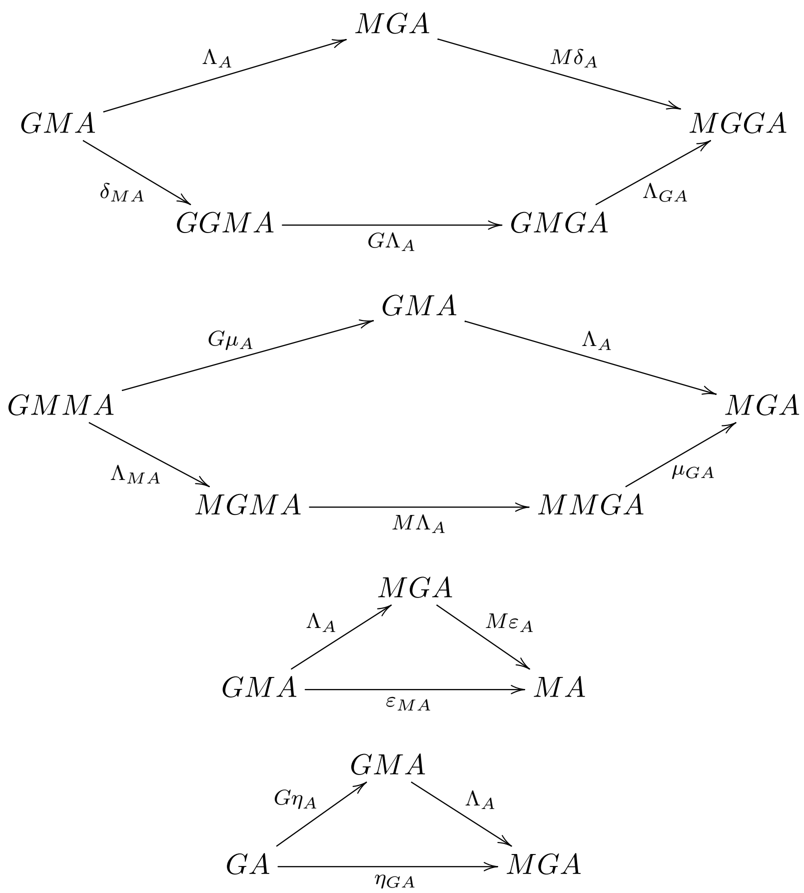


Figure 6.3.: Coherence for a Distributive Law of Comonad over a Monad.

6.3.4. A Distributive Law. Consider a category $\mathbb{C}$ equipped with a comonad (G, δ, ε) and monad (M, μ, η). A *distributive law* (in the sense of e.g. [HHM07]) of G over M is a natural transformation

$$\Lambda \quad : \quad G \circ M \quad \Longrightarrow \quad M \circ G$$

which satisfies the coherence conditions of Fig. 6.3. These coherence conditions ensure in particular that we can define a category $\mathbf{Kl}(\lambda)$, whose objects are the objects of $\mathbb{C}$, and whose morphisms are given by $\mathbf{Kl}(\lambda)[A, B] := \mathbb{C}[GA, MB]$.

In our case, the comonad is the comonad T of §6.3.1. For the monad, we equip the functor $(-)^{\blacktriangleright}$ of (6.6) with the unit and multiplication

$$\begin{array}{rcrcl} (f_\eta, F_\eta) & : & (U, X) & \nrightarrow & (U^{\blacktriangleright X}, X) \\ (f_\mu, F_\mu) & : & (U^{\blacktriangleright X \times \blacktriangleright X}, X) & \nrightarrow & (U^{\blacktriangleright X}, X) \end{array}$$

where $F_\eta = F_\mu = \mathrm{id}_X$, $f_\eta(u, x) = u$ and $f_\mu(h, x) = h(x, x)$ (modulo exponential transpose).

We now define the distributive law

$$\zeta \quad : \quad T \circ \blacktriangleright(-) \quad \Longrightarrow \quad \blacktriangleright(-) \circ T$$

Its component at object $A = (U, X)$ is the morphism

$$\begin{array}{rcrcl} \zeta_A = (f^\zeta, F^\zeta) & : & (U^{\blacktriangleright X}, X^{U^{\blacktriangleright X}}) & \to\!\!\!\!\!+ & (U^{\blacktriangleright (X^U)}, X^U) \\ \text{with} \qquad f^\zeta & : & U^{\blacktriangleright X} \times \blacktriangleright(X^U) & \longrightarrow & U \\ F^\zeta & : & U^{\blacktriangleright X} \times X^U & \longrightarrow & X \end{array}$$

defined as follows. Let $f^\zeta_1(\theta_1, \bullet) := \theta_1$. Given $\xi \in (X^U)_n$, $\theta \in (U^{\blacktriangleright X})_n$ and $\theta' \in (U^{\blacktriangleright X})_{n+1}$, let

$$\begin{array}{rcl} F^\zeta_n(\theta, \xi) & := & \mathsf{fix}^X_n(\xi \circ \theta) \\ f^\zeta_{n+1}(\theta', \xi) & := & \theta'_{n+1}(\mathsf{fix}^X_n(\xi \circ r_n(\theta'))) \\ & = & \theta'_{n+1}(F_n(r_n(\theta'), \xi)) \end{array}$$

Proposition 6.3.2 ([Rib18, App. F.5 & H]). *The family of maps $\zeta_A : T(A^\blacktriangleright) \to\!\!\!\!\!+ (TA)^\blacktriangleright$ forms a distributive law.*

We write $\mathbf{DZ}_{\mathscr{S}}$ for $\mathbf{Kl}(\zeta)$,

Corollary 6.3.3. *The category $\mathbf{DZ}$ is equivalent to the full subcategory of $\mathbf{DZ}_{\mathscr{S}}$ whose objects are of the form $(\mathsf{Str}(U), \mathsf{Str}(X))$.*

6.3.5. Symmetric Monoidal Structure. A benefit of Cor. 6.3.3 is that $\mathbf{DZ}$ (and $\mathbf{DZ}_{\mathscr{S}}$) can be seen as inheriting the monoidal structure of $\mathbf{DC}(\mathscr{S})$, which is itself given by the monoidal product $\otimes$ of $\mathbf{G}(\mathscr{S})$. First, it is well-known (see [HS03]) that for an SMC $(\mathbb{C}, \otimes, \mathbf{I})$, the category $\mathbf{G}(\mathbb{C})$ is symmetric monoidal with

$$(U, X) \otimes (V, Y) \quad = \quad (U \otimes V, X \otimes Y) \qquad \text{with unit} \quad \mathbf{I} = (\mathbf{I}, \mathbf{I})$$

Moreover, it is easy to see that the monad T is lax symmetric monoidal and that the monad $(-)^\blacktriangleright$ is oplax symmetric monoidal (we refer to e.g. [Mel09] for definitions). Furthermore, the following is easy to verify (see [Rib18, App. D.4.3 & D.5]).

Proposition 6.3.4. *Consider a lax symmetric monoidal monad T and an oplax symmetric monoidal comonad G on a symmetric monoidal category $(\mathbb{C}, \otimes, \mathbf{I})$. Assume a distributive law $\Lambda : GT \Rightarrow TG$ such that*

$$\begin{array}{ccc} G(TA \otimes TB) & \xrightarrow{G(m^2_{A,B})} & GT(A \otimes B) \\ \downarrow{\scriptstyle g^2_{TA,TB}} & & \downarrow{\scriptstyle \Lambda_{A \otimes B}} \\ GTA \otimes GTB & & TG(A \otimes B) \\ \downarrow{\scriptstyle \Lambda_A \otimes \Lambda_B} & & \downarrow{\scriptstyle T(g^2_{A,B})} \\ TGA \otimes TGB & \xrightarrow[m^2_{GA,GB}]{} & T(GA \otimes GB) \end{array} \tag{6.7}$$

where (m^2, m^0) is the strength of T and (g^2, g^0) is the strength of G.

Then $\mathbf{Kl}(\lambda)$ is symmetric monoidal, with, on objects, *the same monoidal structure as* $(\mathbb{C}, \otimes, \mathbf{I})$.

Proposition 6.3.5. *The distributive law $\zeta : T \circ (-)^\blacktriangleright \Rightarrow (-)^\blacktriangleright \circ T$ satisfies (6.7).*

Corollary 6.3.6. *The category $\mathbf{DZ}_{\mathscr{S}}$ is symmetric monoidal.*

Since $\mathsf{Str}((M \times N)) \simeq \mathsf{Str}(M) \times \mathsf{Str}(N)$, we also get:

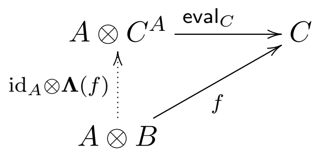
Corollary 6.3.7. *The category $\mathbf{DZ}$ is symmetric monoidal. On objects we have*

$$(U, X) \otimes (V, Y) \quad = \quad (U \otimes V, X \otimes Y) \qquad \textit{with unit} \quad \mathbf{I} = (\mathbf{I}, \mathbf{I})$$

6.3.6. Monoidal Closure. Recall from e.g. [Mel09] that a symmetric monoidal category $(\mathbb{C}, \otimes, \mathbf{I})$ is *closed* if for every object A, the functor $A \otimes (-)$ has a right adjoint $(-)^A$. According to [ML98, Thm. IV.1.2], it is sufficient to show that for every object C there is an object C^A and map

$$\mathsf{eval}_C \quad : \quad A \otimes C^A \quad \longrightarrow \quad C$$

such that for every $f : A \otimes B \to C$ there is a unique $\mathbf{\Lambda}(f) : B \to C^A$ with



Proposition 6.2.2 gives a very simple way to describe the monoidal closed structure of $(\mathbf{DZ}, \otimes, \mathbf{I})$ (with $\otimes, \mathbf{I}$ given by Cor. 6.3.7). The idea is that the representation of $\mathbf{DZ}$-morphisms as pairs of functions $(f, F) : (U, X) \longrightarrow\!\!\!\!\!\!\!\!\;\, + \;\, (V, Y)$ as in (6.2) can be read as:

$$\begin{array}{lclcl} f & : & \bigcup_{n \in \mathbb{N}} (U^n \times Y^n) & \longrightarrow & (U \longrightarrow V) \\ F & : & \bigcup_{n \in \mathbb{N}} (U^n \times Y^n) & \longrightarrow & (U \times Y \longrightarrow X) \end{array}$$

Proposition 6.3.8. *The category* $\mathbf{DZ}$ *is symmetric monoidal closed. The linear exponent of* $A = (U, X)$ *and* $B = (V, Y)$ *is* $A \multimap_{\mathbf{DZ}} B \;:=\; (V^U \times X^{U \times Y}, U \times Y)$.

6.4. Toward a Simple Approach to Synchronous Arrow Games

The material of §6.3 gives a very simple analogue of the indexed categories $\mathbf{SAG}_{(-)}$ of Chap. 5. We shall content ourselves with an outline since much of the details of this Section are provided in [Rib18].

We fix the following objects of $\mathbf{DZ}$:

$$\mathfrak{D} \quad := \quad (\mathbf{1}, \mathfrak{D}) \qquad \text{and} \qquad \Sigma \quad := \quad (\Sigma, \mathbf{1}) \qquad\qquad \text{(for each alphabet } \Sigma)$$

The idea is that one may consider a variant of $\mathbf{SAG}_\Sigma$ consisting of those $\mathbf{DZ}$-strategies

$$\sigma \quad : \quad \Sigma \otimes (A \otimes \mathfrak{D}) \quad \multimap_{\mathbf{DZ}} \quad \Sigma \otimes (B \otimes \mathfrak{D})$$

which satisfy a synchronicity constraint obtained by the obvious adaptation of the trace functions tr of §5.2. But such synchronous (total zig-zag) strategies can actually be equivalently represented by $\mathbf{DZ}$-maps

$$\Sigma \otimes A \quad \multimap_{\mathbf{DZ}} \quad B \otimes \mathfrak{D}$$

The key to make this work is to note the following. First, objects of the form Σ (resp. $\mathfrak{D}$) are commutative comonoids (resp. monoids) in $\mathbf{DZ}$. As a consequence, tensoring with Σ (resp. with $\mathfrak{D}$) gives an oplax symmetric monoidal comonad $\Sigma \otimes (-)$ of *comonoid indexing with* Σ (resp. a lax symmetric monoidal monad $(-) \otimes \mathfrak{D}$ of *monoid indexing with* $\mathfrak{D}$) [HS99, HS03]. Then, it is easy to see that the associativity structure maps of $\mathbf{DZ}$ induce a distributive law (in the sense of [HHM07] and §6.3.4)

$$\Phi^{\Sigma}_{(-)} \quad := \quad \alpha^{-1}_{\Sigma, (-), \mathfrak{D}} \quad : \quad \Sigma \otimes ((-) \otimes \mathfrak{D}) \quad \Longrightarrow \quad (\Sigma \otimes (-)) \otimes \mathfrak{D} \tag{6.8}$$

which moreover satisfies the additional requirement (6.7) of Prop. 6.3.4. We thus get:

Proposition 6.4.1. *For each alphabet* Σ*, the category* $\mathsf{DialZ}(\Sigma) := \mathbf{Kl}(\Phi^{\Sigma})$ *is symmetric monoidal. On objects we have*

$$(U, X) \otimes (V, Y) \quad = \quad (U \otimes V, X \otimes Y) \qquad \textit{with unit} \quad \mathbf{I} = (\mathbf{I}, \mathbf{I})$$

Hence, the objects of $\mathsf{DialZ}(\Sigma)$ are full positive games $A = (U, X)$, $B = (V, Y)$ etc, and a morphism of $\mathsf{DialZ}(\Sigma)$ from A to B is a total zigzag strategy

$$\sigma \quad : \quad \Sigma \otimes A \quad \multimap_{\mathbf{DZ}} \quad B \otimes \mathfrak{D}$$

In the following, we let $\mathbf{DZ}_{\mathfrak{D}}$ be the Kleisli category $\mathbf{Kl}(\mathfrak{D})$ of the monad $(-) \otimes \mathfrak{D}$ on $\mathbf{DZ}$. Note that $\mathbf{DZ}_{\mathfrak{D}}$ is symmetric monoidal as $(-) \otimes \mathfrak{D}$ is lax symmetric monoidal.

6.4.1. Symmetric Monoidal Closed Structure. The monoidal closed structure of $\mathbf{DZ}$ lifts to $\mathbf{DZ}_{\mathfrak{D}}$ and to $\mathsf{DialZ}(\Sigma)$. In the case of $\mathbf{DZ}_{\mathfrak{D}}$, since

$$\mathbf{DZ}_{\mathfrak{D}}[A \otimes B\ ,\ C] \quad = \quad \mathbf{DZ}[A \otimes B\ ,\ C \otimes \mathfrak{D}] \quad \simeq \quad \mathbf{DZ}[A\ ,\ (B \multimap_{\mathbf{DZ}} C \otimes \mathfrak{D})]$$

we should have $(A \multimap_{\mathbf{DZ}_{\mathfrak{D}}} B) \otimes \mathfrak{D} \simeq (A \multimap_{\mathbf{DZ}} B \otimes \mathfrak{D})$. Given $A = (U, X)$ and $B = (V, Y)$ this leads to $(A \multimap_{\mathbf{DZ}_{\mathfrak{D}}} B) = (W, Z)$ with

$$(W, Z \times \mathfrak{D}) \quad \simeq \quad (V^U \times X^{U \times Y \times \mathfrak{D}}\ ,\ U \times Y \times \mathfrak{D})$$

We therefore let

$$(U, X) \multimap_{\mathbf{DZ}_{\mathfrak{D}}} (V, Y) \quad := \quad (V^U \times X^{U \times Y \times \mathfrak{D}}\ ,\ U \times Y)$$

The closed structure of $\mathbf{DZ}_{\mathfrak{D}}$ directly lifts to $\mathsf{DialZ}(\Sigma)$ since

$$\mathsf{DialZ}(\Sigma)[A \otimes B\ ,\ C] \quad = \quad \mathbf{DZ}_{\mathfrak{D}}[\Sigma \otimes (A \otimes B)\ ,\ C] \quad \simeq \quad \mathbf{DZ}_{\mathfrak{D}}[\Sigma \otimes A\ ,\ B \multimap_{\mathbf{DZ}_{\mathfrak{D}}} C]$$

Proposition 6.4.2. *The categories* $\mathbf{DZ}_{\mathfrak{D}}$ *and* $\mathsf{DialZ}(\Sigma)$ *are symmetric monoidal closed.*

6.4.2. Indexed Structure. Using a small amount of fibred category theory (for which we refer to [Jac01]), we obtain a simple (strict) indexed structure on the categories $\mathsf{DialZ}(-)$, which may be reminiscent from [MM15].

We follow the pattern of *simple fibrations* $\mathsf{s} : \mathsf{s}(\mathbb{B}) \to \mathbb{B}$ over a category $\mathbb{B}$ with finite products (see e.g. [Jac01, Chap. 1] but also [Hyl02, Hof11]). Recall that for a simple fibration $\mathsf{s} : \mathsf{s}(\mathbb{B}) \to \mathbb{B}$, the fibre over I is the Kleisli category of the comonad of comonoid indexing with I.

Fix a symmetric monoidal category $\mathbb{C}$. The category $\mathbf{Comon}(\mathbb{C})$ of commutative comonoids of $\mathbb{C}$ has finite products. This gives a (strict) indexed category

$$(-)^{\star} \quad : \quad \mathbf{Comon}(\mathbb{C})^{\mathrm{op}} \longrightarrow \mathbf{Cat}$$

taking each comonoid K to the Kleisli category $\mathbf{Kl}(K)$ of comonoid indexing with K. On maps, $(-)^{\star}$ takes a comonoid morphism $u : K \to L$ to the functor $u^{\star} : \mathbf{Kl}(L) \to \mathbf{Kl}(K)$ which is the identity on objects and takes $f : L \otimes A \to B$ to $f \circ (u \otimes \mathrm{id}_A) : K \otimes A \to B$.

In particular it would make sense to consider the above indexed category for the particular case of $\mathbf{Comon}(\mathbf{DZ}_{\mathfrak{D}})$:

$$(-)^{\star} \quad : \quad \mathbf{Comon}(\mathbf{DZ}_{\mathfrak{D}})^{\mathrm{op}} \longrightarrow \mathbf{Cat}$$

We shall actually do this for a full subcategory of $\mathbf{Comon}(\mathbf{DZ}_{\mathfrak{D}})^{\mathrm{op}}$ corresponding to the base category $\mathbf{T}$ of Def. 4.2.1. Note that for each alphabet Σ, the object Σ of $\mathbf{DZ}$ is still a commutative comonoid in $\mathbf{DZ}_{\mathfrak{D}}$ (see e.g. [Rib18, Prop. D.11, App. D.6.5]), while the distributive law Φ^{Σ} of (6.8) implies that the comonad of comonoid indexing with Σ lifts to $\mathbf{DZ}_{\mathfrak{D}}$. In particular, the category $\mathsf{DialZ}(\Sigma)$ is the Kleisli category of comonoid indexing with Σ in $\mathbf{DZ}_{\mathfrak{D}}$.

We moreover note the following.

Proposition 6.4.3. *The base category* $\mathbf{T}$ *of Def. 4.2.1 is (isomorphic to) the full subcategory of* $\mathbf{Comon}(\mathbf{DZ}_{\mathfrak{D}})$ *whose objects are of the form* $(\Sigma, \mathbf{1})$ *for* Σ *an alphabet.*

This induces a (strict) indexed structure

$$\mathsf{DialZ}(-) \quad := \quad (-)^{\star} \quad : \quad \mathbf{T}^{\mathrm{op}} \longrightarrow \mathbf{Cat}$$

Explicitly, this indexed category takes an alphabet Σ to the category $\mathsf{DialZ}(\Sigma)$ and a $\mathbf{T}$-morphism $M : \Sigma \to \Gamma$ to the functor $M^{\star} : \mathsf{DialZ}(\Gamma) \to \mathsf{DialZ}(\Sigma)$ defined as above.

Note that since objects of the form $\Sigma = (\Sigma, \mathbf{1})$ are closed under $\otimes$ and are commutative comonoids in $\mathbf{DZ}_{\mathfrak{D}}$, and since the commutative comonoids of a symmetric monoidal category have finite products, it follows that the base category $\mathbf{T}$ has finite products.

6.4.3. Quantifications. We easily get universal and existential quantifications (also called resp. *simple products* and *simple coproducts*, see e.g. [Jac01, Chap. 1]) for the indexed category $\mathsf{DialZ}(-)$.

Let $(-)^{\star} : \mathbb{B} \to \mathbf{Cat}$ be an indexed category. For each object I of $\mathbb{B}$, we write $\mathbb{E}_I$ for the category $I^{\star}$. Recall from e.g. [Jac01, Chap. 1] (see also §4.2) that existential (resp. universal) quantifications for $(-)^{\star}$ are given given resp. by left adjoints $\coprod_{I,J} : \mathbb{E}_{I\times J} \to \mathbb{E}_I$ and right adjoints $\prod_{I,J} : \mathbb{E}_{I\times J} \to \mathbb{E}_I$ to the *weakening functors* $\pi^{\star} : \mathbb{E}_I \to \mathbb{E}_{I\times J}$ induced by $\mathbb{B}$-projections $\pi : I \times J \to I$. The families of operations $(\coprod_{I,J})_{I,J}$ and $(\prod_{I,J})_{I,J}$ are moreover required to satisfy some coherence conditions, called the *Beck-Chevalley* conditions, which insure that they are preserved by substitution.

It is well-known (see e.g. [Jac01, Chap. 1]) that the simple fibration $\mathsf{s} : \mathsf{s}(\mathbb{B}) \to \mathbb{B}$ always has simple coproducts, and has simple products iff $\mathbb{B}$ is Cartesian closed. They are given by

$$\coprod_{I,J}(I \times J, X) \quad := \quad (I, J \times X) \qquad \text{and} \qquad \prod_{I,J}(I \times J, X) \quad := \quad (I, X^J)$$

This directly extends to DialZ.

Proposition 6.4.4. *The weakening functors* $[\pi]^{\star} : \mathsf{DialZ}(\Sigma) \to \mathsf{DialZ}(\Sigma \times \Gamma)$ *induced by projection functions* $\pi : \Sigma \times \Gamma \to \Sigma$ *have left and right adjoints given by*

$$\coprod_{\Sigma,\Gamma}(U, X) := (\Gamma \times U, X) \qquad \text{and} \qquad \prod_{\Sigma,\Gamma}(U, X) := (U^{\Gamma}, \Gamma \times X) \simeq (\Gamma \multimap_{\mathbf{DZ}_{\mathfrak{D}}} (U, X))$$

The Beck-Chevalley conditions amount, for $L \in \mathbf{T}[\Delta, \Sigma]$, to the equalities

$$L^{\star}(\bigsqcup_{\Sigma,\Gamma}(U, X)) \quad = \quad \bigsqcup_{\Delta,\Gamma}(L \times \mathrm{Id}_{\Gamma})^{\star}(U, X) \qquad \text{(for } \bigsqcup \in \{\textstyle\coprod, \prod\})$$

which follow from the fact that substitution functors are identities on objects.

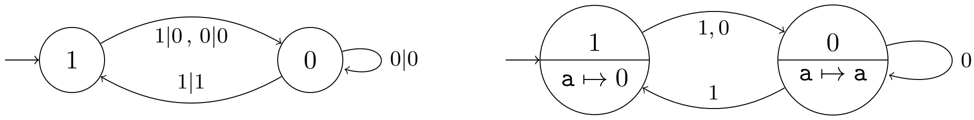


Figure 6.4.: A Mealy machine (left) and an equivalent eager (Moore) machine (right).

6.5. Finite State Strategies as Mealy Machines

We now provide some details on the representation of *finite-state* **DZ** morphisms. The key is to simulate enough of the ▶ and next operations of §6.3 in the setting of Mealy and eager Moore machines (§2.3, see also [PR19, Pra19]).

6.5.1. Back to Causal and Eager Functions. Causal and f.s. causal functions form categories with finite products. Let $\mathbf{S}$ be the category whose objects are alphabets and whose maps from Σ to Γ are causal functions $F : \Sigma^\omega \to \Gamma^\omega$.

Moreover, recall that the identity function $\Sigma^\omega \to \Sigma^\omega$ is f.s. causal, and note that the composition of f.s. causal functions is f.s. causal. We let $\mathbf{M}$ be the wide subcategory of $\mathbf{S}$ whose maps are finite-state causal functions. Note that in order to obtain the required identity and composition laws, it is crucial that $\mathbf{M}$ is a category of *functions* and not of *machines*.

Proposition 6.5.1. *The Cartesian product of $\Sigma_1, \ldots, \Sigma_n$ (for $n \geq 0$) in $\mathbf{S}, \mathbf{M}$ is given by the product of sets $\Sigma_1 \times \cdots \times \Sigma_n$ (so that $\mathbf{1}$ is terminal).*

Eager functions do not form a category since the identity of $\mathbf{S}$ is not eager. On the other hand, eager functions are closed under composition with causal functions.

Proposition 6.5.2. *If F is eager and G, H are causal then $H \circ F \circ G$ is eager.*

Note the usual currying bijections

$$\Sigma^+ \to \Gamma \quad \simeq \quad \Sigma^* \times \Sigma \to \Gamma \quad \simeq \quad \Sigma^* \to \Gamma^\Sigma$$

Hence, eager functions $\Sigma \to_{\mathbf{E}} \Gamma^\Sigma$ are in bijection with causal functions $\Sigma \to_{\mathbf{S}} \Gamma$. This easily extends to machines. Given a Mealy machine $\mathcal{M} : \Sigma \to \Gamma$, let $\mathbf{\Lambda}(\mathcal{M}) : \Sigma \to \Gamma^\Sigma$ be the eager machine defined as $\mathcal{M}$ but with output map taking $q \in Q_\mathcal{M}$ to $(\mathtt{a} \mapsto \lambda_\mathcal{M}(q, \mathtt{a})) \in \Gamma^\Sigma$.

Example 6.5.3. *Figure 6.4 (left) displays the Mealy machine $\mathcal{M} : \mathbf{2} \to \mathbf{2}$ of Ex. 2.3.1.(c). Then $\mathbf{\Lambda}(\mathcal{M}) : \mathbf{2} \to \mathbf{2}^\mathbf{2}$ is the eager machine displayed in Fig. 6.4 (right, where the output is indicated within states).*

Notation 6.5.4. *We use the following notations on eager functions.*

- *First, let @ be the pointwise lift to $\mathbf{M}$ of the usual application function $\Gamma^\Sigma \times \Sigma \to \Gamma$. We often write $(F)G$ for $@(F, G)$.*

  Consider a Mealy machine $\mathcal{M} : \Sigma \to \Gamma$ and the induced eager machine $\mathbf{\Lambda}(\mathcal{M}) : \Sigma \to \Gamma^\Sigma$. We have
  $$F_\mathcal{M}(B) \quad = \quad @\big(F_{\mathbf{\Lambda}(\mathcal{M})}(B),\, B\big) \qquad \text{(for all } B \in \Sigma^\omega\text{)}$$

- *Given an eager $F : \Gamma \to_{\mathbf{E}} \Sigma^\Gamma$, we write $\mathrm{ev}(F)$ for the causal $@(F(-), -) : \Gamma \to_{\mathbf{S}} \Sigma$.*

- *Given a causal* $F : \Gamma \to_{\mathbf{S}} \Sigma$*, we write* $\mathbf{\Lambda}(F)$ *for the eager* $\Gamma \to_{\mathbf{E}} \Sigma^\Gamma$ *such that* $F = \mathrm{ev}(\mathbf{\Lambda}(F))$.
- *Given a (f.s.) causal* $F : \Sigma \times \Gamma \to_{\mathbf{S}} \Delta$*, we say that* F *is* eager in Γ *if* F *can be represented by a(n) (f.s.) eager function* $\mathbf{\Lambda}(F) : \Sigma \times \Gamma \to_{\mathbf{E}} \Delta^\Sigma$ *as*

$$F(B, C) \quad = \quad @\big(\mathbf{\Lambda}(F)(B, C),\, B\big) \qquad\qquad (\text{for all } B \in \Sigma^\omega,\, C \in \Gamma^\omega)$$

Eager functions admit fixpoints similar to those of contractive maps in the topos of trees (see [BMSS12, Thm. 2.4] and §6.3.2).

Proposition 6.5.5. *For each* $F : \Sigma \times \Gamma \longrightarrow_{\mathbf{E}} \Sigma^\Gamma$ *there is a* $\mathsf{fix}(F) : \Gamma \longrightarrow_{\mathbf{E}} \Sigma^\Gamma$ *s.t.*

$$\mathsf{fix}(F)(C) \quad = \quad F\big(\mathrm{ev}(\mathsf{fix}(F))(C)\,,\, C\big) \qquad\qquad (\text{for all } C \in \Gamma^\omega)$$

If F *is induced by the eager machine* $\mathcal{E} : \Sigma \times \Gamma \to \Sigma^\Gamma$*, then* $\mathsf{fix}(F)$ *is induced by the eager* $\mathcal{H} : \Gamma \to \Sigma^\Gamma$ *defined as* $\mathcal{E}$ *but with* $\partial_{\mathcal{H}} : (q, \mathtt{b}) \mapsto \partial_{\mathcal{E}}\big(q,\, ((\lambda_{\mathcal{E}}(q))\mathtt{b}, \mathtt{b})\big)$.

6.5.2. Representation of Total Zig-Zag Strategies. Fix full positive games $A = (U, X)$ and $B = (V, Y)$ with U, X, V, Y non-empty and finite. It follows from the monoidal closure of $\mathbf{DZ}$ that a total zigzag P-strategy $\sigma : A \multimap B$ can be represented as a P-strategy in the full positive game $A \multimap_{\mathbf{DZ}} B$ of Prop. 6.3.8, and thus as an eager function. This induces a notion of finite-state strategy for $\mathbf{DZ}$. A more explicit formulation can be given as follows.

Recall from Prop. 6.2.2 that a total zig-zag strategy is given by a pair of functions (f, F) with

$$\begin{array}{lcccl} f & : & \bigcup_{n>0} (U^n \times Y^{n-1}) & \longrightarrow & V \\ F & : & \bigcup_{n>0} (U^n \times Y^{n}) & \longrightarrow & X \end{array}$$

With the terminology of §2.3 and §6.5.1, this is amounts to say that F induces a causal function $U \times Y \to_{\mathbf{S}} X$ and that f induces a causal function $U \times Y \to_{\mathbf{S}} V$ which is eager in Y.

Definition 6.5.6 (Finite-State Strategy)**.** *Given* A *and* B *as above, we say that a strategy* $\sigma : A \multimap_{\mathbf{DZ}} B$ *is* finite-state *if, w.r.t. to its functional representation* (f, F) *via Prop. 6.2.2, the function* F *induces a finite-state function* $U \times Y \to_{\mathbf{M}} X$ *and the function* f *induces a finite-state function* $U \times Y \to_{\mathbf{M}} V$ *which is eager in* Y.

That finite-state strategies compose can be proved similarly as for total zig-zag strategies (§6.2 and §6.3.3), using the *finite-state* fixpoint operator of Prop. 6.5.5. Moreover, the categorical structure of $\mathbf{DZ}$ presented in this Section, namely the monoidal closed structure and quantifications, restricts to finite-state strategies. This is easy to check (but quite tedious), and we refer to [PR19, App. B] for details.

Recall from §6.4 that a morphism of $\mathsf{DialZ}(\Sigma)$ from A to B (with A, B as above) is a total zig-zag strategy

$$\sigma \quad : \quad \Sigma \otimes A \quad \multimap_{\mathbf{DZ}} \quad B \otimes \mathfrak{D}$$

and that a morphism of the base category $\mathbf{T}$ (Def. 4.2.1 and Prop. 6.4.3) is a total zig-zag strategy

$$M \quad : \quad \Sigma \quad \multimap_{\mathbf{DZ}} \quad \Gamma \otimes \mathfrak{D}$$

Hence, Def. 6.5.6 immediately gives a notion of finite-state morphism of $\mathsf{DialZ}(\Sigma)$ and of finite-state $\mathbf{T}$-morphism. Moreover, it is easy to see that the wide subcategory of $\mathbf{T}$ consisting of the finite-state morphisms has finite products (see also §6.4.2). Explicitly, recalling from Def. 4.2.1 that a morphism $M : \Sigma \to_{\mathbf{T}} \Gamma$ can be represented as a function

$$M \quad : \quad \bigcup_{n>0} \left(\Sigma^n \times \mathfrak{D}^{n-1}\right) \quad \longrightarrow \quad \Gamma$$

that is, as a causal function $M : \Sigma \times \mathfrak{D} \to_{\mathbf{S}} \Gamma$ which is eager in $\mathfrak{D}$, we say that M is a *finite state* $\mathbf{T}$ *morphism* if M is a f.s. causal function. The explicit description of finite-state $\mathsf{DialZ}(\Sigma)$ morphisms is similar and omitted.

7. Categories of Uniform Tree Automata

This chapter presents the notion of "*uniform*" tree automata from [Rib18]. Uniform automata are essentially a presentation of alternating automata whose categories of substituted acceptance games are, when forgetting about winning, full subcategories of $\mathsf{DialZ}(-)$. As a consequence, uniform automata inherit the structure on zig-zag games presented in Chap. 6, thereby solving the difficulty raised in §5.5 w.r.t. the functoriality of linear negation on usual alternating automata.

We begin by giving the definition of uniform automata, while §7.1 presents the specialization of $\mathsf{DialAut}$ to uniform automata. Then §7.2 presents a basic set of connectives on uniform automata, mainly based on the corresponding structure for DialZ presented in Chap. 6. Non-deterministic automata and the exponential modality $!(-)$ induced by the *Simulation Theorem* [EJ91, MS95] are briefly discussed in §7.3 and §7.4, with an emphasis on applications. Finally, we sketch in §7.5 the particular case of uniform automata on ω-words, which is further developed in [PR18b, Pra19].

Definition 7.0.1 (Uniform Tree Automata)**.** *A* uniform tree automaton $\mathcal{A}$ *over* Σ *(notation* $\mathcal{A} : \Sigma$*) has the form*

$$\mathcal{A} \quad = \quad (Q_{\mathcal{A}}\,,\, q^{\imath}_{\mathcal{A}}\,,\, U\,,\, X\,,\, \partial_{\mathcal{A}}\,,\, \Omega_{\mathcal{A}}) \tag{7.1}$$

where $Q_{\mathcal{A}}$ *is the finite set of* states*,* $q^{\imath}_{\mathcal{A}} \in Q_{\mathcal{A}}$ *is the* initial state*,* U *and* X *are finite non-empty sets of resp.* P *and* O*-moves, the* acceptance condition $\Omega_{\mathcal{A}}$ *is an* ω*-regular subset of* $Q^{\omega}_{\mathcal{A}}$*, and the* transition function $\partial_{\mathcal{A}}$ *has the form*

$$\partial_{\mathcal{A}} \quad : \quad Q_{\mathcal{A}} \times \Sigma \quad \longrightarrow \quad U \times X \quad \longrightarrow \quad (\mathfrak{D} \longrightarrow Q_{\mathcal{A}}) \tag{7.2}$$

Following the usual terminology, an automaton $\mathcal{A}$ *as in (7.1) is* non-deterministic *if* $X \simeq \mathbf{1}$*,* universal *if* $U \simeq \mathbf{1}$*, and* deterministic *if* $U \simeq X \simeq \mathbf{1}$*.*

Example 7.0.2. *(i) The* unit automaton $\mathbf{I}_{\Sigma} : \Sigma$ *is the unique uniform deterministic automaton over* Σ *with state set* $\mathbf{1}$ *(with* $\bullet$ *initial) and acceptance condition* $\mathbf{1}^{\omega}$*. Explicitly,*

$$\mathbf{I}_{\Sigma} \quad := \quad (\mathbf{1}, \bullet, \mathbf{1}, \mathbf{1}, \partial_{\mathbf{1}}, \mathbf{1}^{\omega})$$

where $\partial_{\mathbf{1}}$ *is the unique function*

$$\partial_{\mathbf{1}} \quad : \quad \mathbf{1} \times \Sigma \quad \longrightarrow \quad \mathbf{1} \times \mathbf{1} \quad \longrightarrow \quad (\mathfrak{D} \longrightarrow \mathbf{1})$$

We write $\mathbf{I}$ *for* $\mathbf{I}_{\Sigma}$ *when* Σ *is clear from the context.*

(ii) Each alternating automaton $\mathcal{A}$ *can be translated to a uniform automaton* $\overline{\mathcal{A}}$*. The automaton* $\overline{\mathcal{A}}$ *simulates* $\mathcal{A}$ *as long as* P *and* O *respect the transition function of* $\mathcal{A}$*, and switches to an accepting (resp. rejecting) state as soon as* O *(resp.* P*) plays a move not allowed by* $\mathcal{A}$*. Assuming*

$$\partial_{\mathcal{A}} \quad : \quad Q_{\mathcal{A}} \times \Sigma \quad \longrightarrow \quad \mathcal{P}(\mathcal{P}(Q_{\mathcal{A}} \times \mathfrak{D}))$$

we let $\overline{\mathcal{A}}$ be the uniform automaton

$$(\overline{\mathcal{A}} : \Sigma) \quad := \quad (Q_{\mathcal{A}} + \mathbb{B}\,,\, q^{\imath}_{\mathcal{A}}\,,\, \mathcal{P}(Q_{\mathcal{A}} \times \mathfrak{D})\,,\, Q_{\mathcal{A}}\,,\, \partial_{\overline{\mathcal{A}}}\,,\, \Omega_{\overline{\mathcal{A}}})$$

where $\mathbb{B} := \{\mathbb{t}, \mathbb{f}\}$, with transitions given by $\partial_{\overline{\mathcal{A}}}(\mathbb{b}, \mathtt{a}, -, -, -) := \mathbb{b}$ if $\mathbb{b} \in \mathbb{B}$ and for $q \in Q_{\mathcal{A}}$:

$$\partial_{\overline{\mathcal{A}}}(q, \mathtt{a}, \gamma, q', d) := \begin{cases} q' & \text{if } \gamma \in \partial_{\mathcal{A}}(q, \mathtt{a}) \text{ and } (q', d) \in \gamma \\ \mathbb{t} & \text{if } \gamma \in \partial_{\mathcal{A}}(q, \mathtt{a}) \text{ and } (q', d) \notin \gamma \\ \mathbb{f} & \text{if } \gamma \notin \partial_{\mathcal{A}}(q, \mathtt{a}) \end{cases}$$

and with $\Omega_{\overline{\mathcal{A}}} \;:=\; \Omega_{\mathcal{A}} + Q^*_{\mathcal{A}}.\mathbb{t}^{\omega}$.

7.1. Indexed Categories of Uniform Tree Automata

We now define the analogue for uniform automata of substituted acceptance games and of (linear) synchronous arrow games, respectively presented in §5.1 and §5.2.

7.1.1. Uniform Substituted Acceptance Games. Consider a uniform automaton $\mathcal{A} : \Gamma$ as in (7.1), and a morphism $M \in \mathbf{T}[\Sigma, \Gamma]$. The *uniform substituted acceptance game* $\Sigma \vdash \mathcal{A}(M)$ is the full positive game with P-moves $\Sigma \times U$ and O-moves $X \times \mathfrak{D}$. So a play in $\Sigma \vdash \mathcal{A}(M)$ has the form

$$\begin{array}{ccccccccccccc} \mathsf{P} & & \mathsf{O} & & \mathsf{P} & & \mathsf{O} & & & & \mathsf{P} & & \mathsf{O} \\ (\mathtt{a}_0, u_0) & \cdot & (x_0, d_0) & \cdot & (\mathtt{a}_1, u_1) & \cdot & (x_1, d_1) & \cdot & \cdots & \cdot & (\mathtt{a}_n, u_n) & \cdot & (x_n, d_n) \quad \cdot \quad \cdots \\ \pitchfork & & \pitchfork & & \pitchfork & & \pitchfork & & & & \pitchfork & & \pitchfork \\ \Sigma \times U & & X \times \mathfrak{D} & & \Sigma \times U & & X \times \mathfrak{D} & & & & \Sigma \times U & & X \times \mathfrak{D} \end{array}$$

Similarly as in a substituted acceptance game for a usual non-deterministic or alternating automaton [Rib15], P chooses input letters and O chooses tree directions.

We now equip $\Sigma \vdash \mathcal{A}(M)$ with a winning condition $\mathcal{W}_{\mathcal{A}(M)} \subseteq ((\Sigma \times U) \cdot (X \times \mathfrak{D}))^{\omega}$. Each infinite play $\chi = ((\mathtt{a}_k, u_k) \cdot (x_k, d_k))_k \in ((\Sigma \times U) \cdot (X \times \mathfrak{D}))^{\omega}$ generates an infinite sequence of states $(q_k)_k \in Q^{\omega}_{\mathcal{A}}$ as follows. We let $q_0 := q^{\imath}_{\mathcal{A}}$ and

$$\begin{array}{lccl} & q_{k+1} & := & \partial_{\mathcal{A}}(q_k\,,\, \mathtt{b}_k\,,\, u_k\,,\, x_k\,,\, d_k) \\ \text{where} & \mathtt{b}_k & := & M(\mathtt{a}_0 \cdot \ldots \cdot \mathtt{a}_k\,,\, d_0 \cdot \ldots \cdot d_{k-1}) \end{array}$$

Then χ is winning (*i.e.* $\chi \in \mathcal{W}_{\mathcal{A}(M)}$) iff $(q_k)_k$ is accepting (*i.e.* iff $(q_k)_k \in \Omega_{\mathcal{A}}$).

Similarly as for usual alternating automata, acceptance for uniform tree automata can be defined via substituted acceptance games (see Rem. 5.1.2).

Definition 7.1.1. *Consider a uniform automaton $\mathcal{A}$ over Σ.*

(i) $\mathcal{A}$ accepts *the tree $T : \mathfrak{D}^* \to \Sigma$ if there is a winning* P*-strategy in $\mathbf{1} \vdash \mathcal{A}(\dot{T})$.*

(ii) Let $\mathcal{L}(\mathcal{A}) \subseteq \Sigma^{\mathfrak{D}^*}$, the language *of $\mathcal{A}$, be the set of trees accepted by $\mathcal{A}$.*

7.1.2. Uniform Linear Synchronous Arrow Games. The main idea is that a uniform automaton $\mathcal{A} : \Sigma$ as in (7.1) generates a $\mathsf{DialZ}(\Sigma)$ object (U, X). It then follows from §6.4 that the indexed category $\mathsf{DialZ}(-)$ contains as full subcategories the analogue of the indexed category of synchronous arrow games (§5.3).

It remains to handle winning. We write

$$(-)^{\uparrow} \quad : \quad \mathsf{DialZ}(\Sigma) \quad \longrightarrow \quad \mathbf{DZ}$$

for the canonical lifting functor obtained from the distributive law (6.8). This functor takes a total zig-zag strategy

$$\sigma \quad : \quad \Sigma \otimes A \quad \multimap_{\mathbf{DZ}} \quad B \otimes \mathfrak{D}$$

to a total zig-zag strategy

$$\sigma^{\uparrow} \quad : \quad \Sigma \otimes (A \otimes \mathfrak{D}) \quad \multimap_{\mathbf{DZ}} \quad \Sigma \otimes (B \otimes \mathfrak{D})$$

Definition 7.1.2 (The Category $\mathsf{DialAut}(\Sigma)$)**.** *Fix an alphabet* Σ*.*

- *The objects of the category* $\mathsf{DialAut}(\Sigma)$ *are tuples* $(U, X, \mathcal{W}_A)$ *where* U *and* X *are non-empty sets and where* $\mathcal{W}_A \subseteq ((\Sigma \times U) \cdot (X \times \mathfrak{D}))^{\omega}$*.*
- *The* $\mathsf{DialAut}(\Sigma)$ *morphisms from* $(U, X, \mathcal{W}_A)$ *to* $(V, Y, \mathcal{W}_B)$ *are total zig-zag strategies*

$$\sigma \quad : \quad \Sigma \otimes (U, X) \quad \multimap_{\mathbf{DZ}} \quad (V, Y) \otimes \mathfrak{D}$$

whose lift $\sigma^{\uparrow}$ *are winning strategies*

$$\sigma^{\uparrow} \quad : \quad (\Sigma \times U, X \times \mathfrak{D}, \mathcal{W}_A) \quad \multimap_{\mathbf{DZ}^{\mathrm{W}}} \quad (\Sigma \times V, Y \times \mathfrak{D}, \mathcal{W}_B)$$

Note that a $\mathsf{DialAut}(\Sigma)$ morphism from $(U, X, \mathcal{W}_A)$ to $(V, Y, \mathcal{W}_B)$ is in particular a $\mathsf{DialZ}(\Sigma)$ morphism from (U, X) to (V, Y). In order to equip the categories $\mathsf{DialAut}(-)$ with a (strict) indexed structure, we simply extend the indexed structure of $\mathsf{DialZ}(-)$ with winning. It is easy (but tedious) to do this directly "by hand" as in [Rib18], while a (better) abstract treatment is provided in [Pra19].

Example 7.1.3. *An automaton* $\mathcal{A} : \Gamma$ *as in (7.1) together with a tree morphism* $M \in \mathbf{T}[\Sigma, \Gamma]$ *generates a* $\mathsf{DialAut}(\Sigma)$ *object*

$$\Sigma \vdash \mathcal{A}(M) \quad := \quad (U, X, \mathcal{W}_{\mathcal{A}(M)})$$

where $\mathcal{W}_{\mathcal{A}(M)}$ *is defined as in §7.1.1.*

7.1.3. Substitution and Language Inclusion. We now check that $\mathsf{DialAut}(-)$ is correct w.r.t. language inclusion. First, consider substituted acceptance games $\Sigma \vdash \mathcal{A}(M)$ and $\Sigma \vdash \mathcal{B}(N)$ in the sense of §7.1.1. Following Ex. 7.1.3, We thus obtain $\mathsf{DialAut}(\Sigma)$ objects, that we still write $\Sigma \vdash \mathcal{A}(M)$ and $\Sigma \vdash \mathcal{B}(N)$ (or simply $\mathcal{A}(M)$ and $\mathcal{B}(N)$). Now, the indexed structure of $\mathsf{DialAut}(-)$ gives, from

$$\sigma \quad : \quad \mathcal{A}(M) \quad \multimap_{\mathsf{DialAut}(\Sigma)} \quad \mathcal{B}(N) \qquad \text{and} \qquad L \in \mathbf{T}[\Gamma, \Sigma]$$

a morphism

$$L^{\star}(\sigma) \quad : \quad \mathcal{A}(M \circ L) \quad \multimap_{\mathsf{DialAut}(\Gamma)} \quad \mathcal{B}(N \circ L)$$

where $(-)^\star$ is the substitution functor of $\mathsf{DialAut}$. Hence, $\mathsf{DialAut}$ interprets all instances of the (Subst) rule (4.7) of the form

$$\frac{M \ ; \ \mathcal{A} \vdash \mathcal{B}}{M \circ L \ ; \ \mathcal{A} \vdash \mathcal{B}} \quad \text{(where } M \in \mathbf{T}[\Sigma, \Delta] \text{ and } L \in \mathbf{T}[\Gamma, \Sigma])$$

In particular, given $\mathcal{A}, \mathcal{B} : \Sigma$, for all Σ-labeled tree T (and skipping the $\dot{T}$ notation of §4.2.(b)) we have

$$\frac{\Sigma \ ; \ \mathcal{A} \vdash \mathcal{B}}{T \ ; \ \mathcal{A} \vdash \mathcal{B}}$$

Assume given $\sigma : \mathcal{A} \multimap \mathcal{B}$. If $T \in \mathcal{L}(\mathcal{A})$, then there is some $\tau : \mathbf{I_1} \multimap \mathcal{A}(T)$. It follows that we obtain $T^\star(\sigma) \circ \tau : \mathbf{I_1} \multimap \mathcal{B}(T)$, which implies $T \in \mathcal{L}(\mathcal{B})$. In other words, $\sigma : \mathcal{A} \multimap \mathcal{B}$ and T induce a function

$$\big(\mathbf{I} \multimap \mathcal{A}(T)\big) \quad \longrightarrow \quad \big(\mathbf{I} \multimap \mathcal{B}(T)\big), \quad \tau \ \longmapsto \ T^\star(\sigma) \circ \tau$$

and we have shown:

Proposition 7.1.4. *If* P *has a winning strategy in* $\Sigma \vdash \mathcal{A} \multimap \mathcal{B}$*, then* $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$*.*

7.2. Connectives and Deduction Rules on Uniform Tree Automata

This Section, together with §7.3 and §7.4, presents a basic set of connectives on uniform automata, namely the linear connectives $\otimes$ and $\multimap$, falsity $\perp\!\!\!\!\!\!\perp$, universal and existential quantifications, as well as an exponential modality $!(-)$ induced by the *Simulation Theorem* [EJ91, MS95] (see also §2.2). The corresponding deduction rules are presented in Fig. 7.1.

The fibrewise symmetric monoidal closed structure of $\mathsf{DialZ}(-)$, as well as the existential and universal quantifications (see Chap. 6) directly lifts to $\mathsf{DialAut}$ and is easily reflected in uniform tree automata. We present the constructions for uniform automata in §7.2.1 (and refer to [Rib18] for details on $\mathsf{DialAut}$). Then, a particular attention to falsity is given in §7.2.2, where we devise an automaton $\perp\!\!\!\!\!\!\perp$ such that (thanks to a usual determinacy result) $\mathcal{A} \multimap \perp\!\!\!\!\!\!\perp$ is a (linear) complement of $\mathcal{A}$ in the usual sense. Quantifiers are briefly discussed in §7.2.3. We defer important basic facts on non-deterministic automata to §7.3, and briefly discuss in §7.4 the exponential construction $!(-)$ and some of its basic consequences.

The general correctness result is the following (where $\otimes_{\mathsf{DA}}$ denotes the fibrewise monoidal product of $\mathsf{DialAut}$). We refer to §6.5.2 for details on *finite-state* strategies.

Proposition 7.2.1 (Adequacy)**.** *Let* $M \in \mathbf{T}[\Sigma, \Gamma]$*. If the sequent* $M \ ; \ \mathcal{A}_1, \ldots, \mathcal{A}_n \vdash \mathcal{B}$ *is derivable using the rules of Fig. 7.1, then there is a winning finite-state winning* P*-strategy*

$$\sigma \quad : \quad \mathcal{A}_1(M) \otimes_{\mathsf{DA}} \cdots \otimes_{\mathsf{DA}} \mathcal{A}_n(M) \quad \multimap_{\mathsf{DialAut}(\Sigma)} \quad \mathcal{B}(M)$$

In particular, if $\mathcal{A} \vdash \mathcal{B}$ is derivable, then by combining Prop. 7.2.1 with Prop. 7.1.4, we obtain a strategy witnessing that $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$. Note also that strategies in Prop. 7.2.1 are obtained from derivations in a purely compositional way. Moreover, all the rules of Fig. 7.1 are compatible with cut-elimination (see Rem. 7.2.3).

$$\frac{M \;;\; \overline{\mathcal{A}}, \mathcal{A}, \mathcal{B}, \overline{\mathcal{C}} \vdash \mathcal{C}}{M \;;\; \overline{\mathcal{A}}, \mathcal{B}, \mathcal{A}, \overline{\mathcal{C}} \vdash \mathcal{C}} \tag{Exchange}$$

$$\frac{M \;;\; \overline{\mathcal{A}} \vdash \mathcal{A}}{M \circ M' \;;\; \overline{\mathcal{A}} \vdash \mathcal{A}} \tag{Subst}$$

$$\frac{M \;;\; \overline{\mathcal{A}} \vdash \mathcal{A} \qquad M \;;\; \overline{\mathcal{B}}, \mathcal{A}, \overline{\mathcal{C}} \vdash \mathcal{C}}{M \;;\; \overline{\mathcal{B}}, \overline{\mathcal{A}}, \overline{\mathcal{C}} \vdash \mathcal{C}} \tag{Cut}$$

$$\frac{}{M \;;\; \mathcal{A} \vdash \mathcal{A}} \tag{Axiom}$$

$$\frac{M \;;\; \overline{\mathcal{A}}, \mathcal{A}, \mathcal{B}, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \;;\; \overline{\mathcal{A}}, \mathcal{A} \otimes \mathcal{B}, \overline{\mathcal{B}} \vdash \mathcal{C}} \tag{Left $\otimes$}$$

$$\frac{M \;;\; \overline{\mathcal{A}} \vdash \mathcal{A} \qquad M \;;\; \overline{\mathcal{B}} \vdash \mathcal{B}}{M \;;\; \overline{\mathcal{A}}, \overline{\mathcal{B}} \vdash \mathcal{A} \otimes \mathcal{B}} \tag{Right $\otimes$}$$

$$\frac{M \;;\; \overline{\mathcal{A}}, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \;;\; \overline{\mathcal{A}}, \mathbf{I}, \overline{\mathcal{B}} \vdash \mathcal{C}} \tag{Left $\mathbf{I}$}$$

$$\frac{}{M \;;\; \vdash \mathbf{I}} \tag{Right $\mathbf{I}$}$$

$$\frac{M \;;\; \overline{\mathcal{A}} \vdash \mathcal{A} \qquad M \;;\; \overline{\mathcal{B}}, \mathcal{B}, \overline{\mathcal{C}} \vdash \mathcal{C}}{M \;;\; \overline{\mathcal{B}}, \overline{\mathcal{A}}, \mathcal{A} \multimap \mathcal{B}, \overline{\mathcal{C}} \vdash \mathcal{C}} \tag{Left $\multimap$}$$

$$\frac{M \;;\; \overline{\mathcal{A}}, \mathcal{B} \vdash \mathcal{C}}{M \;;\; \overline{\mathcal{A}} \vdash \mathcal{B} \multimap \mathcal{C}} \tag{Right $\multimap$}$$

$$\frac{M \;;\; \overline{\mathcal{A}}, \mathcal{A}, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \;;\; \overline{\mathcal{A}}, !\mathcal{A}, \overline{\mathcal{B}} \vdash \mathcal{C}} \tag{Dereliction}$$

$$\frac{M \;;\; \overline{\mathcal{N}} \vdash \mathcal{A}}{M \;;\; \overline{\mathcal{N}} \vdash !\mathcal{A}} \tag{Promotion}$$

$$\frac{M \;;\; \overline{\mathcal{A}}, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \;;\; \overline{\mathcal{A}}, \mathcal{N}, \overline{\mathcal{B}} \vdash \mathcal{C}} \tag{Weak$_{\mathrm{ND}}$}$$

$$\frac{M \;;\; \overline{\mathcal{A}}, \mathcal{N}, \mathcal{N}, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \;;\; \overline{\mathcal{A}}, \mathcal{N}, \overline{\mathcal{B}} \vdash \mathcal{C}} \tag{Contr$_{\mathrm{ND}}$}$$

$$\frac{M \times \mathrm{Id}_\Gamma \;;\; \overline{\mathcal{A}[\pi]}, \mathcal{B} \vdash \mathcal{A}[\pi]}{M \;;\; \overline{\mathcal{A}}, \exists_\Gamma \mathcal{B} \vdash \mathcal{A}} \tag{Left $\exists$}$$

$$\frac{M \times N \;;\; \overline{\mathcal{A}} \vdash \mathcal{A}}{M \times N \;;\; \overline{\mathcal{A}} \vdash (\exists_\Gamma \mathcal{A})[\pi]} \tag{Right $\exists$}$$

$$\frac{M \times N \;;\; \overline{\mathcal{A}}, \mathcal{B} \vdash \mathcal{A}}{M \times N \;;\; \overline{\mathcal{A}}, (\forall_\Gamma \mathcal{B})[\pi] \vdash \mathcal{A}} \tag{Left $\forall$}$$

$$\frac{M \times \mathrm{Id}_\Gamma \;;\; \overline{\mathcal{A}[\pi]} \vdash \mathcal{A}}{M \;;\; \overline{\mathcal{A}} \vdash \forall_\Gamma A} \tag{Right $\forall$}$$

Figure 7.1.: Deduction rules on uniform automata, where M, M' are composable, $\mathcal{N}, \overline{\mathcal{N}}$ are non-deterministic, and where the weakening functor $(-)[\pi]$ takes automata over Σ to automata over $\Sigma \times \Gamma$.

7.2.1. Monoidal Closed Structure. The (fibrewise) symmetric monoidal closed structure of DialZ (and DialAut) induces the corresponding connectives on uniform automata.

Definition 7.2.2 (Monoidal Product and Linear Implication on Uniform Automata). *Consider uniform automata*

$$\begin{array}{rcl} \mathcal{A} & = & (Q_\mathcal{A}\,,\, q^\imath_\mathcal{A}\,,\, U\,,\, X\,,\, \partial_\mathcal{A}\,,\, \Omega_\mathcal{A}) \\ \mathcal{B} & = & (Q_\mathcal{B}\,,\, q^\imath_\mathcal{B}\,,\, V\,,\, Y\,,\, \partial_\mathcal{B}\,,\, \Omega_\mathcal{B}) \end{array}$$

so that

$$\begin{array}{lrclclcl} & \partial_\mathcal{A} & : & Q_\mathcal{A}\times\Sigma & \longrightarrow & U\times X & \longrightarrow & (\mathfrak{D}\longrightarrow Q_\mathcal{A}) \\ \textit{and} & \partial_\mathcal{B} & : & Q_\mathcal{B}\times\Sigma & \longrightarrow & V\times Y & \longrightarrow & (\mathfrak{D}\longrightarrow Q_\mathcal{B}) \end{array}$$

- *We let $\mathcal{A}\otimes\mathcal{B}$ be the automaton over Σ defined as*

$$\mathcal{A}\otimes\mathcal{B} \quad := \quad (Q_\mathcal{A}\times Q_\mathcal{B}\,,\, (q^\imath_\mathcal{A}, q^\imath_\mathcal{B})\,,\, U\times V\,,\, X\times Y\,,\, \partial_{\mathcal{A}\otimes\mathcal{B}}\,,\, \Omega_{\mathcal{A}\otimes\mathcal{B}})$$

with transitions given by

$$\partial_{\mathcal{A}\otimes\mathcal{B}}\big((q_\mathcal{A}, q_\mathcal{B})\,,\, \mathtt{a}\,,\, (u,v)\,,\, (x,y)\,,\, d\big) \quad := \quad (q'_\mathcal{A}\,,\, q'_\mathcal{B})$$

where

$$q'_\mathcal{A} \;:=\; \partial_\mathcal{A}(q_\mathcal{A}\,,\, \mathtt{a}\,,\, u\,,\, x\,,\, d) \qquad \textit{and} \qquad q'_\mathcal{B} \;:=\; \partial_\mathcal{B}(q_\mathcal{B}\,,\, \mathtt{a}\,,\, v\,,\, y\,,\, d)$$

and with acceptance condition given by

$$\big((q_n, q'_n)\big)_n \in \Omega_{\mathcal{A}\otimes\mathcal{B}} \qquad \textit{iff} \qquad \big((q_n)_n \in \Omega_\mathcal{A} \textit{ and } (q'_n)_n \in \Omega_\mathcal{B}\big) \tag{7.3}$$

- *We let $(\mathcal{A}\multimap\mathcal{B})$ be the automaton over Σ defined as*

$$(\mathcal{A}\multimap\mathcal{B}) \quad := \quad \Big(Q_\mathcal{A}\times Q_\mathcal{B}\,,\, (q^\imath_\mathcal{A}, q^\imath_\mathcal{B})\,,\, V^U\times X^{U\times Y\times\mathfrak{D}}\,,\, U\times Y\,,\, \partial_{\mathcal{A}\multimap\mathcal{B}}\,,\, \Omega_{\mathcal{A}\multimap\mathcal{B}}\Big)$$

with transitions given by

$$\partial_{\mathcal{A}\multimap\mathcal{B}}\big((q_\mathcal{A}, q_\mathcal{B})\,,\, \mathtt{a}\,,\, (f,F)\,,\, (u,y)\,,\, d\big) \quad := \quad (q'_\mathcal{A}\,,\, q'_\mathcal{B})$$

where

$$q'_\mathcal{A} \;=\; \partial_\mathcal{A}\big(q_\mathcal{A}\,,\, \mathtt{a}\,,\, u\,,\, F(u,y,d)\,,\, d\big) \qquad \textit{and} \qquad q'_\mathcal{B} \;=\; \partial_\mathcal{B}\big(q_\mathcal{B}\,,\, \mathtt{a}\,,\, f(u)\,,\, y\,,\, d\big)$$

and with acceptance condition given by

$$\big((q_n, q'_n)\big)_n \in \Omega_{\mathcal{A}\multimap\mathcal{B}} \qquad \textit{iff} \qquad \big((q_n)_n \in \Omega_\mathcal{A} \;\Rightarrow\; (q'_n)_n \in \Omega_\mathcal{B}\big)$$

Note that $\Omega_{\mathcal{A}\otimes\mathcal{B}}$ as well as $\Omega_{\mathcal{A}\multimap\mathcal{B}}$ are ω-regular since $\Omega_\mathcal{A}$ and $\Omega_\mathcal{B}$ are both assumed to be ω-regular. Note also that $\mathcal{A}\otimes\mathcal{B}$ is non-deterministic (resp. universal, deterministic) if both $\mathcal{A}$ and $\mathcal{B}$ are non-deterministic (resp. universal, deterministic).

Remark 7.2.3 (On Cut-Elimination). *Since we have monoidal closed categories, the interpretation of derivations as strategies for the rules of Fig. 7.1 (but for* (Promotion) *and* (Dereliction)*) is compatible with cut-elimination, in the sense that if a derivation $\mathscr{D}'$ is obtained from a derivation $\mathscr{D}$ by applying the proof transformation steps described in e.g. [Mel09, §3.3], then $\mathscr{D}$ and $\mathscr{D}'$ are interpreted by the same strategy. This in particular applies to the following two derivations:*

$$\dfrac{\dfrac{\dfrac{\mathscr{D}_1}{\mathcal{A}\vdash\mathcal{B}}}{\mathbf{I}\vdash\mathcal{A}\multimap\mathcal{B}} \qquad \dfrac{\dfrac{\mathscr{D}_2}{\mathbf{I}\vdash\mathcal{A}} \qquad \overline{\mathcal{B}\vdash\mathcal{B}}}{\mathcal{A}\multimap\mathcal{B}\vdash\mathcal{B}}}{\mathbf{I}\vdash\mathcal{B}} \qquad\qquad \dfrac{\begin{array}{c}\vdots\\ \mathscr{D}_1[\mathscr{D}_2/\mathcal{A}]\end{array}}{\mathbf{I}\vdash\mathcal{B}}$$

Example 7.2.4. *Proposition 7.2.1 yields a winning* P*-strategy in*

$$\mathcal{B} \otimes \mathcal{B} \otimes (\mathcal{B} \multimap \mathcal{A}) \quad \multimap \quad \mathcal{A} \otimes \mathcal{B}$$

obtained from the proof tree

$$\dfrac{\dfrac{\dfrac{\overline{\mathcal{B} \vdash \mathcal{B}} \qquad \overline{\mathcal{A} \vdash \mathcal{A}}}{\mathcal{B}, \mathcal{B} \multimap \mathcal{A} \vdash \mathcal{A}} \qquad \overline{\mathcal{B} \vdash \mathcal{B}}}{\mathcal{B}, (\mathcal{B} \multimap \mathcal{A}), \mathcal{B} \vdash \mathcal{A} \otimes \mathcal{B}}}{\mathcal{B}, \mathcal{B}, (\mathcal{B} \multimap \mathcal{A}) \vdash \mathcal{A} \otimes \mathcal{B}}$$

Note that in Fig. 7.1 we omitted the *weakening* and *contraction* rules (4.14):

$$(\textsc{Weak}) \quad \dfrac{M \ ; \ \overline{\mathcal{A}}, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \ ; \ \overline{\mathcal{A}}, \mathcal{A}, \overline{\mathcal{B}} \vdash \mathcal{C}} \qquad\qquad \dfrac{M \ ; \ \overline{\mathcal{A}}, \mathcal{A}, \mathcal{A}, \overline{\mathcal{B}} \vdash \mathcal{C}}{M \ ; \ \overline{\mathcal{A}}, \mathcal{A}, \overline{\mathcal{B}} \vdash \mathcal{C}} \quad (\textsc{Contr})$$

Similarly as with usual automata, the contraction rule can be interpreted on *non-deterministic* uniform automata but not on general uniform automata. This rule amounts to providing winning P-strategies in the game

$$\mathcal{A} \quad \multimap \quad \mathcal{A} \otimes \mathcal{A} \tag{7.4}$$

If $\mathcal{A}$ is non-deterministic (say with P-moves U), then a P-strategy in (7.4) simply takes an O-move $u \in U$ in component $\mathcal{A}$ to the pair $(u, u) \in U \times U$ in component $\mathcal{A} \otimes \mathcal{A}$. Note that such strategy may not exist when $\mathcal{A}$ is a general uniform automaton, that is when it is equipped with a set of O-moves $X \not\simeq \mathbf{1}$, since O can play two different $(x, x') \in X \times X$ in the component $\mathcal{A} \otimes \mathcal{A}$, that P may not be able to merge into a single $x'' \in X$ in the left component $\mathcal{A}$.

On the other hand, the weakening rule, which asks for a winning P-strategy in

$$\mathcal{A} \quad \multimap \quad \mathbf{I}$$

can always be realized (since we required the set of P and O-moves to be always non-empty), but in a non-canonical way for general uniform automata. More generally, given $\mathcal{A}$ and $\mathcal{B}$ over the same input alphabet, there is always a winning P-strategy in

$$\mathcal{A} \otimes \mathcal{B} \quad \multimap \quad \mathcal{A} \tag{7.5}$$

Assuming $\mathcal{A}$ and $\mathcal{B}$ are as in Def. 7.2.2, such a strategy takes $(u, v) \in U \times V$ to $u \in U$ and takes $x \in X$ to $(x, y) \in X \times Y$, where y is an arbitrarily chosen element of Y.

We shall come back on the connection between non-deterministic automata, the interpretation of the (Weak) and (Contr) rules and IMELL in §7.3.

Example 7.2.5. *Proposition 7.2.1 actually holds for any extension of the deduction system of Fig. 7.1 with realizable rules, that is with rules*

$$\overline{\mathcal{A} \vdash \mathcal{B}}$$

such that there is a winning P*-strategy in* $\mathcal{A} \multimap \mathcal{B}$*. In particular:*

(i) We can extend the system with the following generalization of (7.5):

$$\overline{\mathcal{A}_1 \, , \, \ldots \, , \, \mathcal{A}_n \vdash \mathcal{A}_i}$$

We thus get

$$
\dfrac{\dfrac{\dfrac{}{\mathcal{A} \vdash \mathcal{A}} \qquad \dfrac{}{\mathcal{B} \vdash \mathcal{B}}}{\mathcal{A}, \mathcal{B} \vdash \mathcal{A} \otimes \mathcal{B}} \qquad \dfrac{}{\mathcal{A} \otimes \mathcal{B} \vdash \mathcal{A}}}{\dfrac{\mathcal{A}, \mathcal{B} \vdash \mathcal{A}}{\mathcal{A} \vdash \mathcal{B} \multimap \mathcal{A}}}
$$

So there is a winning P*-strategy on*

$$
\mathcal{A} \quad \multimap \quad (\mathcal{B} \multimap \mathcal{A})
$$

and by Prop. 7.1.4 we have

$$
\mathcal{L}(\mathcal{A}) \quad \subseteq \quad \mathcal{L}(\mathcal{B} \multimap \mathcal{A})
$$

(ii) For $\mathcal{B}$ non-deterministic, we can extend the system with the following generalizations of (7.4):

$$
\dfrac{}{\mathcal{B} \vdash \mathcal{B} \otimes \ldots \otimes \mathcal{B}}
$$

Continuing Ex. 7.2.4 with $\mathcal{B}$ non-deterministic, we thus have

$$
\dfrac{\dfrac{}{\mathcal{B} \vdash \mathcal{B} \otimes \mathcal{B}} \qquad \dfrac{\dfrac{\vdots}{\mathcal{B}, \mathcal{B}, (\mathcal{B} \multimap \mathcal{A}) \vdash \mathcal{A} \otimes \mathcal{B}}}{\mathcal{B} \otimes \mathcal{B}, (\mathcal{B} \multimap \mathcal{A}) \vdash \mathcal{A} \otimes \mathcal{B}}}{\mathcal{B}, (\mathcal{B} \multimap \mathcal{A}) \vdash \mathcal{A} \otimes \mathcal{B}}
$$

Finally, we note that the monoidal structure together with (7.5) imply that $\otimes$ indeed implements a conjunction on automata.

Proposition 7.2.6. *Given $\mathcal{A}, \mathcal{B} : \Sigma$, we have $\mathcal{L}(\mathcal{A} \otimes \mathcal{B}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B})$.*

7.2.2. Falsity and Complementation. We have already seen in §1.4 that usual alternating automata are equipped with a complementation construction $(-)^{\perp\!\!\!\perp}$ linear in the number of states. Using the monoidal closed structure a similar construction can be done with uniform automata, but now with the expected functoriality.

Definition 7.2.7 (Falsity Automaton)**.** *For each alphabet Σ, the* falsity *(non-deterministic) uniform automaton $\bot$ over Σ is*

$$
\bot \quad := \quad (\mathbb{B}\,,\, \mathbb{f}\,,\, \mathfrak{D}\,,\, \mathbf{1}\,,\, \partial_{\bot}\,,\, \Omega_{\bot})
$$

where $\Omega_{\bot} := \mathbb{B}^* \cdot \mathbb{t}^\omega$ and where

$$
\partial_{\bot}(\mathbb{b}\,,\, \_\,,\, d'\,,\, \bullet\,,\, d) \quad := \quad \begin{cases} \mathbb{f} & \text{if } \mathbb{b} = \mathbb{f} \text{ and } d = d' \\ \mathbb{t} & \text{otherwise} \end{cases}
$$

Note that in the game $\Sigma \vdash \bot$, O looses as soon as it does not play the same tree direction as proposed by P. On the other hand, $\bot$ accepts no tree since in an acceptance game $\bot(T)$, O can always play the same d as P.

Thanks to the determinacy of ω-regular games (see e.g. [Tho97, PP04]), we get:

Proposition 7.2.8. *Given $\mathcal{A} : \Sigma$, we have $\mathcal{L}(\mathcal{A} \multimap \bot) = \Sigma^{\mathfrak{D}^*} \setminus \mathcal{L}(\mathcal{A})$.*

7.2.3. Quantifications. Quantifications in DialZ (and in $\mathsf{DialAut}$) induce quantifications on uniform automata.

Definition 7.2.9. *Given $\mathcal{A} : \Sigma \times \Gamma$ with set of P-moves U and set of O-moves X, let*

$$\begin{array}{lll} (\exists_\Gamma \mathcal{A} : \Sigma) & := & (Q_\mathcal{A}\,,\, q^\imath_\mathcal{A}\,,\, \Gamma \times U\,,\, X\,,\, \partial_{\exists_\Gamma \mathcal{A}}\,,\, \Omega_\mathcal{A}) \\ (\forall_\Gamma \mathcal{A} : \Sigma) & := & (Q_\mathcal{A}\,,\, q^\imath_\mathcal{A}\,,\, U^\Gamma\,,\, \Gamma \times X\,,\, \partial_{\forall_\Gamma \mathcal{A}}\,,\, \Omega_\mathcal{A}) \end{array}$$

where

$$\begin{array}{llll} & \partial_{\exists_\Gamma \mathcal{A}}(q,\, \mathtt{a},\, (\mathtt{b}, u),\, x,\, d) & := & \partial_\mathcal{A}(q,\, (\mathtt{a}, \mathtt{b}),\, u,\, x,\, d) \\ \textit{and} & \partial_{\forall_\Gamma \mathcal{A}}(q,\, \mathtt{a},\, f,\, (\mathtt{b}, x),\, d) & := & \partial_\mathcal{A}(q,\, (\mathtt{a}, \mathtt{b}),\, f(\mathtt{b}),\, x,\, d) \end{array}$$

Quantifications on automata give an $\exists\forall$-structure which is reminiscent from Gödel's *Dialectica* interpretation (see e.g. [AF98, Koh08]). This has been elaborated in [PR19] for the case of ω-words (see also §8.2.6 and [Pra19]).

Example 7.2.10. *Given $\mathcal{A} : \Sigma$ with set of P-moves U and set of O-moves X, let $\mathcal{D}$ be the deterministic automaton*

$$(\mathcal{D} : \Sigma \times U \times X) \quad := \quad (Q_\mathcal{A},\, q^\imath_\mathcal{A},\, \mathbf{1},\, \mathbf{1},\, \partial_\mathcal{D},\, \Omega_\mathcal{A})$$

whose transition function

$$\partial_\mathcal{D} \quad : \quad Q_\mathcal{A} \times (\Sigma \times U \times X) \quad \longrightarrow \quad \mathfrak{D} \quad \longrightarrow \quad Q_\mathcal{A}$$

is obtained from $\partial_\mathcal{A}$ in the obvious way. In $\mathsf{DialAut}_\Sigma$ we have $\mathcal{A} \simeq \exists_U \forall_X \mathcal{D}$.

Let us now discuss the connection between quantifications on automata and in DialZ (and $\mathsf{DialAut}$). First, given $(\mathcal{A} : \Sigma \times \Gamma)$, we have, as objects of $\mathsf{DialZ}(\Sigma)$ (and $\mathsf{DialAut}(\Sigma)$)

$$(\Sigma \vdash \coprod_{\Sigma,\Gamma} \mathcal{A}) \quad = \quad (\Sigma \vdash \exists_\Sigma \mathcal{A}) \qquad \text{and} \qquad (\Sigma \vdash \prod_{\Sigma,\Gamma} \mathcal{A}) \quad = \quad (\Sigma \vdash \forall_\Sigma \mathcal{A})$$

It then follows that the Beck-Chevalley conditions in DialZ (and $\mathsf{DialAut}$) imply

$$\begin{array}{lllll} \coprod_{\Sigma,\Gamma} \mathcal{A}(M \times \mathrm{Id}_\Gamma) & = & M^\star(\coprod_{\Delta,\Gamma} \mathcal{A}) & = & (\exists_\Gamma \mathcal{A})(M) \\ \prod_{\Sigma,\Gamma} \mathcal{A}(M \times \mathrm{Id}_\Gamma) & = & M^\star(\prod_{\Delta,\Gamma} \mathcal{A}) & = & (\forall_\Gamma \mathcal{A})(M) \end{array}$$

Thanks to the adjunctions $\coprod \dashv \pi^\star \dashv \prod$ in DialZ (and $\mathsf{DialAut}$), we then have

$$\begin{array}{lll} \Sigma \vdash (\exists_\Gamma \mathcal{A})(M) \multimap \mathcal{B}(N) & \simeq & \Sigma \times \Gamma \vdash \mathcal{A}(M \times \mathrm{Id}_\Gamma) \multimap \mathcal{B}(N \circ [\pi_\Sigma]) \\ \Sigma \vdash \mathcal{B}(N) \multimap (\forall_\Gamma \mathcal{A})(M) & \simeq & \Sigma \times \Gamma \vdash \mathcal{B}(N \circ [\pi_\Sigma]) \multimap \mathcal{A}(M \times \mathrm{Id}_\Gamma) \end{array} \tag{7.6}$$

It follows that P has winning strategies in

$$\Sigma \times \Gamma \vdash (\forall_\Gamma \mathcal{A})[\pi_\Sigma] \multimap \mathcal{A} \qquad \text{and} \qquad \Sigma \times \Gamma \vdash \mathcal{A} \multimap (\exists_\Gamma \mathcal{A})[\pi_\Sigma] \tag{7.7}$$

We thus get the following corollary to Prop. 6.4.4.

Corollary 7.2.11. *Given uniform automata $\mathcal{A}, \mathcal{B} : \Sigma$, the game $\Sigma \vdash \mathcal{A} \multimap \mathcal{B}$ is equivalent to a regular game on a finite graph. It is therefore decidable whether there exists a winning P-strategy on $\Sigma \vdash \mathcal{A} \multimap \mathcal{B}$, and if there exists such a winning P-strategy, then there exists a finite-state one, which is moreover effectively computable from $\mathcal{A}$ and $\mathcal{B}$.*

Proof. Note that P has a winning strategy in $\Sigma \vdash \mathcal{A} \multimap \mathcal{B}$ iff it has a winning strategy in $\mathbf{1} \vdash \mathbf{I_1} \multimap \forall_\Sigma(\mathcal{A} \multimap \mathcal{B})$. But since in that game O can only play $\bullet$ in the component $\mathbf{I_1}$, it is equivalent to the acceptance game of the automaton $\forall_\Sigma(\mathcal{A} \multimap \mathcal{B}) : \mathbf{1}$ on the unique tree $\mathbf{1} : \mathfrak{D}^* \to \mathbf{1}$. Then conclude as in [Tho97, Ex. 6.12], using Büchi-Landweber Theorem [BL69] (see also [Tho97, Thm. 6.18]). ☐

We also get from (7.7) that existential quantifications are complete in the following sense:

Corollary 7.2.12. *Given $\mathcal{A} : \Sigma \times \Gamma$, we have $\pi_\Gamma(\mathcal{L}(\mathcal{A})) \subseteq \mathcal{L}(\exists_\Gamma \mathcal{A})$.*

The converse inclusion (the correctness of existential quantifications w.r.t. the standard semantics) only holds for *non-deterministic* automata, and is detailed in §7.3. Dually, it follows from (7.7) that universal quantifications are correct w.r.t. the standard semantics, but they are complete only on *universal* automata (see Def. 7.0.1).

Corollary 7.2.13. *Given $\mathcal{A} : \Sigma \times \Gamma$, if $T \in \mathcal{L}(\forall_\Gamma \mathcal{A})$, then for all Γ-labeled tree T' we have $\langle T, T' \rangle \in \mathcal{L}(\mathcal{A})$.*

Remark 7.2.14 (On General Quantifications)**.** *For general uniform automata, existential and universal quantification do not have their expected standard semantics. On the other hand, their categorical properties (see also §6.4.3) imply that they satisfy the expected logical rules (see Fig. 7.1 and (7.7)).*

While this fact may seem surprising, it actually just indicates that automata are living in a non-standard logical universe, differing from usual classical logic. As we shall see in Chap. 8 (and elaborate in §8.2.5 for the particular case of ω-words), a sound basis for this logic is provided by Linear Logic [Gir87].

Example 7.2.15. *Continuing Ex. 7.2.5, we can extend the deduction system with the rule*

$$\frac{\mathcal{L}(\mathcal{A} : \mathbf{1}) \neq \emptyset}{\vdash \mathcal{A}}$$

This rule actually subsumes Ex. 7.2.5. Indeed, following the same reasoning as for Cor. 7.2.11, assuming that

$$\Sigma \ ; \ \mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_n \vdash \mathcal{B}$$

is realizable we get (leaving implicit some structural and cut rules)

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{L}(\forall_\Sigma(\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_n \multimap \mathcal{B})) \neq \emptyset}{\mathbf{1} \ ; \ \vdash \forall_\Sigma(\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_n \multimap \mathcal{B})}}{\Sigma \ ; \ \vdash \mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_n \multimap \mathcal{B}}}{\Sigma \ ; \ \mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_n \vdash \mathcal{B}}}{\Sigma \ ; \ \mathcal{A}_1, \ldots, \mathcal{A}_n \vdash \mathcal{B}}$$

7.3. Non-Deterministic Automata

The key property of non-deterministic automata is that the monoidal product becomes Cartesian when restricted to these. We refer to [Rib18] for details, and just sketch here the main facts.

Consider a substituted acceptance game $\mathcal{N}(L)$ with $\mathcal{N}$ non-deterministic and with set of P-moves U. Hence, the underlying $\mathsf{DialZ}(\Sigma)$-object of $\mathcal{N}(L)$ is of the form (U, I) with $I \simeq \mathbf{1}$, and thus are commutative comonoids. This means that we have canonical realizers for

$$\mathcal{N}(L) \quad \multimap \quad \mathcal{N}(L) \otimes \mathcal{N}(L) \qquad \text{and} \qquad \mathcal{N}(L) \quad \multimap \quad \mathbf{I} \tag{7.8}$$

which, thanks to well-known results (see e.g. [Mel09, Cor. 18, §6.5]), implies that the monoidal structure of uniform automata is Cartesian on non-deterministic automata.

Let us now sketch some consequences of this.

7.3.1. Deduction Rules for Non-Deterministic Automata. Similarly as with usual (total) non-deterministic automata (see §4.4), the Cartesian structure on non-deterministic automata is the reason why we could take, in the deduction system of Fig. 7.1, the following the structural weakening and contraction rules:

$$(\text{Weak}_{\text{ND}})\ \frac{M\ ;\ \overline{\mathcal{A}},\overline{\mathcal{B}}\vdash\mathcal{C}}{M\ ;\ \overline{\mathcal{A}},\mathcal{N},\overline{\mathcal{B}}\vdash\mathcal{C}} \qquad \frac{M\ ;\ \overline{\mathcal{A}},\mathcal{N},\mathcal{N},\overline{\mathcal{B}}\vdash\mathcal{C}}{M\ ;\ \overline{\mathcal{A}},\mathcal{N},\overline{\mathcal{B}}\vdash\mathcal{C}}\ (\text{Contr}_{\text{ND}}) \tag{7.9}$$

where $\mathcal{N}$ is required to be non-deterministic (while $\overline{\mathcal{A}}$, $\overline{\mathcal{B}}$ and $\mathcal{C}$ can be arbitrary).

7.3.2. Existential Quantifications and Extraction. An other nice consequence of the Cartesian structure on non-deterministic automata is the fact that existential quantifications behave similarly as the usual *sum types* of Type Theory (see e.g. [Jac01, Chap. 10]). Consider a non-deterministic automaton $\mathcal{N} : \Sigma \times \Gamma$ with set of P-moves U, and let T be a Σ-labeled tree (so that $T : \mathfrak{D}^* \to \Sigma$). It directly follows from Prop. 6.2.2 that a winning P-strategy in $\mathbf{1} \vdash \mathbf{I} \multimap (\exists_\Gamma \mathcal{A})(\dot{T})$ is given by a function

$$\bigcup_{n\in\mathbb{N}} \mathfrak{D}^n \quad\longrightarrow\quad \Gamma \times U$$

hence by a pair of functions

$$\left(\bigcup_{n\in\mathbb{N}} \mathfrak{D}^n \quad\longrightarrow\quad \Gamma\right) \times \left(\bigcup_{n\in\mathbb{N}} \mathfrak{D}^n \quad\longrightarrow\quad U\right)$$

and therefore by a tree $T' : \mathfrak{D}^* \to \Gamma$ together with a winning P-strategy in $\mathbf{1} \vdash \mathbf{I} \multimap \mathcal{A}\langle \dot{T}, \dot{T}' \rangle$. We thus have shown

Proposition 7.3.1. *Given a non-deterministic automaton $\mathcal{N} : \Sigma \times \Gamma$, a winning* P*-strategy $\sigma : \mathbf{1} \multimap \exists_\Sigma \mathcal{N}$ is of the form $\sigma = \langle T, \tau \rangle$ where T is a Σ-labeled tree and τ is a winning* P*-strategy in $\mathbf{1} \multimap \mathcal{N}(T)$ (so in particular $T \in \mathcal{L}(\mathcal{N})$).*

In particular, we get the following fact, which completes Cor. 7.2.12 and mirrors the well-known situation with usual non-deterministic automata.

Corollary 7.3.2. *If $\mathcal{N} : \Sigma \times \Gamma$ is* non-deterministic *then $\mathcal{L}(\exists_\Gamma \mathcal{N}) = \pi_\Gamma(\mathcal{L}(\mathcal{N}))$.*

Moreover, it follows from Prop. 7.3.1 that our computational interpretation makes it possible to effectively extract witnesses from (interpretations of) proofs, in the sense of §1.6 and §1.7. Let $\mathcal{N} : \Sigma$ be non-deterministic with set of P-moves U, and consider a derivation $\mathscr{D}$ of the sequent

$$\mathbf{1}\ ;\ \vdash \exists_\Sigma \mathcal{N}$$

using the rules of Fig. 7.1. Ex. 7.2.5. Then adequacy (Prop. 7.2.1) gives a strategy

$$\sigma \quad : \quad \mathbf{I} \quad\multimap\quad \exists_\Sigma \mathcal{N}$$

(effectively computed by induction on $\mathscr{D}$), and which by Prop. 7.3.1 is of the form

$$\begin{array}{lrcl} & \langle T, \tau \rangle & : & \bigcup_{n\in\mathbb{N}} \mathfrak{D}^n \quad\longrightarrow\quad \Sigma \times U \\ \text{where} & \tau & : & \mathbf{I} \quad\multimap\quad \mathcal{N}(T) \end{array}$$

7.3.3. Effective Realizers from Witnesses of Non-Emptiness. Similarly as with usual non-deterministic automata (see e.g. [Tho97]), thanks to the Büchi-Landweber Theorem [BL69], Cor. 7.3.2 implies the decidability of emptiness for non-deterministic automata as well as the *Rabin Basis Theorem* [Rab72], stating that if $\mathcal{L}(\mathcal{N}) \neq \emptyset$, then its contains a regular tree T and a finite state winning P-strategy on $\mathcal{N}(T)$ (both effectively definable from $\mathcal{N}$).

Corollary 7.3.3. *Given a non-deterministic automaton $\mathcal{N} : \Sigma$, one can decide whether $\mathcal{L}(\mathcal{N})$ is empty. Moreover, if $\mathcal{L}(\mathcal{N}) \neq \emptyset$ then one can effectively build from $\mathcal{N}$ a regular tree $T \in \mathcal{L}(\mathcal{N})$ together with a finite state winning* P*-strategy on $\mathbf{I} \multimap \mathcal{N}(T)$.*

More generally, strategies witnessing (non-)emptiness obtained via Cor. 7.3.2 can be lifted to winning strategies in games of the form $\mathcal{A} \multimap \mathcal{B}^{\perp}$, for $\mathcal{A}$ and $\mathcal{B}$ non-deterministic. In the setting of Ex. 7.2.15, if $\mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B}) = \emptyset$, then an O-strategy witnessing $\mathcal{L}(\mathcal{A} \otimes \mathcal{B}) = \emptyset$, which corresponds via Prop. 7.2.8[1] to a P-strategy witnessing $\mathbf{1} \in \mathcal{L}((\exists_{\Sigma}(\mathcal{A} \otimes \mathcal{B}))^{\perp})$, can be lifted to a winning P-strategy in $\mathcal{A} \multimap \mathcal{B}^{\perp}$. Indeed, from $\mathcal{L}((\exists_{\Sigma}(\mathcal{A} \otimes \mathcal{B}))^{\perp}) \neq \emptyset$, we can derive (again leaving implicit some structural and (Cut) rules):

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{L}\big(\exists_{\Sigma}(\mathcal{A} \otimes \mathcal{B}) \multimap \perp\big) \neq \emptyset}{\mathbf{1}\ ;\ \vdash \exists_{\Sigma}(\mathcal{A} \otimes \mathcal{B}) \multimap \perp}}{\mathbf{1}\ ;\ \exists_{\Sigma}(\mathcal{A} \otimes \mathcal{B}) \vdash \perp}}{\Sigma\ ;\ \mathcal{A} \otimes \mathcal{B} \vdash \perp}}{\Sigma\ ;\ \mathcal{A}, \mathcal{B} \vdash \perp}}{\Sigma\ ;\ \mathcal{A} \vdash \mathcal{B}^{\perp}}
$$

from which Adequacy (Prop. 7.2.1) gives a (finite-state) realizer of $\mathcal{A} \multimap \mathcal{B}^{\perp}$.

Proposition 7.3.4. *Given non-deterministic $\mathcal{A}, \mathcal{B} : \Sigma$, if $\mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B}) = \emptyset$, then there are winning* P*-strategies in $\mathcal{A} \otimes \mathcal{B} \multimap \perp$ and $\mathcal{A} \multimap \mathcal{B}^{\perp}$. Moreover, these* P*-strategies can be assumed to be finite state and can be effectively obtained from $\mathcal{A}$ and $\mathcal{B}$.*

Proposition 7.3.4, together with Ex. 7.2.5.(ii), implies the following extension of Ex. 7.2.5.(i).

Corollary 7.3.5. *If $\mathcal{A}, \mathcal{B} : \Sigma$ are non-deterministic and such that $\mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B}) = \emptyset$, then $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B} \multimap \mathcal{A}) \subseteq \mathcal{L}(\mathcal{B}^{\perp})$.*

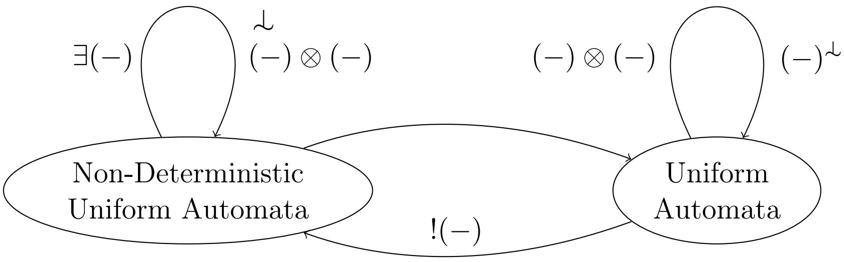
7.4. Simulation and the Exponential Modality of IMELL

Recall that similarly as in the usual setting, uniform automata have linear complements (§7.2.2), and that non-deterministic automata have correct existential quantifications (§7.3.2). On the other hand, we mentioned in §1.4 that in the usual setting, the *Simulation Theorem* [MS87, EJ91, MS95] (see also §2.2) says that each alternating automaton $\mathcal{A}$ can be simulated by a non-deterministic automaton $!\mathcal{A}$ (of exponential size) with $\mathcal{L}(!\mathcal{A}) = \mathcal{L}(\mathcal{A})$.

One can show that in our setting, an easy adaptation of the construction used in [Wal02] gives a similar simulation operation $!(-)$, taking a uniform automaton $\mathcal{A} : \Sigma$ to a non-deterministic automaton $!\mathcal{A} : \Sigma$ with $\mathcal{L}(!\mathcal{A}) = \mathcal{L}(\mathcal{A})$, thus completing the picture (1.5) of §1.5 for our notion

[1]More precisely, this is direction ($\Leftarrow$) in the proof of Prop. 7.2.8.

of uniform automata:



(7.10)

Moreover, we can show that the operation $!(-)$ satisfies the *deduction* rules of the exponential modality $!(-)$ of IMELL:

$$\frac{M\ ;\ !\overline{\mathcal{A}}\ \vdash\ \mathcal{A}}{M\ ;\ !\overline{\mathcal{A}}\ \vdash\ !\mathcal{A}} \qquad \frac{M\ ;\ \overline{\mathcal{A}},\mathcal{B}\ \vdash\ \mathcal{A}}{M\ ;\ \overline{\mathcal{A}},!\mathcal{B}\ \vdash\ \mathcal{A}} \qquad \frac{M\ ;\ \overline{\mathcal{A}},\ \vdash\ \mathcal{A}}{M\ ;\ \overline{\mathcal{A}},!\mathcal{B}\ \vdash\ \mathcal{A}} \qquad \frac{M\ ;\ \overline{\mathcal{A}},!\mathcal{B},!\mathcal{B}\ \vdash\ \mathcal{A}}{M\ ;\ \overline{\mathcal{A}},!\mathcal{B}\ \vdash\ \mathcal{A}} \tag{7.11}$$

It follows that the exponential ! makes it possible to define, using Girard's decomposition, an intuitionistic implication $(-) \to (-)$ as $\mathcal{A} \to \mathcal{B} := !\mathcal{A} \multimap \mathcal{B}$.

We shall not detail the construction of $!(-)$ nor its correctness proofs, for which we refer to [Rib18], and rather only make couple of remarks on these. First, for a uniform automaton

$$\mathcal{A} \quad = \quad (Q_{\mathcal{A}},\, q^{\imath}_{\mathcal{A}},\, U,\, X,\, \partial_{\mathcal{A}},\, \Omega_{\mathcal{A}})$$

the exponential automaton $!\mathcal{A}$ has shape

$$!\mathcal{A} \quad = \quad (Q_{!\mathcal{A}},\, q^{\imath}_{!\mathcal{A}},\, U^{Q_{\mathcal{A}}},\, \mathbf{1},\, \partial_{!\mathcal{A}},\, \Omega_{!\mathcal{A}})$$

A simple but important observation (to our knowledge due to [MS95] for usual alternating automata) is that $!\mathcal{A}$ is deterministic whenever $\mathcal{A}$ is a universal automaton (*i.e.* with $U \simeq \mathbf{1}$). We elaborate on this in the case of ω-words (§7.5).

The rules (7.11) are an obvious adaptation to our context of the rules displayed in (4.16) and (4.17) of §4.5. Since $!\mathcal{A}$ is non-deterministic, adequacy of the *weakening* and *contraction* rules

$$\frac{M\ ;\ \mathcal{A}_1,\ldots,\mathcal{A}_n\ \vdash\ \mathcal{A}}{M\ ;\ \mathcal{A}_1,\ldots,\mathcal{A}_n,!\mathcal{B}\ \vdash\ \mathcal{A}} \qquad\qquad \frac{M\ ;\ \mathcal{A}_1,\ldots,\mathcal{A}_n,!\mathcal{B},!\mathcal{B}\ \vdash\ \mathcal{A}}{M\ ;\ \mathcal{A}_1,\ldots,\mathcal{A}_n,!\mathcal{B}\ \vdash\ \mathcal{A}}$$

directly follow from the rules ($\textsc{Weak}_{\textsc{ND}}$) and ($\textsc{Contr}_{\textsc{ND}}$) displayed in (7.9). The rule

$$(\textsc{Dereliction})\ \frac{M\ ;\ \mathcal{A}_1,\ldots,\mathcal{A}_n,\mathcal{B}\ \vdash\ \mathcal{A}}{M\ ;\ \mathcal{A}_1,\ldots,\mathcal{A}_n,!\mathcal{B}\ \vdash\ \mathcal{A}}$$

follows from the existence of "*co-unit*" maps $!\mathcal{A} \multimap \mathcal{A}$, which, when $\mathcal{A}$ is in state $q_{\mathcal{A}}$, simply take an O move $f \in U^{Q_{\mathcal{A}}}$ to the P move $f(q_{\mathcal{A}}) \in U$. The difficult rule is

$$(\textsc{Promotion})\ \frac{M\ ;\ !\mathcal{A}_1,\ldots,!\mathcal{A}_n\ \vdash\ \mathcal{A}}{M\ ;\ !\mathcal{A}_1,\ldots,!\mathcal{A}_n\ \vdash\ !\mathcal{A}}$$

We have seen in §7.3 above that the symmetric monoidal structure of $\mathsf{DialAut}_\Sigma$ is Cartesian on non-deterministic automata, in other words that non-deterministic automata have a canonical comonoid structure (7.8). It follows that similarly as with usual IMELL-exponentials (see §4.5

but also [Mel09]), the simulation operation $!(-)$ adds to an arbitrary automaton $\mathcal{A}$ the structure allowing $!\mathcal{A}$ to be equipped with canonical maps:

$$!\mathcal{A} \quad \multimap \quad !\mathcal{A} \otimes !\mathcal{A} \qquad \text{and} \qquad !\mathcal{A} \quad \multimap \quad \mathbf{I}$$

On the other hand, recall from §7.2.1 that for a uniform automaton $\mathcal{A}$ with set of O-moves X, realizers of

$$\mathcal{A} \quad \multimap \quad \mathcal{A} \otimes \mathcal{A}$$

may not exist because O can play two different $(x, x') \in X \times X$ in the right component $\mathcal{A} \otimes \mathcal{A}$, that P may not be able to merge into a single $x'' \in X$ in the left component $\mathcal{A}$.

Usual solutions to this merging problem for IMELL-exponentials (see e.g. [Mel09, AC98, Mel04]) amount to equip objects of the form $!A$ with some duplication and memory abilities, essentially allowing $!A$ to run several copies of A. However (and this is via (1.4) §1.4, the crux of Rabin's Theorem [Rab69]), such recipes cannot (at least in an obvious way) be applied to automata on infinite trees, because $!\mathcal{A}$ must be a finite-state automaton, while plays in acceptance games (which are infinite) would require an infinite memory.

Phrased in modern terms, the solution is given by the existence of positional winning strategies in parity games (see §2.2). In our case, we resort on a notion of positionality for game graphs similar to those of [Rib15] (see also Rem. 2.2.5 and Chap. 5). Thanks to usual operations on automata, we turn $\mathcal{A}$ to an isomorphic *parity* automaton $\mathcal{A}^\dagger$ (with possibly more states than $\mathcal{A}$). As a result, all automata in the premise of the rule are equipped with parity conditions. Since parity conditions are closed under complement, it follows that the game

$$!\mathcal{A}_1 \otimes \cdots \otimes !\mathcal{A}_n \quad \multimap \quad \mathcal{A}^\dagger$$

is equipped with a disjunction of parity conditions, also known as a *Rabin condition* (see e.g. [Tho97]). We then rely on the known fact that in Rabin games, if P has a winning strategy, then P has *positional* winning strategy [Kla94, KK95, Jut97, Zie98] (this in general also hold for O only if the Rabin condition is itself equivalent to a parity condition).[2] Unfortunately, positionality is not preserved by composition, and the interpretation of the (Promotion) rule is not preserved by cut-elimination (in the sense of Rem. 7.2.3).

Remark 7.4.1 (On (Non) Standard Semantics (Continuing Rem. 7.2.14))**.** *In (7.10), we have only displayed existential quantifications for non-deterministic automata, because as with usual alternating automata, they are correct (in the sense of Cor. 7.3.2) only on non-deterministic automata. Similarly, we have not displayed universal quantifications because they are only complete on* universal *automata (see Def. 7.0.1). Moreover, we displayed linear negations but not general linear implications, since besides Ex. 7.2.5 and Cor. 7.3.5 we do not know much on their standard meaning.*

On the other hand, these connectives have the expected categorical properties and thus the deduction rules of Fig. 7.1 hold on general uniform automata. We refer to §8.2.5 for further comments in the case of ω-words.

We now gather consequences of the rules (7.11), thus extending §7.3.1-7.3.3. First, the rule (Dereliction) implies that $\mathcal{L}(!\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A})$, while the rule (Promotion) gives the converse inclusion $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(!\mathcal{A})$. We thus have, as expected:

Corollary 7.4.2. $\mathcal{L}(\mathcal{A}) = \mathcal{L}(!\mathcal{A})$.

[2] A property analyzed using the notion of *split trees* in [Zie98] (see also [PP04]). We leave for further work to give logical account of this notion.

Corollary 7.4.2 gives the extension of Cor. 7.3.3 to general uniform automata.

Corollary 7.4.3. *Given a uniform automaton $\mathcal{A}$, one can decide whether $\mathcal{L}(\mathcal{A})$ is empty. Moreover, if $\mathcal{L}(\mathcal{A}) \neq \emptyset$ then one can effectively build from $\mathcal{A}$ a regular tree $T \in \mathcal{L}(\mathcal{A})$ together with a finite state winning* P*-strategy on* $\mathbf{1} \vdash \mathbf{I} \multimap \mathcal{A}(\dot{T})$.

We also obtain the following lifting property, extending Prop. 7.3.4. Let $?\mathcal{A} := (!\mathcal{A}^{\perp})^{\perp}$.

Proposition 7.4.4 (Weak Completeness)**.** *Given automata $\mathcal{A}, \mathcal{B} : \Sigma$, if $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$ then there is an effective winning* P*-strategy in* $\Sigma \vdash !\mathcal{A} \multimap ?\mathcal{B}$.

Proposition 7.4.4 is a completeness result on realizability w.r.t. language inclusion. It is only a weak converse to the soundness of realizability w.r.t. language inclusion (Prop. 7.1.4, §7.1.3), because it imposes constraints on the *shape* of automata for the implication to be realizable (while it imposes no constraint on the *languages* involved as $\mathcal{L}(\mathcal{A}) = \mathcal{L}(!\mathcal{A})$ and $\mathcal{L}(\mathcal{B}) = \mathcal{L}(?\mathcal{B})$). Note that since non-determinism is preserved by $\otimes$, Prop. 7.4.4 can be extended to:

$$\mathcal{L}(\mathcal{A}_1) \cap \dots \cap \mathcal{L}(\mathcal{A}_n) \quad \subseteq \quad \mathcal{L}(\mathcal{B}) \qquad \Longrightarrow \qquad !\mathcal{A}_1 \otimes \ldots \otimes !\mathcal{A}_n \multimap ?\mathcal{B} \text{ is realized.}$$

As an example of use of the exponential rules, we mention a negative translation of the law of Peirce $((A \to B) \to A) \to A$. The law of Peirce gives full classical logic when added to intuitionistic logic. Recall that $\mathcal{A} \to \mathcal{B} := !\mathcal{A} \multimap \mathcal{B}$.

Example 7.4.5. *One can derive $((?\mathcal{A} \to ?\mathcal{B}) \to ?\mathcal{A}) \to ?\mathcal{A}$ thanks to the exponential rules (see also Ex. 8.1.3).*

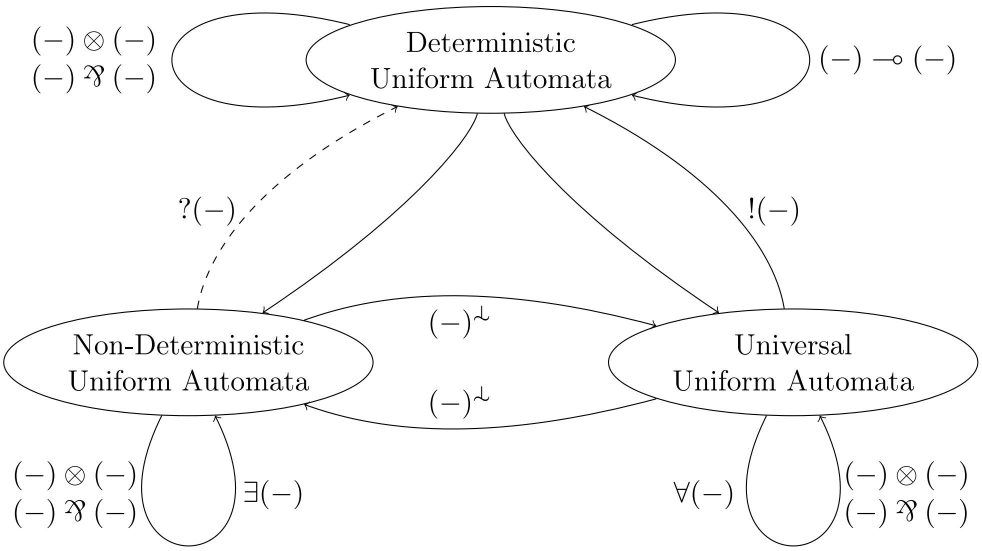
7.4.1. Further Examples. Weak Completeness (Prop. 7.4.4), namely that $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$ implies the existence of (finite-state) realizers of $!\mathcal{A} \multimap ?\mathcal{B}$, seems related to a similar universal property of the *guidable automata* of [CL08]. Besides, we show in [Rib18, App. C] that the construction of [CL08] (which is based on the complementation construction of [Tho97, Proof of Thm. 6.9] rather than on the Simulation Theorem) can be reproduced in our setting, with the same universal property. Furthermore, we also show in [Rib18, App. C] how this universal property combined with our linear arrow makes it possible to reproduce the construction used for the separation property of [SA05, Thm. 2.7].

7.5. Uniform Automata on Infinite Words

We now briefly discuss the (much simpler) case of uniform automata on ω-words, *i.e.* the case of $\mathfrak{D} \simeq \mathbf{1}$. We refer to [PR18b, Pra19] for details.

The simple but fundamental phenomenon on ω-words is that when $\mathfrak{D} \simeq \mathbf{1}$, the falsity automaton $\perp$ of §7.2.2 becomes *deterministic*, so that linear negation $(-) \multimap \perp$ turns non-deterministic automata to universal ones and *vice-versa*. Further, since $!\mathcal{A}$ is deterministic whenever $\mathcal{A}$ is universal, it follows that taking $?\mathcal{A} := !(\mathcal{A} \multimap \perp) \multimap \perp$, the automaton $?\mathcal{A}$ is deterministic whenever $\mathcal{A}$ is non-deterministic (a basic observation, to our knowledge due to [MS95]). In particular, the *Simulation Theorem* (together with linear complementation) implies McNaughton's Determinization Theorem [McN66], and the classes of general automata, non-deterministic, uniform and deterministic automata all have the same expressive power.

Moreover, uniform automata on ω-words can be equipped with a multiplicative disjunction $\parr$, defined exactly as $\otimes$, but with acceptance condition given by a disjunction rather than a conjunction as in (7.3). In addition, since the class of universal automata is as expressive as the other classes of automata on ω-words, the universal quantifier on automata may now be thought about as providing a real primitive universal quantification at the level of automata. The situation can be summarized with the following refinement of (7.10) (§7.4, see also (1.5), §1.5), in which for simplicity we do not display general uniform automata, nor the contravariant action of $\multimap$ on the polarity of its first argument:



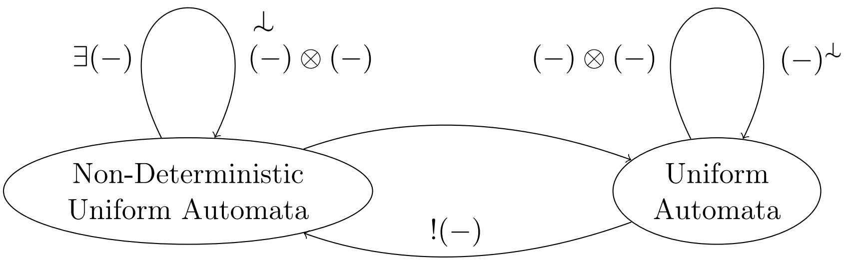
(7.12)

8. A Curry-Howard Interpretation of a Linear Variant of MSO

This Chapter presents the logical counterpart of the realizability model of Chap. 7, namely the logics LMSO and their relations to MSO. In consists of three Sections. We begin in §8.1 with the case of infinite trees. The focus is on the general setting and on general soundness properties, of the realizability model w.r.t. LMSO, but also of syntactic translations of MSO to LMSO. Then §8.2 discusses the case of ω-words, which was explored in collaboration with Pierre Pradic. It turns out that the corresponding linear system has good properties for *Church's Synthesis* (see §8.2.2–8.2.4 for a presentation), as well as good completeness properties (in relation to §3.1.4). We finally come back to infinite trees in §8.3, with a brief discussion of the axiomatic aspects of MSO($\mathfrak{D}$) (Chap. 3) w.r.t. the realizability model.

8.1. A Linear Monadic Second-Order Logic over Infinite Trees

This Section presents a version of LMSO over infinite trees, which essentially results from the picture (1.5) (§1.5) of the realizability model:



(8.1)

As we have seen in Chap. 3 that giving a complete axiomatization to MSO over infinite trees is not yet done in a completely satisfactory way, we shall content ourselves here with the structural aspects of factorizing the translations of MSO to automata through LMSO. In particular, we shall depart from the setting of MSO($\mathfrak{D}$) (§3.1) and only consider many-sorted logics with equality, respectively MSO($\mathbf{T}$) and LMSO($\mathbf{T}$). These logics are based on a many-sorted term language for finite-state $\mathbf{T}$ morphisms (§6.5.2) that we present in §8.1.1. Then §8.1.2 presents the logic MSO($\mathbf{T}$) itself. Its language is based on the logical connectives of MSO given in §1.2 (which themselves followed a structure suggested by (8.1)). We also give a deduction system for MSO($\mathbf{T}$), which is merely many-sorted classical first-order logic with equality, with the proviso that we adopt sequent with exactly one formula on the right of the $\vdash$, ultimately because (as noted in §4.3) there is no multiplicative disjunction ($\parr$) on tree automata. LMSO($\mathbf{T}$) itself is discussed in §8.1.3–8.1.7. The language of LMSO($\mathbf{T}$) (§8.1.3) follows the connectives on uniform automata devised in Chap. 7, and is thus slightly richer than (8.1). Deduction for LMSO($\mathbf{T}$) (§8.1.5) is essentially given by the rules for LMSO presented in Fig. 1.1. Perhaps the most important part of this Section is §8.1.7, which discusses syntactic translations from

$\mathsf{MSO}(\mathbf{T})$ to $\mathsf{LMSO}(\mathbf{T})$ based on the usual T and Q translations of classical logic to classical linear logic [DJS97]. On the other hand, §8.1.4 and §8.1.6 contain mainly bureaucratic material concerning respectively the interpretation of LMSO formulas and proofs as automata and finite-state realizers.

8.1.1. A Term Language. Following the approach of [PR18b, PR19] (see also [Pra19]), when devising proof-relevant translations of MSO to the automata-based realizability model of Chap. 7, it is convenient to start from a version of MSO with a term language for finite-state $\mathbf{T}$-morphisms.

For inessential technical reasons, we shall restrict to *binary alphabets*, which are finite non-empty sets of the form $\mathbf{2}^p$ for some $p \in \mathbb{N}$, with $\mathbf{1} = \mathbf{2}^0$. Note that binary alphabets are closed under Cartesian products and set-theoretic function spaces. It follows that taking $[\![o]\!] := \mathbf{2}$, we have a binary alphabet $[\![\tau]\!]$ for each simple type $\tau \in \mathrm{ST}$, where

$$\sigma, \tau \in \mathrm{ST} \quad ::= \quad \mathbf{1} \quad | \quad o \quad | \quad \sigma \times \tau \quad | \quad \sigma \to \tau$$

We often write $(\tau)\sigma$ for the type $\sigma \to \tau$.

The idea is simply to have one function symbol for each f.s. $\mathbf{T}$ morphism. More precisely, we have a many-sorted signature, with one sort for each simple type $\tau \in \mathrm{ST}$, and with one function symbol of arity $(\sigma_1, \dots, \sigma_n; \tau)$ for each finite-state morphism $[\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!] \to_{\mathbf{T}} [\![\tau]\!]$. A term $\mathtt{t}$ of sort τ (notation $\mathtt{t}^\tau$) with free variables among $x_1^{\sigma_1}, \dots, x_n^{\sigma_n}$ (we say that $\mathtt{t}$ is of arity $(\sigma_1, \dots, \sigma_n; \tau)$) thus induces a finite-state $[\![\mathtt{t}]\!] : [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!] \to_{\mathbf{T}} [\![\tau]\!]$. Given a valuation $x_i \mapsto M_i \in [\![\sigma_i]\!]^{\mathfrak{D}^*} \simeq \mathbf{T}[\mathbf{1}, [\![\sigma_i]\!]]$ for $i \in \{1, \dots, n\}$, we then obtain a labeled $\mathfrak{D}$-ary tree

$$[\![\mathtt{t}]\!] \circ \langle M_1, \dots, M_n \rangle \quad \in \quad \mathbf{T}[\mathbf{1}, [\![\tau]\!]] \quad \simeq \quad [\![\tau]\!]^{\mathfrak{D}^*}$$

8.1.2. The Logic $\mathsf{MSO}(\mathbf{T})$. The logic $\mathsf{MSO}(\mathbf{T})$ is “simply” classical first-order logic on top of the term language of §8.1.1. While the choice of connectives may seem irrelevant at first sight, it is actually quite crucial to respect picture (8.1) together with the two interpretations of negation discussed in §1.5:

$$(\neg\varphi)^{\mathrm{nd}} \quad := \quad !(\varphi^{\mathrm{nd}} \multimap \bot) \qquad \text{and} \qquad (\neg\varphi)^{\mathrm{alt}} \quad := \quad \varphi^{\mathrm{alt}} \multimap \bot \tag{8.2}$$

the former living in the world of non-deterministic automata, the latter in that of (alternating) uniform automata. The moral of (8.1) and (8.2) is that it may be wise to think that we are living not so far from (first-order) *tensorial logic* (see e.g. [Mel13] and [Mel17b, §5.6, p. 137]), and to take the basic connectives proposed in §1.2, namely

$$\varphi, \psi \quad ::= \quad \mathtt{t}^\tau \doteq \mathtt{u}^\tau \quad | \quad \top \quad | \quad \bot \quad | \quad \neg\varphi \quad | \quad \varphi \wedge \psi \quad | \quad (\exists x^\tau)\varphi$$

$\mathsf{MSO}(\mathbf{T})$ has an obvious *standard* interpretation, obtained by letting $\exists x^\tau$ range over $\mathbf{T}[\mathbf{1}, [\![\tau]\!]] \simeq [\![\tau]\!]^{\mathfrak{D}^*}$, and by interpreting sorted equalities $\mathtt{t}^\tau \doteq \mathtt{u}^\tau$ by equality over $\mathbf{T}[\mathbf{1}, [\![\tau]\!]] \simeq [\![\tau]\!]^{\mathfrak{D}^*}$. Given a formula $\varphi(x_1^{\sigma_1}, \dots, x_n^{\sigma_n})$ with free variables a displayed and a valuation $x_i \mapsto M_i \in [\![\sigma_i]\!]^{\mathfrak{D}^*} \simeq \mathbf{T}[\mathbf{1}, [\![\sigma_i]\!]]$ for $i \in \{1, \dots, n\}$, we write $\models \varphi(M_1, \dots, M_n)$ when φ holds under the standard interpretation, with x_i interpreted by M_i. We let $\mathcal{L}(\varphi)$ be the set of all $\langle M_1, \dots, M_n \rangle \in \mathbf{T}[\mathbf{1}, [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!]]$ such that $\models \varphi(M_1, \dots, M_n)$. In the following, we often confuse $\mathbf{T}[\mathbf{1}, [\![\sigma]\!]]$ with $[\![\sigma]\!]^{\mathfrak{D}^*}$.

The logic $\mathsf{MSO}(\mathbf{T})$ (w.r.t. this standard interpretation) is of course definable in $\mathsf{MSO}(\mathfrak{D})$ w.r.t. its standard model (see §3.1.3).

In order for our Curry-Howard approach to make sense, also need a deduction system for $\mathsf{MSO}(\mathbf{T})$. As we have seen in Chap. 3 that giving a complete axiomatization to MSO over

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

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

$$\frac{\overline{\varphi}, \varphi \vdash \psi}{\overline{\varphi}, (\exists z^\tau)\varphi \vdash \psi} \qquad \frac{\overline{\varphi} \vdash \varphi[\mathtt{t}^\tau / x^\tau]}{\overline{\varphi} \vdash (\exists x^\tau)\varphi}$$

Figure 8.1.: Deduction Rules for $\mathsf{MSO}(\mathbf{T})$ (where z^τ is fresh).

infinite trees is not yet done in a completely satisfactory way, we shall content ourselves from the moment with classical deduction for first-order many-sorted logic, augmented with all true equations between finite-state $\mathbf{T}$-morphisms. We defer to §8.3 a discussion of the basic axioms of FSO (§3.1.2). The much simpler case of infinite words is addressed in §8.2.

Definition 8.1.1 (Deduction for $\mathsf{MSO}(\mathbf{T})$)**.** *Deduction for* $\mathsf{MSO}(\mathbf{T})$ *is given by the rules of Fig. 8.1 augmented with the following axiom schemes.*

- Equality Axioms. *Given terms* $\mathtt{t}^\tau$ *and* $\mathtt{u}^\tau$ *of the same arity* $(\sigma_1, \dots, \sigma_n; \tau)$*:*

$$\frac{}{\vdash\ \mathtt{t}^\tau \doteq \mathtt{t}^\tau} \qquad \frac{}{\mathtt{t}^\tau \doteq \mathtt{u}^\tau,\ \varphi[\mathtt{t}^\tau/x^\tau]\ \vdash\ \varphi[\mathtt{u}^\tau/x^\tau]} \qquad \frac{[\![\mathtt{t}^\tau]\!] = [\![\mathtt{u}^\tau]\!]}{\vdash\ \mathtt{t}^\tau \doteq \mathtt{u}^\tau}$$

- Classical Logic. *For each formula* φ*:*

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

Note that the rules of Fig. 8.1 involve (intuitionistic) sequents with exactly one formula on the right of the $\vdash$. This is because we do not have a multiplicative disjunction on tree automata (see §4.3-4.4). As a consequence, classical logic is obtained via an additional axiom, which is actually an instance of the usual *Peirce's Law*:

$$\big((\varphi \to \psi) \to \varphi\big) \to \varphi$$

Elimination of double negation is of course derivable in our system (we leave *exchange* rules implicit):

$$\cfrac{\cfrac{\cfrac{}{\neg\varphi\ \vdash\ \neg\varphi}}{\neg\neg\varphi,\ \neg\varphi\ \vdash\ \varphi}}{\neg\neg\varphi\ \vdash\ \varphi}$$

Note also that $\mathsf{MSO}(\mathbf{T})$ proves the following equations (thanks to finite products w.r.t. finite-state $\mathbf{T}$-morphisms):

$$\pi_i(\langle \mathtt{t}_1, \dots, \mathtt{t}_n \rangle) \ \doteq_{\sigma_i}\ \mathtt{t}_i \qquad \text{and} \qquad \mathtt{t}\ \doteq_{\sigma_1 \times \cdots \times \sigma_n}\ \langle \pi_1(\mathtt{t}), \dots, \pi_n(\mathtt{t}) \rangle \tag{8.3}$$

Hence each formula $\varphi(a_1^{\sigma_1}, \dots, a_n^{\sigma_n})$ can be seen as a formula $\varphi(a^{\sigma_1 \times \cdots \times \sigma_n})$.

8.1.3. The Language of $\mathsf{LMSO}(\mathbf{T})$. For the linear variant $\mathsf{LMSO}(\mathbf{T})$ of $\mathsf{MSO}(\mathbf{T})$, we consider connectives exactly mirroring those of Chap. 7 on uniform automata. The formulae of $\mathsf{LMSO}(\mathbf{T})$ are given by the grammar:

$$\varphi, \psi \quad ::= \quad \mathtt{t}^\tau \doteq \mathtt{u}^\tau \quad | \quad \mathbf{I} \quad | \quad \bot \quad | \quad \varphi \otimes \psi \quad | \quad \psi \multimap \varphi \quad | \quad !\varphi \quad | \quad (\exists x^\tau)\varphi \quad | \quad (\forall x^\tau)\varphi$$

Note that we do not impose polarity constraints on existential quantifiers and that we allowed universal quantifiers as well as general linear implications, so that $\mathsf{LMSO}(\mathbf{T})$ should be thought about has having a non-standard semantics (see Rem. 7.2.14 and Rem. 7.4.1). We elaborate on this in the case of ω-words in §8.2.5.

8.1.4. Interpretation of $\mathsf{LMSO}(\mathbf{T})$ Formulae as Uniform Automata. The connectives of $\mathsf{LMSO}(\mathbf{T})$ exactly correspond to the connectives on uniform tree automata devised in Chap. 7. In order to interpret formulae as automata, it thus only remains to properly handle free variables.

Consider an $\mathsf{LMSO}(\mathbf{T})$ formula $\varphi(x_1^{\sigma_1}, \dots, x_n^{\sigma_n})$ with free variables as shown. This formula is interpreted as a uniform automaton $\mathcal{A}(\varphi)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n})$ over the alphabet $[\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!]$. But note that a formula φ with free variables among $x_1^{\sigma_1}, \dots, x_n^{\sigma_n}$ as above is also a formula with free variables among, say, $x_1^{\sigma_1}, \dots, x_n^{\sigma_n}, y_1^{\tau_1}, \dots, y_k^{\tau_k}$. Hence, the interpretation of φ is formally parametrized with a list of free variables containing all the free variables of φ, and with this convention, the automata $\mathcal{A}(\varphi)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n})$ and $\mathcal{A}(\varphi)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}, y_1^{\tau_1}, \dots, y_k^{\tau_k})$ are different (provided $n, k > 0$).

The interpretation is by induction on formulae. Consider first the case of the atomic formulae $\mathbf{I}$ and $\bot$, with the list of free variables $x_1^{\sigma_1}, \dots, x_n^{\sigma_n}$. We let, as expected

$$\begin{array}{rcl} \mathcal{A}(\mathbf{I})(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}) & := & (\mathbf{I} : [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!]) \\ \mathcal{A}(\bot)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}) & := & (\bot : [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!]) \end{array}$$

where $\mathbf{I}$ and $\bot$ are the automata of resp. of Ex. 7.0.2.(i) and §7.2.2, taken over the alphabet $[\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!]$.

Consider now the case of an equality $\mathtt{t}^\tau \doteq \mathtt{u}^\tau$ and a list of variables $x_1^{\sigma_1}, \dots, x_n^{\sigma_n}$ containing all the variables of $\mathtt{t}$ and $\mathtt{u}$. Then both $\mathtt{t}$ and $\mathtt{u}$ can be seen as terms of arity $(\sigma_1, \dots, \sigma_n; \tau)$. We let $\mathcal{A}(\mathtt{t} \doteq \mathtt{u})(x_1^{\sigma_1}, \dots, x_n^{\sigma_n})$ by the (deterministic) automaton over $\Sigma := [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!]$ which stays in an accepting state as long as $[\![\mathtt{t}]\!](\overline{\mathtt{a}}, p)$ and $[\![\mathtt{u}]\!](\overline{\mathtt{a}}, p)$ agree on inputs $\overline{\mathtt{a}} \in \Sigma^{k+1}$ and $p \in \mathfrak{D}^k$, and which goes to a rejecting sink state as soon as they differ.

The multiplicative connectives $\otimes$ and $\multimap$ are interpreted using the corresponding constructions on uniform automata of §7.2.1, while the exponential $!(-)$ is interpreted with the exponential construction mentioned in §7.4.

As for quantifications $(\exists x^\tau)\varphi$ and $(\forall x^\tau)\varphi$, given a list of variables $x_1^{\sigma_1}, \dots, x_n^{\sigma_n}$ (in which x^τ is assumed not to occur), we let

$$\begin{array}{rcl} \mathcal{A}\big((\exists x^\tau)\varphi\big)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}) & := & \exists_{[\![\tau]\!]}\big(\mathcal{A}(\varphi)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}, x^\tau)\big) \\ \mathcal{A}\big((\forall x^\tau)\varphi\big)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}) & := & \forall_{[\![\tau]\!]}\big(\mathcal{A}(\varphi)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}, x^\tau)\big) \end{array}$$

where $\exists_{[\![\tau]\!]}(-)$ and $\forall_{[\![\tau]\!]}(-)$ are the operations devised in §7.2.3.

The image in (8.1) of the interpretation of $\mathsf{LMSO}(\mathbf{T})$ formulae as automata gives rise to a polarized fragment of $\mathsf{LMSO}(\mathbf{T})$. The *deterministic* (notation $\varphi^\pm, \psi^\pm$) and the *weakly positive* (notation φ^+, ψ^+) formulae of $\mathsf{LMSO}(\mathbf{T})$ are defined as

$$\begin{array}{rcl} \varphi^\pm, \psi^\pm & ::= & \mathbf{I} \quad | \quad \mathtt{t}^\tau \doteq \mathtt{u}^\tau \\ \varphi^+, \psi^+ & ::= & \varphi^\pm \quad | \quad \bot \quad | \quad \psi^\pm \multimap \varphi^+ \quad | \quad \varphi^+ \otimes \psi^+ \quad | \quad (\exists x^\tau)\varphi^+ \quad | \quad !\varphi \end{array}$$

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

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

$$
\frac{\overline{\varphi}, \varphi \vdash \psi}{\overline{\varphi} \vdash \varphi \multimap \psi}
\qquad
\frac{\overline{\varphi} \vdash \varphi \qquad \overline{\psi}, \psi \vdash \psi'}{\overline{\varphi}, \overline{\psi}, \varphi \multimap \psi \vdash \psi'}
\qquad
\frac{\overline{\varphi}, \varphi[\mathtt{t}^\tau / x^\tau] \vdash \psi}{\overline{\varphi}, (\forall x^\tau)\varphi \vdash \psi}
\qquad
\frac{\overline{\varphi} \vdash \varphi}{\overline{\varphi} \vdash (\forall z^\tau)\varphi}
$$

$$
\frac{}{\overline{\varphi} \vdash \mathbf{I}}
\qquad
\frac{\overline{\varphi}, \varphi, \psi \vdash \varphi'}{\overline{\varphi}, \varphi \otimes \psi \vdash \varphi'}
\qquad
\frac{\overline{\varphi} \vdash \varphi \qquad \overline{\psi} \vdash \psi}{\overline{\varphi}, \overline{\psi} \vdash \varphi \otimes \psi}
\qquad
\frac{\overline{\varphi}, \varphi \vdash \psi}{\overline{\varphi}, (\exists z^\tau)\varphi \vdash \psi}
\qquad
\frac{\overline{\varphi} \vdash \varphi[\mathtt{t}^\tau / x^\tau]}{\overline{\varphi} \vdash (\exists x^\tau)\varphi}
$$

Figure 8.2.: Deduction Rules for $\mathsf{LMSO}(\mathbf{T})$ (where z^τ is fresh).

Hence positive formulae are interpreted as non-deterministic automata while deterministic formulae are interpreted as deterministic automata. Note that $\perp$ is positive since ⅄ is non-deterministic. We call these polarities *weak* because in the positive $!\varphi$ we do not ask φ to be positive. This is due to the fact that ⅄ is not deterministic, which is unavoidable since deterministic tree automata are less expressive than non-deterministic ones (see also §7.5).

8.1.5. The Theory $\mathsf{LMSO}(\mathbf{T})$. Deduction for $\mathsf{LMSO}(\mathbf{T})$ is defined by analogy with deduction for $\mathsf{MSO}(\mathbf{T})$, taking the usual rules for the linear connectives.

Definition 8.1.2 (Deduction for $\mathsf{LMSO}(\mathbf{T})$)**.** *Deduction for* $\mathsf{MSO}(\mathbf{T})$ *is given by the rules of Fig. 8.2 augmented with the following axiom scheme.*

- Equality Axioms. *Given terms* $\mathtt{t}^\tau$ *and* $\mathtt{u}^\tau$ *of the same arity* $(\sigma_1, \dots, \sigma_n; \tau)$*:*

$$
\frac{}{\vdash \ \mathtt{t}^\tau \doteq \mathtt{t}^\tau}
\qquad\qquad
\frac{}{\mathtt{t}^\tau \doteq \mathtt{u}^\tau,\ \varphi[\mathtt{t}^\tau/x^\tau] \ \vdash \ \varphi[\mathtt{u}^\tau/x^\tau]}
\qquad\qquad
\frac{[\![\mathtt{t}^\tau]\!] = [\![\mathtt{u}^\tau]\!]}{\vdash \ \mathtt{t}^\tau \doteq \mathtt{u}^\tau}
$$

Note that there is no specific rule for $\perp$ (and in particular no Ex Falso rule), so that $\perp$ is not really considered as logical connective. This reflects the specific nature of the automaton ⅄, which is not dual to $\mathbf{I}$.

Similarly as with automata in §7.4, we shall heavily rely on the $?(-)$ exponential modality of Linear Logic. In classical settings, $?(-)$ is a (linear) De Morgan dual of $!(-)$, so that both are interdefinable. In our intuitionistic setting, while $?(-)$ can be thought about (and will be defined) as a primitive connective for automata on ω-words, it is a derived connective in the case of infinite trees. We let

$$
?\varphi \quad := \quad !(\varphi \multimap \perp) \multimap \perp
$$

We insist that the above definition of $?(-)$ is a mere macro, not intended to reflect the usual classical duality with $!(-)$. As shown in Fig. 8.3, we can nevertheless derive the usual introduction rules of $?(-)$:

$$
\frac{\overline{\varphi} \ \vdash \ \varphi}{\overline{\varphi} \ \vdash \ ?\varphi}
\qquad\qquad
\frac{!\overline{\varphi},\ \varphi \ \vdash \ ?\psi}{!\overline{\varphi},\ ?\varphi \ \vdash \ ?\psi}
\tag{8.4}
$$

as well as the usual multiplicative unit law for falsity:

$$
\perp \ \vdash \ ?\varphi \tag{8.5}
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\overline{\varphi} \vdash \varphi} \qquad \dfrac{}{\bot \vdash \bot}}{\overline{\varphi},\ \varphi \multimap \bot \vdash \bot}}{\overline{\varphi},\ !(\varphi \multimap \bot) \vdash \bot}}{\overline{\varphi} \vdash ?\varphi}
\qquad\qquad
\dfrac{\dfrac{\dfrac{}{\bot \vdash \bot}}{\bot,\ !(\varphi \multimap \bot) \vdash \bot}}{\bot \vdash ?\varphi}
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{!\overline{\varphi},\ \varphi \vdash ?\psi} \qquad \dfrac{}{!(\psi \multimap \bot) \vdash !(\psi \multimap \bot)}}{!\overline{\varphi},\ !(\psi \multimap \bot),\ \varphi \vdash \bot}}{!\overline{\varphi},\ !(\psi \multimap \bot) \vdash \varphi \multimap \bot}}{!\overline{\varphi},\ !(\psi \multimap \bot) \vdash !(\varphi \multimap \bot)} \qquad \dfrac{}{\bot \vdash \bot}}{!\overline{\varphi},\ !(\varphi \multimap \bot) \multimap \bot,\ !(\psi \multimap \bot) \vdash \bot}}{!\overline{\varphi},\ ?\varphi \vdash ?\psi}
$$

Figure 8.3.: The Derived Rules (8.4) and (8.5) of the Defined Modality $?(-)$.

In $\mathsf{LMSO}(\mathbf{T})$ we write $\psi \to \varphi$ for $!\psi \multimap \varphi$.

Example 8.1.3 (Peirce's Law). *The negative translation of Peirce's law* $((?\varphi \to ?\psi) \to ?\varphi) \to ?\varphi$ *(mentioned in Ex. 7.4.5 in the setting of automata) can be derived as in Fig. 8.4.*

8.1.6. Interpretation of $\mathsf{LMSO}(\mathbf{T})$ Proofs as Strategies. Thanks to the deduction system on automata of §7.2, we then get an interpretation of $\mathsf{LMSO}(\mathbf{T})$ proofs as strategies, thereby extending Prop. 7.2.1 to $\mathsf{LMSO}(\mathbf{T})$. Again, we must take care of free variables.

Proposition 8.1.4 (Adequacy)**.** *Consider an* $\mathsf{LMSO}(\mathbf{T})$ *sequent*

$$
\varphi_1, \dots, \varphi_k \ \vdash \ \varphi \tag{8.6}
$$

and a list of variables $x_1^{\sigma_1}, \dots, x_n^{\sigma_n}$ *containing all the free variables of* $\varphi_1, \dots, \varphi_k$ *and* φ*. Each proof of (8.6) induces a finite-state strategy in*

$$
\mathcal{A}(\varphi_1)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}) \otimes \dots \otimes \mathcal{A}(\varphi_k)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}) \quad \multimap_{\mathsf{DialAut}([\![\sigma_1]\!] \times \dots \times [\![\sigma_n]\!])} \quad \mathcal{A}(\varphi)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n})
$$

8.1.7. Translations of $\mathsf{MSO}(\mathbf{T})$ to $\mathsf{LMSO}(\mathbf{T})$. There are different possible approaches when devising translations of $\mathsf{MSO}(\mathbf{T})$ formulae to $\mathsf{LMSO}(\mathbf{T})$ formulae. The first (naive) possibility is to factorize translations of $\mathsf{MSO}(\mathbf{T})$ to automata via $\mathsf{LMSO}(\mathbf{T})$. This leads in particular to the translations $(-)^{\mathrm{nd}}$ and $(-)^{\mathrm{alt}}$ considered in §1.5

$$
\begin{array}{lcl@{\qquad\qquad}lcl}
\top^{\mathrm{nd}} & := & \mathbf{I} & \top^{\mathrm{alt}} & := & \mathbf{I} \\
\bot^{\mathrm{nd}} & := & \bot & \bot^{\mathrm{alt}} & := & \bot \\
(\mathtt{t}^{\tau} \doteq \mathtt{u}^{\tau})^{\mathrm{nd}} & := & \mathtt{t}^{\tau} \doteq \mathtt{u}^{\tau} & (\mathtt{t}^{\tau} \doteq \mathtt{u}^{\tau})^{\mathrm{alt}} & := & \mathtt{t}^{\tau} \doteq \mathtt{u}^{\tau} \\
(\neg\varphi)^{\mathrm{nd}} & := & !(\varphi^{\mathrm{nd}} \multimap \bot) & (\neg\varphi)^{\mathrm{alt}} & := & \varphi^{\mathrm{alt}} \multimap \bot \\
(\varphi \wedge \psi)^{\mathrm{nd}} & := & \varphi^{\mathrm{nd}} \otimes \psi^{\mathrm{nd}} & (\varphi \wedge \psi)^{\mathrm{alt}} & := & \varphi^{\mathrm{alt}} \otimes \psi^{\mathrm{alt}} \\
((\exists x^{\tau})\varphi)^{\mathrm{nd}} & := & (\exists x^{\tau})\varphi^{\mathrm{nd}} & ((\exists x^{\tau})\varphi)^{\mathrm{alt}} & := & (\exists x^{\tau})!\varphi^{\mathrm{alt}}
\end{array}
$$

We now have all the material we need to properly state Prop. 1.5.1, the correctness of $(-)^{\mathrm{nd}}$ and $(-)^{\mathrm{alt}}$. We state the result in the setting of $\mathsf{MSO}(\mathbf{T})$ and $\mathsf{LMSO}(\mathbf{T})$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{!(?\varphi \to ?\psi) \multimap ?\varphi \vdash !(?\varphi \to ?\psi) \multimap ?\varphi}}{!((?\varphi \to ?\psi) \to ?\varphi) \vdash !(?\varphi \to ?\psi) \multimap ?\varphi} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{!(\varphi \multimap \bot), ?\varphi \vdash \bot}}{!(\varphi \multimap \bot), !?\varphi \vdash \bot}}{!(\varphi \multimap \bot), !?\varphi, !(\psi \multimap \bot) \vdash \bot}}{!(\varphi \multimap \bot), !?\varphi \vdash ?\psi}}{!(\varphi \multimap \bot) \vdash ?\varphi \to ?\psi}}{!(\varphi \multimap \bot) \vdash !(?\varphi \to ?\psi)}}{!((?\varphi \to ?\psi) \to ?\varphi), !(\varphi \multimap \bot) \vdash ?\varphi} \quad \dfrac{}{!(\varphi \multimap \bot), ?\varphi \vdash \bot}}{!((?\varphi \to ?\psi) \to ?\varphi), !(\varphi \multimap \bot), !(\varphi \multimap \bot) \vdash \bot}}{\dfrac{!((?\varphi \to ?\psi) \to ?\varphi), !(\varphi \multimap \bot) \vdash \bot}{\dfrac{!((?\varphi \to ?\psi) \to ?\varphi) \vdash ?\varphi}{\vdash ((?\varphi \to ?\psi) \to ?\varphi) \to ?\varphi}}}
$$

Figure 8.4.: A Proof of a Negative Translation of Peirce's Law.

Proposition 8.1.5 (Prop. 1.5.1)**.** *Consider a closed* $\mathsf{MSO}(\mathbf{T})$ *formulae* φ*, and let* $(-)^{\text{trans}}$ *be either* $(-)^{\text{nd}}$ *or* $(-)^{\text{alt}}$*. Then* φ *is true (in the standard model) if and only if* $\mathcal{A}(\varphi^{\text{trans}})$ *accepts the unique* **1***-labeled tree.*

We now briefly discuss $(-)^{\text{nd}}$ and $(-)^{\text{alt}}$. Let us begin with $(-)^{\text{nd}}$. First, looking in details at possible implementations of $!(-)$ (e.g. following [Wal02, Rib18]), for $\mathcal{A}$ non-deterministic, $!\mathcal{A}$ is equivalent to $\mathcal{A}$, in the sense that the (Dereliction) counit map $!\mathcal{A} \multimap \mathcal{A}$ has a right inverse (so that $\mathcal{A}$ is a retract of $!\mathcal{A}$ in the sense of [Bor94, Def. 1.7.3]). As a consequence, in terms of realizability one might instead of $(-)^{\text{nd}}$ consider the following translation:

$$
\begin{array}{lcl}
\top^{\text{v}} & := & !\mathbf{I} \\
\bot^{\text{v}} & := & !\bot \\
(\mathtt{t}^\tau \doteq \mathtt{u}^\tau)^{\text{v}} & := & !(\mathtt{t}^\tau \doteq \mathtt{u}^\tau) \\
(\neg\varphi)^{\text{v}} & := & !(\varphi^{\text{v}} \multimap \bot) \\
(\varphi \wedge \psi)^{\text{v}} & := & !(\varphi^{\text{v}} \otimes \psi^{\text{v}}) \\
((\exists x^\tau)\varphi)^{\text{v}} & := & !(\exists x^\tau)\varphi^{\text{v}}
\end{array}
$$

Hence the non-deterministic translation $(-)^{\text{nd}}$ can be seen as a variant of the usual *call-by-value* translation of intuitionistic logic to intuitionistic linear logic (see e.g. [LLW08]).[1]

Let us now turn to $(-)^{\text{alt}}$. First, recall that universal tree automata are strictly less expressive than non-deterministic ones, so that (with (8.1) and §8.1.4 in mind), one cannot hope for a real dual of our discussion on $(-)^{\text{nd}}$ v.s. $(-)^{\text{v}}$. On the other hand, recall from Chap. 7 that alternating automata are not final objects in translations from MSO, since non-emptiness checks have generally to be performed on non-deterministic automata. Hence, the "good" automata returned by $(-)^{\text{alt}}$ could be $!\varphi^{\text{alt}}$ instead of φ^{alt}, and one may think of doing this replacement

[1] We recall here alternating automata are equipped with a natural notion of additive disjunction (defined similarly as $\oplus$ in §4.3) that we skipped from start in **DZ** for the sake of technical simplicity.

recursively in the translation. This may lead to

$$\begin{array}{lcl}
\top^{\mathrm{n}} & := & \mathbf{I} \\
\bot^{\mathrm{n}} & := & \bot \\
(\mathtt{t}^{\tau} \doteq \mathtt{u}^{\tau})^{\mathrm{n}} & := & \mathtt{t}^{\tau} \doteq \mathtt{u}^{\tau} \\
(\neg\varphi)^{\mathrm{n}} & := & !\varphi^{\mathrm{n}} \multimap \bot \\
(\varphi \wedge \psi)^{\mathrm{n}} & := & !\varphi^{\mathrm{n}} \otimes !\psi^{\mathrm{n}} \\
((\exists x^{\tau})\varphi)^{\mathrm{n}} & := & (\exists x^{\tau})!\varphi^{\mathrm{n}}
\end{array}$$

which is reminiscent of the usual *call-by-name* translation of intuitionistic logic to intuitionistic linear logic (see also Rem. 8.1.7 below).

We shall however adopt neither of these possibilities. Because $\mathsf{MSO}(\mathbf{T})$ is intrinsically a classical logic, we shall instead look for translations of classical logic to intuitionistic linear logic. Two canonical possibilities are the "*call-by-name*" (or "*negative*") T-translation and the "*call-by-value*" (or "*positive*") Q-translation of [DJS97] (see also [LR03, LLW08]). These translations are usually formulated for classical sequents with multiple conclusions and assume in their target the dualities of full classical logic. On the other hand, when restricting to sequents with a single conclusion, both T and Q target linear sequents of the form

$$!(-), \ldots, !(-) \ \vdash \ ?(-)$$

while (in view of §8.1.6) we know from Weak Completeness (Prop. 7.4.4) that linear implications of the form $!(-) \otimes \ldots \otimes !(-) \multimap ?(-)$ are complete for language inclusion w.r.t. our automata based realizability model. Besides, we have seen in Ex. 8.1.3 that $\mathsf{LMSO}(\mathbf{T})$ proves a suitable ?-decorated version of Peirce's Law. We shall therefore devise suitable adaptations of the usual T and Q translations.

Definition 8.1.6. *The translations $(-)^T$ and $(-)^Q$ from $\mathsf{MSO}(\mathbf{T})$-formulae to $\mathsf{LMSO}(\mathbf{T})$-formulae are defined as*

$$\varphi^T \ := \ ?\varphi_T \qquad \text{and} \qquad \varphi^Q \ := \ !\varphi_Q$$

where φ_T and φ_Q are inductively defined as follows:

$$\begin{array}{rcl}
\top_T & := & \mathbf{I} \\
\bot_T & := & \bot \\
(\mathtt{t} \doteq \mathtt{u})_T & := & (\mathtt{t} \doteq \mathtt{u}) \\
(\varphi \wedge \psi)_T & := & !\varphi^T \otimes !\psi^T \\
(\neg\varphi)_T & := & !\varphi^T \multimap ?\bot \\
((\exists x^{\tau})\varphi)_T & := & (\exists x^{\tau})!\varphi^T
\end{array}
\qquad
\begin{array}{rcl}
\top_Q & := & \mathbf{I} \\
\bot_Q & := & \bot \\
(\mathtt{t} \doteq \mathtt{u})_Q & := & (\mathtt{t} \doteq \mathtt{u}) \\
(\varphi \wedge \psi)_Q & := & \varphi^Q \otimes \psi^Q \\
(\neg\varphi)_Q & := & \varphi^Q \multimap ?!\bot \\
((\exists x^{\tau})\varphi)_Q & := & (\exists x^{\tau})\varphi^Q
\end{array}$$

Note that φ^Q is always (weakly) positive in the sense of §8.1.4. Moreover, both translations are semantically correct w.r.t. the standard semantic, in the sense that for all $\mathsf{MSO}(\mathbf{T})$ formulae φ we have

$$\mathcal{L}(\mathcal{A}(\varphi^T)) \ = \ \mathcal{L}(\varphi) \qquad \text{and} \qquad \mathcal{L}(\mathcal{A}(\varphi^Q)) \ = \ \mathcal{L}(\varphi)$$

Remark 8.1.7. *The translation $(-)^T$ is an adaptation of the usual T-translation of [DJS97]. Our version differs from the usual one in the treatment of conjunctions. The T-translation usually assumes an additive conjunction & in the linear system and puts*

$$(\varphi \wedge \psi)_T \ := \ \varphi^T \,\&\, \psi^T$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{!(\varphi_T \multimap \bot),\ \varphi_T \ \vdash\ ?\bot}}{!(\varphi_T \multimap \bot),\ ?\varphi_T \ \vdash\ ?\bot}}{!(\varphi_T \multimap \bot),\ !\varphi^T \ \vdash\ ?\bot}}{!(\varphi_T \multimap \bot) \ \vdash\ !\varphi^T \multimap ?\bot}}{!(\varphi_T \multimap \bot) \ \vdash\ ?(!\varphi^T \multimap ?\bot)}}{!(\varphi_T \multimap \bot) \ \vdash\ !?(!\varphi^T \multimap ?\bot)} \qquad \dfrac{\dfrac{\vdots}{!\overline{\varphi}^T,\ !?(!\varphi^T \multimap ?\bot) \ \vdash\ \varphi^T}}{!\overline{\varphi}^T,\ !?(!\varphi^T \multimap ?\bot),\ !(\varphi_T \multimap \bot) \ \vdash\ \bot}$$

$$\dfrac{\dfrac{!\overline{\varphi}^T,\ !(\varphi_T \multimap \bot),\ !(\varphi_T \multimap \bot) \ \vdash\ \bot}{!\overline{\varphi}^T,\ !(\varphi_T \multimap \bot) \ \vdash\ \bot}}{!\overline{\varphi}^T \ \vdash\ \varphi^T}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{!(\varphi^Q \multimap \bot),\ !\varphi_Q \ \vdash\ \bot}}{!(\varphi^Q \multimap \bot),\ !\varphi_Q \ \vdash\ !\bot}}{!(\varphi^Q \multimap \bot),\ !\varphi_Q \ \vdash\ ?!\bot}}{!(\varphi^Q \multimap \bot) \ \vdash\ \varphi^Q \multimap ?!\bot}}{!(\varphi^Q \multimap \bot) \ \vdash\ !(\varphi^Q \multimap ?!\bot)} \qquad \dfrac{\dfrac{\vdots}{\overline{\varphi}^Q,\ !(\varphi^Q \multimap ?!\bot) \ \vdash\ ?\varphi^Q}}{\overline{\varphi}^Q,\ !(\varphi^Q \multimap ?!\bot),\ !(\varphi^Q \multimap \bot) \ \vdash\ \bot}$$

$$\dfrac{\dfrac{\overline{\varphi}^Q,\ !(\varphi^Q \multimap \bot),\ !(\varphi^Q \multimap \bot) \ \vdash\ \bot}{\overline{\varphi}^Q,\ !(\varphi^Q \multimap \bot) \ \vdash\ \bot}}{\overline{\varphi}^Q \ \vdash\ ?\varphi^Q}$$

Figure 8.5.: The $(-)^T$ and $(-)^Q$ Translations of a Rule for Classical Logic (Ex. 8.1.8).

In presence of the usual isomorphism

$$!(\varphi \,\&\, \psi) \quad \simeq \quad !\varphi \otimes !\psi$$

our version amounts to

$$(\varphi \wedge \psi)_T \quad \simeq \quad !(\varphi^T \,\&\, \psi^T)$$

The $(-)^T$ and $(-)^Q$ translations target linear sequents of the form

$$!\varphi_1^T, \ldots, !\varphi_n^T \ \vdash\ \varphi^T \qquad \text{and} \qquad \varphi_1^Q, \ldots, \varphi_n^Q \ \vdash\ ?\varphi^Q$$

that is

$$!?(\varphi_1)_T, \ldots, !?(\varphi_n)_T \ \vdash\ ?\varphi_T \qquad \text{and} \qquad !(\varphi_1)_Q, \ldots, !(\varphi_n)_Q \ \vdash\ ?!\varphi_Q$$

Example 8.1.8. *We can derive the $(-)^T$ and $(-)^Q$ translations of the rule for classical logic*

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

The derivations are given on Fig. 8.5.

The soundness of the translations is then obtained as usual.

Proposition 8.1.9 (Soundness of $(-)^T$ and $(-)^Q$)**.** *If*

$$\varphi_1, \ldots, \varphi_n \ \vdash_{\mathsf{MSO}(\mathbf{T})}\ \varphi$$

then

$$!\varphi_1^T, \ldots, !\varphi_n^T \ \vdash_{\mathsf{LMSO}(\mathbf{T})}\ \varphi^T \qquad \textit{and} \qquad \varphi_1^Q, \ldots, \varphi_n^Q \ \vdash_{\mathsf{LMSO}(\mathbf{T})}\ ?\varphi^Q$$

Proof. The proof is as usual by induction on derivations, using (8.5) for the Ex Falso rule and Ex. 8.1.8 for the rule for classical logic. The rules for negation of $\mathsf{MSO}(\mathbf{T})$ are particular cases of the usual left rule (combined with Ex Falso) and right rule for implication, and are handled as such. We only explicate the unusual remaining cases, namely the equality rules and the cases of $(\varphi \wedge \psi)^T$. We use (8.4).

- *Equality Rules.*

$$\frac{}{\vdash\ \mathtt{t}^\tau \doteq \mathtt{t}^\tau} \qquad \frac{[\![\mathtt{t}^\tau]\!] = [\![\mathtt{u}^\tau]\!]}{\vdash\ \mathtt{t}^\tau \doteq \mathtt{u}^\tau} \qquad \frac{}{\mathtt{t}^\tau \doteq \mathtt{u}^\tau,\ \varphi[\mathtt{t}^\tau/x^\tau]\ \vdash\ \varphi[\mathtt{u}^\tau/x^\tau]}$$

  The first two rules are immediate in the case of both $(-)^T$ and $(-)^Q$. The substitutivity rule is trivial for $(-)^Q$. In the case of $(-)^T$, the result follows from

$$\cfrac{\cfrac{\cfrac{\cfrac{}{\mathtt{t} \doteq \mathtt{u},\ ?\varphi_T[\mathtt{t}/x]\ \vdash\ ?\varphi_T[\mathtt{u}/x]}}{\mathtt{t} \doteq \mathtt{u},\ !?\varphi_T[\mathtt{t}/x]\ \vdash\ ?\varphi_T[\mathtt{u}/x]}}{?(\mathtt{t} \doteq \mathtt{u}),\ !?\varphi_T[\mathtt{t}/x]\ \vdash\ ?\varphi_T[\mathtt{u}/x]}}{!?(\mathtt{t} \doteq \mathtt{u}),\ !?\varphi_T[\mathtt{t}/x]\ \vdash\ ?\varphi_T[\mathtt{u}/x]}$$

- $(-)^T$*-Translation of the Left* $\wedge$*-Rule:*

$$\frac{\overline{\varphi},\ \varphi_0,\ \varphi_1\ \vdash\ \psi}{\overline{\varphi},\ \varphi_0 \wedge \varphi_1\ \vdash\ \psi}$$

  The result follows from

$$\cfrac{\cfrac{\cfrac{\cfrac{\vdots}{!?\overline{\varphi}_T,\ !?(\varphi_0)_T,\ !?(\varphi_1)_T\ \vdash\ ?\psi_T}}{!?\overline{\varphi}_T,\ !?(\varphi_0)_T \otimes !?(\varphi_1)_T\ \vdash\ ?\psi_T}}{!?\overline{\varphi}_T,\ ?(!?(\varphi_0)_T \otimes !?(\varphi_1)_T)\ \vdash\ ?\psi_T}}{!?\overline{\varphi}_T,\ !?(!?(\varphi_0)_T \otimes !?(\varphi_1)_T)\ \vdash\ ?\psi_T}$$

- $(-)^T$*-Translation of the Right* $\wedge$*-Rule*

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

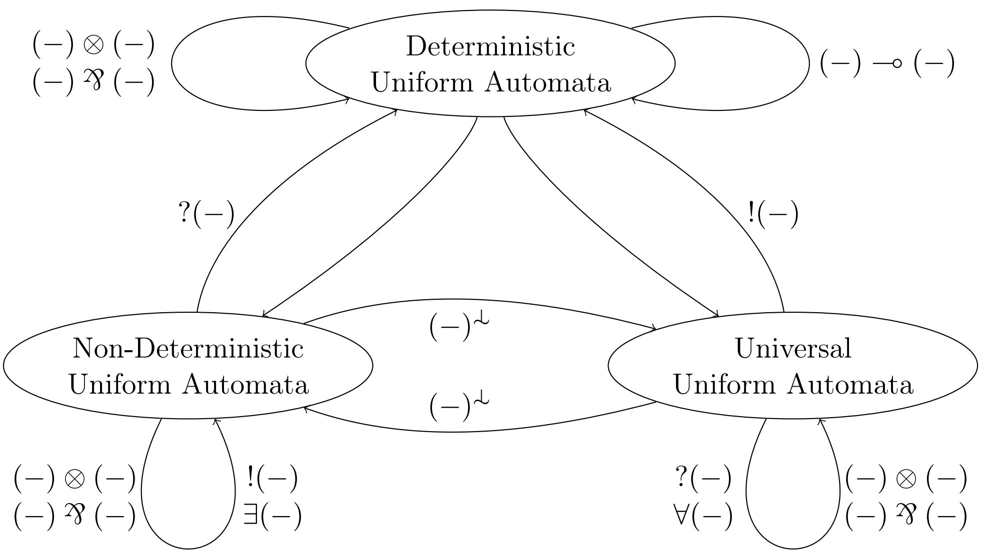
  The result follows from

$$\cfrac{\cfrac{\cfrac{\cfrac{\vdots}{!?\overline{\varphi}_T\ \vdash\ ?\varphi_T}}{!?\overline{\varphi}_T\ \vdash\ !?\varphi_T} \qquad \cfrac{\cfrac{\vdots}{!?\overline{\psi}_T\ \vdash\ ?\psi_T}}{!?\overline{\psi}_T\ \vdash\ !?\psi_T}}{!?\overline{\varphi}_T,\ !?\overline{\psi}_T\ \vdash\ !?\varphi_T \otimes !?\psi_T}}{!?\overline{\varphi}_T,\ !?\overline{\psi}_T\ \vdash\ ?(!?\varphi_T \otimes !?\psi_T)}$$

□

8.2. The Case of Infinite Words

This Section essentially concerns the specialization of §8.1 to the case of ω-words. If we step back to the realizability model defined in Chap. 7, we might recall from §7.5 that when $\mathfrak{D} = \mathbf{1}$, picture (8.1) can be refined to (7.12), in which the exponential modality $?(-)$ amounts to McNaughton's Determinization Theorem [McN66] (Thm. 2.1.2):



(8.7)

In particular, the classes of deterministic, non-deterministic and universal automata all have the same expressive power, and as we shall see below, we can factor translations of MSO to automata in a system which obeys a strong polarity policy. A important consequence of this is that we can devise a version of LMSO in which exponentials are restricted to the polarities depicted in (8.7). But now recall from §7.4 that for an automaton $\mathcal{A}$ with set of P moves U, the automaton $!\mathcal{A}$ has set of P moves $U^{Q_\mathcal{A}}$. Hence, if in (8.7) the modality $!(-)$ is interpreted as the identity on non-deterministic automata (and dualy for $?(-)$), we can avoid the states of automata to appear in the moves of realizers. In other words, realizers can then be extracted from proofs without any concrete appeal to McNaughton's Theorem. Building on Siefkes Theorem [Sie70] (the complete axiomatization of MSO on ω-words, see §3.1.4), it follows that we can give a "*Safraless*" approach to Church's Synthesis.

This Section is organized as follows. We first define in §8.2.1 the logic MSO($\mathbf{M}$), a version of "MSO($\mathbf{T}$) on ω-words" but with slight modifications essentially motivated by (8.7). Then, §8.2.2–8.2.4 are devoted Church's Synthesis. Church's Synthesis problem is stated in §8.2.2, while §8.2.2 presents the historical solution based on the Büchi-Landweber Theorem [BL69] (Thm. 2.4.1), that we reformulate in the setting of MSO($\mathbf{M}$), and §8.2.4 presents by now well-established alternative approaches to Church's Synthesis. We then present in §8.2.5 the system LMSO($\mathbf{M}$), the linear counterpart of MSO($\mathbf{M}$), and recapitulates its main properties in terms of extraction of realizers from proofs. Finally, §8.2.6 is concerned with completeness. We first briefly discuss a system MSO($\mathbf{M}$)$^\omega$, which is essentially the union of MSO($\mathbf{M}$) and MSO$^\omega$ (§3.1.4), as well as its linear counterpart LMSO($\mathbf{M}$)$^\omega$. In this setting Siefkes's Theorem has two important consequences. The first one is that LMSO($\mathbf{M}$)$^\omega$ is complete w.r.t. Church's Synthesis. The second one is the fact that thanks to an adaptation of a usual linear variant [dP87] of Gödel's "*Dialectica*" functional interpretation (see e.g. [AF98, Koh08]), one can show that LMSO($\mathbf{M}$)$^\omega$

is actually a complete theory, in the sense that for all closed formula φ, it either proves φ or $\varphi \multimap \bot$ (a result which by now relies on the restriction of $\mathsf{LMSO}(\mathbf{M})$ to polarized exponentials).

The material presented in this Section is detailed in [PR18b, PR19, Pra19], and we shall only outline these developments.

8.2.1. The Logic $\mathsf{MSO}(\mathbf{M})$. When specializing the base category $\mathbf{T}$ (Def. 4.2.1, §4.2) to ω-words (*i.e.* when $\mathfrak{D} \simeq \mathbf{1}$), we obtain the category $\mathbf{S}$ of alphabets and causal functions (§2.3). As a consequence, *finite-state* $\mathbf{T}$ morphisms (§6.5.2) are now finite-state causal "*Mealy*" functions, *i.e.* morphisms of the category $\mathbf{M}$ (§2.3). We assume a corresponding term language, obtained from the obvious modifications to §8.1.1.

We may have called $\mathsf{MSO}(\mathbf{M})$ the corresponding version of $\mathsf{MSO}(\mathbf{T})$. Actually, following §7.5 and [PR18b, PR19], it makes sense to also assume implication $\to$, disjunction $\vee$ and universal quantifications $(\forall x^\tau)$ as primitive connectives of $\mathsf{MSO}(\mathbf{M})$. We thus officially take the following formulae for $\mathsf{MSO}(\mathbf{M})$:

$$\varphi, \psi \quad ::= \quad \mathtt{t}^\tau \doteq \mathtt{u}^\tau \quad | \quad \top \quad | \quad \bot \quad | \quad \varphi \wedge \psi \quad | \quad \psi \to \varphi \quad | \quad \varphi \vee \psi \quad | \quad (\exists x^\tau)\varphi \quad | \quad (\forall x^\tau)\varphi$$

Similarly as for $\mathsf{MSO}(\mathbf{T})$ v.s. $\mathsf{MSO}(\mathfrak{D})$, the logic $\mathsf{MSO}(\mathbf{M})$ is definable in MSO^ω w.r.t. its standard model (see §3.1.4). We briefly discuss this point in §8.2.6 below.

Notation 8.2.1. *From now on, we shall assume that* $\mathsf{MSO}(\mathbf{M})$ *is equipped with a notion of quantification over* individuals *(i.e. positions in ω-words, aka natural numbers), denoted $\forall x^\iota$ and $\exists x^\iota$. As usual with automata (see e.g. [Tho97, Wal02]), individuals are actually singleton predicates, i.e. streams $B \in \mathbf{2}^\omega \simeq \mathcal{P}(\mathbb{N})$ which happen to be characteristic maps of singletons subsets of $\mathbb{N}$.*

There are different ways of representing such individuals in $\mathsf{MSO}(\mathbf{M})$. *One possibility (that we followed in [PR18b, PR19] for technical reasons), is to assume a specific atomic predicate $\mathsf{N}(x^o)$ such that $\mathsf{N}(B)$ holds iff $B \in \mathbf{2}^\omega \simeq \mathcal{P}(\mathbb{N})$ is a singleton. An other possibility is to note that set containment $x^o \subseteq y^o$ is definable in* $\mathsf{MSO}(\mathbf{M})$ *as an atomic formula of the form $\mathsf{Incl}(x^o, y^o) \doteq \mathbf{0}^\omega$, and then (following e.g. [Wal02]) to represent singletons sets as those non-empty sets which have no non-empty proper subsets.*

We similarly assume available a membership predicate $x^o \mathrel{\dot\in} y^o$ where x^o is intended to be relativized to individuals, and may write $x(k)$ for $k \mathrel{\dot\in} x$.

8.2.2. Church's Synthesis. Church's synthesis [Chu57] consists in the automatic extraction of Mealy machines from input-output specifications, typically presented as $\mathsf{MSO}(\mathbf{M})$ sentences of the form

$$(\forall x^\sigma)(\exists y^\tau)\varphi(x; y) \tag{8.8}$$

A specification as in (8.8) is realized in the sense of Church's Synthesis by a (finite-state) causal $F : [\![\sigma]\!] \to_{\mathbf{M}} [\![\tau]\!]$ when $\varphi(B, F(B))$ holds for all $B \in [\![\sigma]\!]^\omega$.

Example 8.2.2. *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*

$$\Phi(B, C) \quad \overset{def.}{\Longleftrightarrow} \quad \left\{ \begin{array}{lll} & \forall n^\iota (n \mathrel{\dot\in} C & \to \quad n \mathrel{\dot\in} B) \\ \wedge & \forall n^\iota (n \mathrel{\dot\in} C & \to \quad n+1 \mathrel{\dot\notin} C) \\ \wedge & (\exists^\infty n)(n \mathrel{\dot\in} B) & \to \quad (\exists^\infty n)(n \mathrel{\dot\in} C) \end{array} \right.$$

In words, the relation $\Phi(B, C)$ imposes $C \in \mathbf{2}^\omega \simeq \mathcal{P}(\mathbb{N})$ not to contain n whenever B does not contain n, C not to contain two consecutive positions, and moreover C to be infinite whenever *so is B. It is easy to see that this specification is realized in the sense of Church's Synthesis by the finite-state causal function of Ex. 2.3.1.(c) (see also Ex. 6.5.3, §6.5.1), that we repeat here:*

$$F(B)(n) \quad = \quad \begin{cases} 0 & \text{if } n = 0 \text{ or } F(B)(n-1) = 1 \\ B(n) & \text{otherwise} \end{cases}$$

It is important to note here that there are true $\mathsf{MSO}(\mathbf{M})$ sentences of the form (8.8) which can be realized by no continuous (and hence no computable) stream functions. For instance, take $\varphi(x^o; y^o)$ to hold if either $y = \mathbf{0}^\omega$ and x is empty or if $y = \mathbf{1}^\omega$ and x is non-empty. Hence, we can see Church's Synthesis as a decision problem for a form of constructivity in MSO. This is made technically precise in §8.2.5 and §8.2.6 thanks to linear logic.

8.2.3. Büchi-Landweber Theorem. Traditional solutions to Church's synthesis turn specifications to infinite two-player games with ω-regular winning conditions (see e.g. [Tho08, Fin16]). Consider an $\mathsf{MSO}(\mathbf{M})$ formula $\varphi(u^\tau, x^\sigma)$ with no free variable other than u, x. We see this formula as defining a full positive game $\mathcal{G}(\varphi)(u^\tau, x^\sigma)$ with P moves $\mathtt{u} \in [\![\tau]\!]$ and O moves $\mathtt{x} \in [\![\sigma]\!]$. Hence P begins, and then the two players alternate, producing an infinite play of the form

$$\chi \quad := \quad \mathtt{u}_0\mathtt{x}_0 \cdots \mathtt{u}_n\mathtt{x}_n \cdots \quad \simeq \quad ((\mathtt{u}_k)_k, (\mathtt{x}_k)_k) \in [\![\tau]\!]^\omega \times [\![\sigma]\!]^\omega$$

The play χ is winning for P if $\varphi((\mathtt{u}_k)_k, (\mathtt{x})_k)$ holds. Otherwise χ is winning for O. Similarly as in §2.3, strategies for P resp. O in this game are functions

$$[\![\sigma]\!]^* \longrightarrow [\![\tau]\!] \qquad \text{resp.} \qquad [\![\tau]\!]^+ \longrightarrow [\![\sigma]\!] \quad \simeq \quad [\![\tau]\!]^* \longrightarrow [\![\sigma]\!]^{[\![\tau]\!]}$$

and finite-state strategies are represented by f.s. eager functions. In particular, a realizer of $(\forall x^\sigma)(\exists u^\tau)\varphi(u, x)$ in the sense of Church is a f.s. P-strategy in

$$\mathcal{G}\big(\varphi((u)x, x)\big)\big(u^{(\tau)\sigma}, x^\sigma\big)$$

Most approaches to Church's synthesis reduce to Büchi-Landweber Theorem [BL69] (see Thm. 2.4.1). In the context of this Section, we consider the Büchi-Landweber Theorem in following form. Note that an O-strategy in the game $\mathcal{G}(\varphi)(u^\tau, x^\sigma)$ is a P-strategy in the game $\mathcal{G}\big(\neg\varphi(u, (x)u)\big)\big(x^{(\sigma)\tau}, u^\tau\big)$.

Theorem 8.2.3 ([BL69])**.** *Let $\varphi(u^\tau, x^\sigma)$ be an $\mathsf{MSO}(\mathbf{M})$-formula with only u, x free. Then either there is an eager term $\mathtt{u}(x)$ of arity $(\sigma; \tau)$ such that $\models (\forall x)\varphi(\mathtt{u}(x), x)$ or there is an eager term $\mathtt{x}(u)$ of arity $(\tau; (\sigma)\tau)$ such that $\models (\forall u)\neg\varphi(u, \mathrm{ev}(\mathtt{x})(u))$. It is decidable which case holds and the terms are computable from φ.*

8.2.4. Some Other Approaches to Church's Synthesis. An other general solution to Church's Synthesis for $\mathsf{MSO}(\mathbf{M})$ goes via infinite trees [Rab72] (see also [KPV06]), noting that an eager causal function from say $[\![\sigma]\!]$ to $[\![\tau]\!]$ can be represented by an infinite $[\![\tau]\!]$-labeled $[\![\sigma]\!]$-ary tree.

However, the historical approaches to these solutions (either directly using Büchi-Landweber or going via MSO on infinite trees) do not directly lead to applicable algorithms. The reason is that in both cases, one relies on McNaughton's Determinization Theorem [McN66]. The best known (and possible) constructions for McNaughton's Theorem (such as *Safra's trees*, see e.g. [GTW02]) give deterministic Muller automata with $2^{O(n\log(n))}$ states from non-deterministic Büchi automata (NBA) with n states. This is actually the same asymptotic complexity as for Büchi's Theorem [Büc62] (see e.g. [Tho97]), but while the latter is amenable to tractable implementations (see e.g. [FL12]), this is to our knowledge not yet the case for McNaughton's Theorem (see e.g. [KPV06, FJR11, BJP+12, Fin16]). Also, the states of automata obtained from McNaughton's Theorem have a complex structure, making it difficult to implement subsequent algorithms (e.g. game solving). These difficulties are worsened by the theoretical complexity of translating MSO formulae to automata, which is known to be non-elementary (see e.g. [GTW02]).

A well studied particular case of Church synthesis starts from formulae of LTL, a modal logic corresponding to an expressive first-order fragment of MSO. The translation of LTL-formulae to NBA's is exponential (see e.g. [BK08]), and Church's synthesis for LTL is known to be 2EXPTIME complete [Ros92].

Several approaches (coined "*Safraless*") considered synthesis procedures which avoid McNaughton's Theorem. First, let us mention [BJP+12], who reduces synthesis for a fragment of LTL (including past formulae) to solving so-called $\mathbf{GR}(1)$ conditions, which are of the form:

$$\bigwedge_{1\leq i\leq n} \exists^\infty k.\, \mathsf{A}_i(\overline{y(k)}) \quad \longrightarrow \quad \bigwedge_{1\leq j\leq m} \exists^\infty k.\, \mathsf{G}_j(\overline{z(k)}) \tag{8.9}$$

where each A_i (resp. G_j) is a propositional formula with p (resp. q) variables, representing a subset of Σ (resp. Γ). It is shown in [BJP+12] that conditions of the form (8.9) can be solved in time $O(nm(2^{p+q})^2)$.

An other trend of Safraless approaches is to use upper bounds on the size of automata provided by McNaughton's Theorem in order to bound the search space for finite-state realizers. Such approaches were pioneered by [KV05, KPV06] using tree automata, and have been developed for automata on ω-words e.g.in [FJR11].

The idea of [KV05, KPV06, FJR11] is to reduce specifications to formal duals of NBA's, namely universal automata equipped with a co-Büchi acceptance condition (*i.e.* specifying a set of states which should *not* be visited infinitely often in an accepting run), that we call UCAs. Formally, a UCA $\mathcal{U}$ over Σ (notation $\mathcal{U} : \Sigma$) has the form

$$\mathcal{U} \quad = \quad (Q_{\mathcal{U}}\,,\, q^{\imath}_{\mathcal{U}}\,,\, \mathbf{1}\,,\, \mathcal{U}_{\mathsf{O}}\,,\, \partial_{\mathcal{U}}\,,\, F_{\mathcal{U}})$$

where $\mathcal{U}_{\mathsf{O}}$ is the alphabet of O moves and $F_{\mathcal{U}} \subseteq Q_{\mathcal{U}}$ is the set of final states. A *run* $\rho \in \mathcal{U}_{\mathsf{O}}^\omega$ is *accepting* over $B \in \Sigma^\omega$ if there are at most finitely many final states in the corresponding sequence of states $(q_n)_n \in Q_{\mathcal{U}}^\omega$. $\mathcal{U}$ accepts $B \in \Sigma^\omega$ if every run of $\mathcal{U}$ is accepting on B. So, as an $\mathsf{MSO}(\mathbf{M})$ formula, acceptance of a UCA $\mathcal{U}$ over $B \in \Sigma^\omega$ is expressed by a formula of the form

$$(\forall \rho^{\mathcal{U}_{\mathsf{O}}})(\forall^\infty n)\neg \mathsf{F}[\mathtt{t}(B,\rho)(n)]$$

We say that a f.s. causal $F : \Gamma \to_{\mathbf{M}} \Sigma$ *realizes* a UCA $\mathcal{U} : \Gamma \times \Sigma$ if $\mathcal{U}$ accepts the each pair $\langle B, F(B)\rangle$ for $B \in \Gamma^\omega$.

The method underlying [FJR11] is to translate the negation of a specification into an NBA $\mathcal{B}$, and then take as UCA the dual of $\mathcal{B}$. In our setting, following §7.5, the universal co-Büchi $\mathcal{U}$ is obtained from a non-deterministic Büchi $\mathcal{B}$ by taking as O-moves the set of P-moves of $\mathcal{B}$. The crucial observation is [FJR11, Thm. 2], that we state as follows in our setting:

Theorem 8.2.4 ([FJR11])**.** *Consider simple types σ, τ and universal co-Büchi automaton $\mathcal{U} : [\![\sigma]\!] \times [\![\tau]\!]$. For each $K \geq 0$, consider the following $\mathsf{MSO}(\mathbf{M})$-formula $\varphi_K(x^\sigma; y^\tau)$:*

$$(\forall \rho^{\mathcal{U}_{\mathsf{O}}})\big(\forall n \mathrel{\dot\in} \{m \mid m \geq K\}\big)\neg \mathsf{F}[\mathtt{t}(x,y,\rho)(n)] \tag{8.10}$$

Then every realizer of $\varphi_K(x^\sigma; y^\tau)$ is a realizer of $\mathcal{U}$. Moreover, there is some K exponential in the size of $\mathcal{U}$ such that $\mathcal{U}$ is realizable iff $\varphi_K(x^\sigma; y^\tau)$ is realizable.

The interest of Thm. 8.2.4 is that for fixed $K \geq 0$, the formula $\varphi_K(x^\sigma, y^\tau)$ can be turned, thanks to a simple powerset construction, to a deterministic automaton specifying a *safety* game, which is in general much simpler to solve than a parity game.

An other trend of approaches to solving games for Church's Synthesis goes via "*Good-for-Games*" automata [HP06]. We come back on this in Ex. 8.2.10 below.

8.2.5. The Logic $\mathsf{LMSO}(\mathbf{M})$. A first possible definition of $\mathsf{LMSO}(\mathbf{M})$ could be the direct adaptation of $\mathsf{LMSO}(\mathbf{T})$ to ω-words. In addition, since automata on ω-words are equipped with a multiplicative disjunction $⅋$, we might as well assume this connective in $\mathsf{LMSO}(\mathbf{M})$. This might lead to

$$\varphi, \psi \quad ::= \quad \mathbf{I} \mid \bot \mid \mathtt{t}^\tau \doteq \mathtt{u}^\tau \mid \varphi \multimap \psi \mid \varphi \otimes \psi \mid \varphi ⅋ \psi \mid\ !\varphi \mid (\exists x^\tau)\varphi \mid (\forall x^\tau)\varphi$$

However, as discussed in §7.5, since the automaton $\perp$ on ω-words is deterministic, picture (8.1) can be refined to (8.7). In particular, the classes of deterministic, non-deterministic and universal automata all have the same expressive power, and as we shall below, we can factor translations of $\mathsf{MSO}(\mathbf{T})$ to automata in a system which obeys a strong polarity policy.

The idea is to have three categories of polarized formulae, namely the *positive* formulae $(\varphi^+, \psi^+, \dots)$, the *negative* formulae $(\varphi^-, \psi^-, \dots)$ and the *deterministic* formulae $(\varphi^\pm, \psi^\pm, \dots)$ corresponding respectively to non-deterministic, uniform and deterministic automata. Moreover, recall from §7.5 that on non-deterministic automata, the exponential modality $?(-)$ amounts to McNaughton's Determinization Theorem [McN66]. We consider $?(-)$ as a primitive in this case (and so replaced the dashed arrow in (7.12) by a solid one in (8.7)). In addition, since $!\varphi^+$ and $?\varphi^-$ are always respectively positive and negative, it makes sense to officially allow these formulae in our polarized fragment, having in mind the following interpretation:

$$\mathcal{A}(!\varphi^+)(\overline{x}) \quad := \quad \mathcal{A}(\varphi^+)(\overline{x}) \qquad \text{and} \qquad \mathcal{A}(?\varphi^-)(\overline{x}) \quad := \quad \mathcal{A}(\varphi^-)(\overline{x})$$

We thus arrive at the following *polarized* fragment of $\mathsf{LMSO}(\mathbf{T})$:

$$\begin{array}{lll}
\varphi^\pm, \psi^\pm & ::= & \mathbf{I} \mid \bot \mid \mathtt{t}^\tau \doteq \mathtt{u}^\tau \mid !(\varphi^-) \mid ?(\varphi^+) \mid \varphi^\pm \otimes \psi^\pm \mid \varphi^\pm ⅋ \psi^\pm \mid \varphi^\pm \multimap \psi^\pm \\
\varphi^+, \psi^+ & ::= & \varphi^\pm \mid !(\varphi^+) \mid (\exists x^\sigma)\varphi^+ \mid \varphi^+ \otimes \psi^+ \mid \varphi^+ ⅋ \psi^+ \mid \varphi^- \multimap \psi^+ \\
\varphi^-, \psi^- & ::= & \varphi^\pm \mid ?(\varphi^-) \mid (\forall x^\sigma)\varphi^- \mid \varphi^- \otimes \psi^- \mid \varphi^- ⅋ \psi^- \mid \varphi^+ \multimap \psi^-
\end{array}$$

Note that in contrast with the case of infinite trees (§8.1.4) this fragment is "*strongly*" or hereditarily polarized as exponentials are only applied to polarized formulae. Moreover, making the aforementioned modifications to §8.1.4, as expected:

- if φ is positive then the automaton $\mathcal{A}(\varphi)$ is non-deterministic,
- if φ is negative then the automaton $\mathcal{A}(\varphi)$ is universal,
- if φ is deterministic then the automaton $\mathcal{A}(\varphi)$ is deterministic.

Let us now reconsider the interpretations $(-)^T$ and $(-)^Q$ in this context. First, we have to extend Def. 8.1.6 to the language of $\mathsf{MSO}(\mathbf{M})$. Following [LLW08] (but keeping our specific interpretations of $\wedge$ and $\vee$), we arrive at

$$\begin{array}{rcl} \top_T &:=& \mathbf{I} \\ \bot_T &:=& \bot \\ (\mathtt{t} \doteq \mathtt{u})_T &:=& (\mathtt{t} \doteq \mathtt{u}) \\ (\varphi \wedge \psi)_T &:=& !\varphi^T \otimes !\psi^T \\ (\varphi \vee \psi)_T &:=& \varphi^T \parr \psi^T \\ (\varphi \to \psi)_T &:=& !\varphi^T \multimap \psi^T \\ ((\exists x^\tau)\varphi)_T &:=& (\exists x^\tau)!\varphi^T \\ ((\forall x^\tau)\varphi)_T &:=& (\forall x^\tau)\varphi^T \end{array} \qquad \begin{array}{rcl} \top_Q &:=& \mathbf{I} \\ \bot_Q &:=& \bot \\ (\mathtt{t} \doteq \mathtt{u})_Q &:=& (\mathtt{t} \doteq \mathtt{u}) \\ (\varphi \wedge \psi)_Q &:=& \varphi^Q \otimes \psi^Q \\ (\varphi \vee \psi)_Q &:=& ?\varphi^Q \parr ?\psi^Q \\ (\varphi \to \psi)_Q &:=& \varphi^Q \multimap ?\psi^Q \\ ((\exists x^\tau)\varphi)_Q &:=& (\exists x^\tau)\varphi^Q \\ ((\forall x^\tau)\varphi)_Q &:=& (\forall x^\tau)?\varphi^Q \end{array}$$

where $\varphi^T = ?\varphi_T$ and $\varphi^Q = !\varphi_Q$. A crucial observation is that $(-)^T$ and $(-)^Q$ are now both polarized:

Proposition 8.2.5. *Given an* $\mathsf{MSO}(\mathbf{M})$ *formula* φ*, the* $\mathsf{LMSO}(\mathbf{M})$ *formula* φ^T *is negative and the* $\mathsf{LMSO}(\mathbf{M})$ *formula* φ^Q *is positive.*

Proof. A trivial induction on formulae. Note that φ_T and φ_Q can be either deterministic, positive of negative, while $?\varphi_T$ is always negative (or deterministic) and $!\varphi_Q$ is always positive (or deterministic). □

We thus have a semantically well-behaved polarized fragment of $\mathsf{LMSO}(\mathbf{M})$. Following [PR18b, PR19], we shall actually officially define $\mathsf{LMSO}(\mathbf{M})$ as the extension of the polarized fragment with multiplicative connectives and quantifications:

$$\varphi, \psi \quad ::= \quad \varphi^+ \mid \varphi^- \mid \varphi \multimap \psi \mid \varphi \otimes \psi \mid \varphi \parr \psi \mid (\exists x^\tau)\varphi \mid (\forall x^\tau)\varphi$$

We insist that we do not allow unpolarized exponentials in $\mathsf{LMSO}(\mathbf{M})$. To each $\mathsf{LMSO}(\mathbf{M})$ formula φ we associate an $\mathsf{MSO}(\mathbf{M})$ formula $\lfloor\varphi\rfloor$ obtained by replacing the linear connectives by their classical counterpart:

$$\begin{array}{rcl} \lfloor \mathbf{I} \rfloor &:=& \top \\ \lfloor \varphi \otimes \psi \rfloor &:=& \lfloor\varphi\rfloor \wedge \lfloor\psi\rfloor \\ \lfloor !\varphi \rfloor &:=& \lfloor\varphi\rfloor \\ \lfloor ?\varphi \rfloor &:=& \lfloor\varphi\rfloor \end{array} \qquad \begin{array}{rcl} \lfloor \bot \rfloor &:=& \bot \\ \lfloor \varphi \parr \psi \rfloor &:=& \lfloor\varphi\rfloor \vee \lfloor\psi\rfloor \\ \lfloor (\exists x^\tau)\varphi \rfloor &:=& (\exists x^\tau)\lfloor\varphi\rfloor \\ \lfloor (\forall x^\tau)\varphi \rfloor &:=& (\forall x^\tau)\lfloor\varphi\rfloor \end{array} \qquad \begin{array}{rcl} \lfloor \mathtt{t}^\tau \doteq \mathtt{u}^\tau \rfloor &:=& \mathtt{t}^\tau \doteq \mathtt{u}^\tau \\ \lfloor \psi \multimap \varphi \rfloor &:=& \lfloor\psi\rfloor \to \lfloor\varphi\rfloor \end{array}$$

Polarized $\mathsf{LMSO}(\mathbf{M})$ formulae have their expected classical meaning, in the following sense.

Proposition 8.2.6. *If* φ *is a polarized formula of* $\mathsf{LMSO}(\mathbf{M})$*, then*

$$\mathcal{L}(\mathcal{A}(\varphi)) \quad = \quad \mathcal{L}(\lfloor\varphi\rfloor)$$

Note that $\mathsf{LMSO}(\mathbf{M})$ allow non-standard quantifier alternations, as in

$$(\forall x^\tau)(\exists y^\sigma)\varphi(x; y)$$

Such unpolarized formulae are particularly relevant in the context of Church's Synthesis. We say that an LMSO formula $\varphi(\overline{x}^{\overline{\sigma}})$ is realizable if there is a winning P-strategy in the game

$$\mathbf{I} \quad \multimap_{\mathsf{DialAut}[\![\overline{\sigma}]\!]} \quad \mathcal{A}(\varphi)(\overline{x}^{\overline{\sigma}})$$

or equivalently if $\mathcal{L}(\mathcal{A}(\varphi)(\overline{x}^{\overline{\sigma}})) = [\![\overline{\sigma}]\!]^{\mathfrak{D}^*}$.

Proposition 8.2.7. *Given an* $\mathsf{MSO}(\mathbf{M})$ *formula* $\varphi(x^\tau; y^\sigma)$ *(with free variables as shown), the following are equivalent:*

- *The Church's Synthesis problem for* $(\forall x^\tau)(\exists y^\sigma)\varphi(x; y)$ *is realizable.*
- *The* $\mathsf{LMSO}(\mathbf{M})$ *formula* $(\forall x^\tau)(\exists y^\sigma)!\varphi^T(x; y)$ *is realizable.*
- *The* $\mathsf{LMSO}(\mathbf{M})$ *formula* $(\forall x^\tau)(\exists y^\sigma)?\varphi^Q(x; y)$ *is realizable.*

Recall that since $\mathsf{MSO}(\mathbf{M})$ has true but unrealizable $\forall\exists$-statements, we cannot deduce anything w.r.t. Church's Synthesis from the validity of $\forall\exists$-statements of $\mathsf{MSO}(\mathbf{M})$. Hence $\mathsf{LMSO}(\mathbf{M})$ allows to speak of realizability for Church's Synthesis in a much more precise manner than $\mathsf{MSO}(\mathbf{M})$. Besides, the realizability semantics of $\mathsf{LMSO}(\mathbf{M})$ is necessarily non-standard, as witnessed by the following.

Example 8.2.8 (Functional Linear Choice ([PR18b])). *The following scheme is realizable:*

$$(\forall x^\sigma)(\exists y^\tau)\varphi(x, y) \quad \multimap \quad (\exists f^{(\tau)\sigma})(\forall x^\sigma)\varphi(x, (f)x) \tag{LAC}$$

Note that the formulae of Ex 8.2.8 are not polarized. In particular, the $\lfloor - \rfloor$ translation of Ex. 8.2.8 is in general false. The following surprising fact is due to Pierre Pradic.

Example 8.2.9 ([PR18b]). *The following scheme is realizable*

$$((\varphi \multimap \bot) \multimap \bot) \quad \multimap \quad \varphi$$

Example 8.2.10 (Good-for-Games Automata). Good for Games *automata were introduced in [HP06] as a particular class of non-deterministic automata having good compositionality properties with games. Several equivalent definitions have been introduced then, see e.g. [BL19]. In particular* history determinism *[Col09] (actually a variation of the notion of* guidable *automata [CL08], see also §7.4.1), is the property for a non-deterministic automaton that non-determinism can solved by a causal (and thus f.s. causal) function of the input.*

Good-for-Games automata are interesting for Church's Synthesis because they allow to avoid McNaughton's Determinization Theorem when solving games. Besides, Good-for-Games coBüchi automata are exponentially smaller than their deterministic counterparts [KS15, BL19].

In the case of non-deterministic uniform automata, this property can be expressed within the language of $\mathsf{LMSO}(\mathbf{M})$*. Consider a non-deterministic automaton* $\mathcal{A} : [\![\sigma]\!]$ *with alphabet of* P *moves* U*. Following Ex. 7.2.10, we can represent* $\mathcal{A}$ *as an* $\mathsf{LMSO}(\mathbf{M})$*-formula* $(\exists u^U)\delta^\pm(x^\sigma, u^U)$*, where* $\delta^\pm$ *is deterministic. Then,* $\mathcal{A}$ *is history deterministic if and only if the following* $\mathsf{LMSO}(\mathbf{M})$ *formula is realizable:*

$$(\forall x^\sigma)\big(?(\exists u^U)\delta^\pm(x^\sigma, u^U) \quad \multimap \quad (\exists u^U)\delta^\pm(x^\sigma, u^U)\big)$$

Note that $?(\exists u)\delta^\pm$ *is deterministic, so that witnesses for* $(\exists u^U)$ *in the conclusion actually only depend on* x^σ*.*

To summarize, $\mathsf{LMSO}(\mathbf{M})$ has a semantically well-behaved polarized fragment, surrounded by a non-standard part with a relevant semantics for Church's Synthesis. We thus find it interesting to devise a deduction system for $\mathsf{LMSO}(\mathbf{M})$. But this comes with a caveat. $\mathsf{LMSO}(\mathbf{M})$ is an intuitionistic logic (similarly as $\mathsf{LMSO}(\mathbf{T})$), which is nevertheless equipped with a multiplicative disjunction, together with a primitive $?(-)$ exponential. Hence we would like deduction for $\mathsf{LMSO}(\mathbf{M})$ to use sequents with multiple conclusions on the right of the $\vdash$. But for such a

$$
\dfrac{}{\varphi \vdash \varphi}
\qquad
\dfrac{\overline{\varphi} \vdash \gamma, \overline{\varphi}' \qquad \overline{\psi}, \gamma \vdash \overline{\psi}'}{\overline{\varphi}, \overline{\psi} \vdash \overline{\varphi}', \overline{\psi}'}
\qquad
\dfrac{\overline{\varphi}, \varphi, \psi, \overline{\psi} \vdash \overline{\varphi}'}{\overline{\varphi}, \psi, \varphi, \overline{\psi} \vdash \overline{\varphi}'}
\qquad
\dfrac{\overline{\varphi} \vdash \overline{\varphi}', \varphi, \psi, \overline{\psi}'}{\overline{\varphi} \vdash \overline{\varphi}', \psi, \varphi, \overline{\psi}'}
$$

$$
\dfrac{\overline{\varphi} \vdash \overline{\psi}}{\overline{\varphi}, \mathbf{I} \vdash \overline{\psi}}
\qquad
\dfrac{}{\vdash \mathbf{I}}
\qquad
\dfrac{\overline{\varphi}, \varphi_0, \varphi_1 \vdash \overline{\varphi}'}{\overline{\varphi}, \varphi_0 \otimes \varphi_1 \vdash \overline{\varphi}'}
\qquad
\dfrac{\overline{\varphi} \vdash \varphi, \overline{\varphi}' \qquad \overline{\psi} \vdash \psi, \overline{\psi}'}{\overline{\varphi}, \overline{\psi} \vdash \varphi \otimes \psi, \overline{\varphi}', \overline{\psi}'}
\qquad
\dfrac{\overline{\varphi}, \varphi \vdash \psi}{\overline{\varphi} \vdash \varphi \multimap \psi}
$$

$$
\dfrac{}{\bot \vdash}
\qquad
\dfrac{\overline{\varphi} \vdash \overline{\psi}}{\overline{\varphi} \vdash \bot, \overline{\psi}}
\qquad
\dfrac{\overline{\varphi}, \varphi \vdash \overline{\varphi}' \qquad \overline{\psi}, \psi \vdash \overline{\psi}'}{\overline{\varphi}, \overline{\psi}, \varphi \parr \psi \vdash \overline{\varphi}', \overline{\psi}'}
\qquad
\dfrac{\overline{\varphi} \vdash \varphi_0, \varphi_1, \overline{\varphi}'}{\overline{\varphi} \vdash \varphi_0 \parr \varphi_1, \overline{\varphi}'}
\qquad
\dfrac{\overline{\varphi} \vdash \varphi, \overline{\varphi}' \qquad \overline{\psi}, \psi \vdash \overline{\psi}'}{\overline{\varphi}, \overline{\psi}, \varphi \multimap \psi \vdash \overline{\varphi}', \overline{\psi}'}
$$

$$
\dfrac{\overline{\varphi}, \varphi \vdash \overline{\varphi}'}{\overline{\varphi}, (\exists z^\tau)\varphi \vdash \overline{\varphi}'}
\qquad
\dfrac{\overline{\varphi} \vdash \varphi[\mathtt{t}^\tau/x^\tau], \overline{\varphi}'}{\overline{\varphi} \vdash (\exists x^\tau)\varphi, \overline{\varphi}'}
\qquad
\dfrac{\overline{\varphi}, \varphi[\mathtt{t}^\tau/x^\tau] \vdash \overline{\varphi}'}{\overline{\varphi}, (\forall x^\tau)\varphi \vdash \overline{\varphi}'}
\qquad
\dfrac{\overline{\varphi} \vdash \varphi, ?\overline{\psi}}{\overline{\varphi} \vdash (\forall z^\tau)\varphi, ?\overline{\psi}}
$$

$$
\dfrac{\overline{\psi} \vdash \overline{\psi}'}{\overline{\psi}, !\varphi \vdash \overline{\psi}'}
\qquad
\dfrac{\overline{\psi}, !\varphi, !\varphi \vdash \overline{\psi}'}{\overline{\psi}, !\varphi \vdash \overline{\psi}'}
\qquad
\dfrac{\overline{\varphi}, \varphi \vdash \overline{\varphi}'}{\overline{\varphi}, !\varphi \vdash \overline{\varphi}'}
\qquad
\dfrac{!\overline{\varphi} \vdash \varphi, ?\overline{\psi}}{!\overline{\varphi} \vdash !\varphi, ?\overline{\psi}}
\qquad
\dfrac{\overline{\varphi}, !\varphi \vdash \psi, ?\overline{\psi}}{\overline{\varphi} \vdash !\varphi \multimap \psi, ?\overline{\psi}}
$$

$$
\dfrac{\overline{\psi} \vdash \overline{\psi}'}{\overline{\psi} \vdash ?\varphi, \overline{\psi}'}
\qquad
\dfrac{\overline{\psi} \vdash ?\varphi, ?\varphi, \overline{\psi}'}{\overline{\psi} \vdash ?\varphi, \overline{\psi}'}
\qquad
\dfrac{\overline{\varphi} \vdash \varphi, \overline{\psi}}{\overline{\varphi} \vdash ?\varphi, \overline{\psi}}
\qquad
\dfrac{!\overline{\varphi}, \varphi \vdash ?\overline{\psi}}{!\overline{\varphi}, ?\varphi \vdash ?\overline{\psi}}
$$

Figure 8.6.: Deduction Rules for $\mathsf{LMSO}(\mathbf{M})$ (where z^τ is fresh).

logic to be intuitionistic, one has to restrict the right $\multimap$ and $\forall$ rules to sequents with a single conclusion, while it is known (see e.g. [HdP93, Bie96, BH18]) that this restriction is problematic w.r.t. cut-elimination. On the other and, we can nevertheless adopt such a system since we dot not formally rely on cut-elimination for witness extraction as our realizability model already carries all the computational information we need.

Deduction for $\mathsf{LMSO}(\mathbf{M})$ consists of the rules of Fig. 8.6, together with the *Equality Axioms* of $\mathsf{LMSO}(\mathbf{T})$ (Def. 8.1.2) restricted to the language of $\mathsf{LMSO}(\mathbf{M})$. Note that our right rules for $\multimap$ and $\forall$ are quite specific:

$$
\dfrac{\overline{\varphi}, \varphi \vdash \psi}{\overline{\varphi} \vdash \varphi \multimap \psi}
\qquad
\dfrac{\overline{\varphi}, !\varphi \vdash \psi, ?\overline{\psi}}{\overline{\varphi} \vdash !\varphi \multimap \psi, ?\overline{\psi}}
\qquad
\dfrac{\overline{\varphi} \vdash \varphi, ?\overline{\psi}}{\overline{\varphi} \vdash (\forall z^\tau)\varphi, ?\overline{\psi}}
$$

Proposition 8.1.9 easily extends, assuming a usual two-sided classical sequent calculus for $\mathsf{MSO}(\mathbf{M})$.

Proposition 8.2.11 (Soundness of $(-)^T$ and $(-)^Q$)**.** *If*

$$
\psi_1, \ldots, \psi_m \vdash_{\mathsf{MSO}(\mathbf{M})} \varphi_1, \ldots, \varphi_n
$$

then

$$
!\psi_1^T, \ldots, !\psi_m^T \vdash_{\mathsf{LMSO}(\mathbf{T})} \varphi_1^T, \ldots, \varphi_n^T \qquad \text{and} \qquad \psi_1^Q, \ldots, \psi_m^Q \vdash_{\mathsf{LMSO}(\mathbf{T})} ?\varphi_1^Q, \ldots, ?\varphi_n^Q
$$

Adequacy of realizability (Prop. 8.1.4) is easily adapted from $\mathsf{LMSO}(\mathbf{T})$ to $\mathsf{LMSO}(\mathbf{M})$.

Proposition 8.2.12 (Adequacy)**.** *Consider an* $\mathsf{LMSO}(\mathbf{M})$ *sequent*

$$
\psi_1, \ldots, \psi_k \ \vdash \ \varphi_1, \ldots, \varphi_m \tag{8.11}
$$

and a list of variables $x_1^{\sigma_1}, \dots, x_n^{\sigma_n}$ *containing all free the variables of* $\psi_1, \dots, \psi_k$ *and* $\varphi_1, \dots, \varphi_m$. *Each proof of (8.11) induces a finite-state strategy in*

$$\mathcal{A}(\psi_1)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}) \otimes \dots \otimes \mathcal{A}(\psi_k)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}) \quad \multimap_{\mathsf{DialAut}([\![\sigma_1]\!] \times \dots \times [\![\sigma_n]\!])} \\ \mathcal{A}(\varphi_1)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}) \parr \dots \parr \mathcal{A}(\varphi_m)(x_1^{\sigma_1}, \dots, x_n^{\sigma_n})$$

(where DialAut *is understood with* $\mathfrak{D} = \mathbf{1}$*).*

We immediately deduce the expected result on witness extraction.

Corollary 8.2.13 (Extraction)**.** *Consider a closed* $\mathsf{LMSO}(\mathbf{M})$ *formula* $\varphi := (\forall x^\tau)(\exists y^\sigma)\delta(x, y)$ *with* δ *deterministic. From a proof of* φ *in* $\mathsf{LMSO}(\mathbf{M})$*, one can extract a term* $\mathtt{t}(x)$ *such that* $\models (\forall x^\tau)\lfloor \delta(x, \mathtt{t}(x)) \rfloor$.

Following Prop. 8.2.7, Cor. 8.2.13 applies to $\delta := {!}\psi^T(x^\tau, y^\sigma)$ and to $\delta := {?}\psi^Q(x^\tau, y^\sigma)$ for ψ an $\mathsf{MSO}(\mathbf{M})$ formula. In such cases, the conclusion of Cor. 8.2.13 can be replaced with $\models (\forall x^\tau)\psi(x, \mathtt{t}(x))$.

An important feature of the restricted exponentials of $\mathsf{LMSO}(\mathbf{M})$ is that realizer extraction in Prop. 8.2.12 (and thus also in Cor. 8.2.13) do not appeal to McNaughton's Determinization Theorem.

8.2.6. A Complete Axiomatization of the Realizability Model. While some choices made in $\mathsf{LMSO}(\mathbf{M})$ may seem odd (in part. the absence of unpolarized exponentials), it turns out that we can completely axiomatize its realizability model, with an axiomatization similar to that of MSO^ω (§3.1.4). This result was published in [PR19].

We start from a complete axiomatization of $\mathsf{MSO}(\mathbf{M})$, based on the complete axiomatization MSO^ω of §3.1.4. As noted in §3.1.1, we see our axiomatizations of MSO as subsystems of second-order Peano arithmetic, and as such, we shall assume a sort of *individuals* intended to range over ω-words positions $n \in \mathbb{N}$. As mentioned in Not. 8.2.1, the option we adopted in [PR19] (and also [PR18b]) is to extend the language of $\mathsf{MSO}(\mathbf{M})$ with suitable atomic predicates and relations. In particular, we define individuals by relativization to an atomic predicate $\mathsf{N}(x^o)$. We also extend $\mathsf{MSO}(\mathbf{M})$ with the atomic predicates of MSO^ω (§3.1.4), in which the individuals of MSO^ω are replaced by $\mathsf{MSO}(\mathbf{M})$ variables of sort o relativized to N, and the monadic predicate variables of MSO^ω are replaced by $\mathsf{MSO}(\mathbf{M})$ variables of sort o. We shall call $\mathsf{MSO}(\mathbf{M})^\omega$ the resulting system. $\mathsf{MSO}(\mathbf{M})^\omega$ has in addition other atomic predicates, mainly devised to have technically smoother translations between the different systems, and is unfortunately not very elegant. We shall thus not repeat it here and refer to [PR19, §4.1] for its full definition (see also [PR17, PR18b] for systems in the same spirit).

We are now going to sketch the axioms of $\mathsf{MSO}(\mathbf{M})^\omega$. The main difference with MSO^ω is that we have to handle the term language for f.s. causal functions. To this end, we shall give a defining axiom for each f.s. causal function. First, thanks to the Cartesian structure on the term language (8.3), each term $\mathtt{t}^\tau$ can be seen as a tuple of terms of sort o with free variables also of sort o. Then, we use the fact that for each term $\mathtt{t}^o$ with free variables $x_1^o, \dots, x_k^o$, there is an MSO^ω formula $\delta_{\mathtt{t}}(x, X_1, \dots, X_k)$ such that

$$\mathtt{t}^o(B_1, \dots, B_k)(n) = 1 \quad \Longleftrightarrow \quad \mathfrak{N} \models \delta_{\mathtt{t}}(n, B_1, \dots, B_k) \qquad (\text{for } B_1, \dots, B_k \in \mathbf{2}^\omega \text{ and } n \in \mathbb{N})$$

Remark 8.2.14. *This is essentially what gives the interpretation of* $\mathsf{MSO}(\mathbf{M})$ *in* MSO^ω *(over their standard models) mentioned in §8.2.1. Again using the Cartesian structure on the term language (8.3), we reduce each equality* $\mathtt{t}^\tau \doteq \mathtt{u}^\tau$ *to a conjunction of equalities of the form*

$$\mathtt{t}_1^o \doteq \mathtt{u}_1^o \ \wedge \ \dots \ \wedge \ \mathtt{t}_n^o \doteq \mathtt{u}_n^o$$

and then turn each of these equalities to a logical equivalence between the corresponding $\delta_{\mathtt{t}_i}$ *and* $\delta_{\mathtt{u}_i}$*. We refer to the appendices of [PR18b, PR19] for details.*

The axioms of $\mathsf{MSO}(\mathbf{M})^\omega$ extend those of $\mathsf{MSO}(\mathbf{M})$ with a bunch of axioms similar to the (adaptation to MSO^ω of the) axioms of Fig. 3.1 (§3.1.2), together with the following:

- *Induction.*

$$\frac{\overline{\varphi},\, \mathsf{Z}(z) \;\vdash\; \varphi[z/x],\, \overline{\varphi}' \qquad \overline{\varphi},\, \mathsf{S}(y,z),\, \varphi[y/x] \;\vdash\; \varphi[z/x],\, \overline{\varphi}'}{\overline{\varphi} \;\vdash\; (\forall x^\iota)\varphi,\, \overline{\varphi}'} \qquad (y, z \text{ fresh})$$

- *Comprehension.*

$$\frac{}{\vdash\; (\exists y^o)(\forall x^\iota)\,(x \mathrel{\dot\in} y \;\leftrightarrow\; \varphi)} \qquad (y \text{ fresh})$$

- *Definition of Mealy Machines.*

$$\frac{}{\vdash\; (\forall \overline{y}^o)(\forall x^\iota)\,(x \mathrel{\dot\in} \mathtt{t}^o(\overline{y}) \;\leftrightarrow\; \delta_{\mathtt{t}}(x, \overline{y}))}$$

The formulation of induction in $\mathsf{MSO}(\mathbf{M})^\omega$ is intended to allow for a smoother translation to the corresponding extension $\mathsf{LMSO}(\mathbf{M})^\omega$ of $\mathsf{LMSO}(\mathbf{M})$. We of course have the expected extension of Siefkes' Theorem [Sie70] (Thm. 3.1.4, the completeness of MSO^ω).

Theorem 8.2.15. $\mathsf{MSO}(\mathbf{M})^\omega$ *is complete.*

In particular, we can now strengthen the statement of Büchi-Landweber Theorem given in Thm. 8.2.3 as providing provability in $\mathsf{MSO}(\mathbf{M})^\omega$.

The logic $\mathsf{LMSO}(\mathbf{M})^\omega$ is intended to be to $\mathsf{LMSO}(\mathbf{M})$ what $\mathsf{MSO}(\mathbf{M})^\omega$ is to $\mathsf{MSO}(\mathbf{M})$. This implies that there should be translations from $\mathsf{MSO}(\mathbf{M})^\omega$ to $\mathsf{LMSO}(\mathbf{M})^\omega$. But while the $(-)^T$ and $(-)^Q$ translations were perfectly fine with $\mathsf{MSO}(\mathbf{M})$ and $\mathsf{LMSO}(\mathbf{M})$, this is not longer the case for $\mathsf{MSO}(\mathbf{M})^\omega$. The reason is that in the Comprehension scheme of $\mathsf{LMSO}(\mathbf{M})^\omega$, we will have (for the adequacy of realizability) to assume that the comprehension formula is deterministic. As a consequence, we look for variants of $(-)^T$ and $(-)^Q$ targeting deterministic formulae. In addition, for the complete axiomatization of the realizability model of $\mathsf{LMSO}(\mathbf{M})$, we shall actually require a translation $(-)^L$ such that the following equivalences

$$?\varphi^+ \multimapboth \lfloor \varphi^+ \rfloor^L \qquad \delta^\pm \multimapboth \lfloor \delta^\pm \rfloor^L \qquad !\psi^- \multimapboth \lfloor \psi^- \rfloor^L \tag{8.12}$$

are provable, possibly using extra axioms that we require to be realizable. It is easy to see that (8.12) implies the following scheme (DEXP):

$$\delta \;\multimap\; !\delta \qquad \text{and} \qquad ?\delta \;\multimap\; \delta \qquad (\delta \text{ deterministic})$$

and that $(-)^L$ should yield deterministic formulae. While $(-)^T$ and $(-)^Q$ can be adapted accordingly, (8.12) induces axioms which make the resulting translations equivalent to the deterministic $(-)^L$-translation of [PR18b]:

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
(\mathtt{t}^\tau \doteq \mathtt{u}^\tau)^L & := & \mathtt{t}^\tau \doteq \mathtt{u}^\tau & (\psi \to \varphi)^L & := & \psi^L \multimap \varphi^L & (\forall x^\sigma.\varphi)^L & := & !(\forall x^\sigma)\varphi^L \\
\top^L & := & \mathbf{I} & (\varphi \wedge \psi)^L & := & \varphi^L \otimes \psi^L & (\exists x^\sigma.\varphi)^L & := & ?(\exists x^\sigma)\varphi^L \\
\bot^L & := & \bot & (\varphi \vee \psi)^L & := & \varphi^L \parr \psi^L & & &
\end{array}$$

Proposition 8.2.16 ([PR19]). *The scheme (8.12) is equivalent in* $\mathsf{LMSO}(\mathbf{M})$ *to* $(\mathsf{DEXP}) + (\mathsf{PEXP})$*, where* (PEXP) *are the following* polarized exponential *axioms, with polarities as shown:*

$$
\begin{array}{rcl@{\qquad}rcl}
?(\varphi^+) & \multimap & ?!(\varphi^+) & !?(\psi^-) & \multimap & !(\psi^-) \\
!(\varphi^-) \multimap ?(\psi^+) & \multimap & ?(\varphi^- \multimap \psi^+) & ?(\varphi^+) \multimap !(\psi^-) & \multimap & !(\varphi^+ \multimap \psi^-) \\
?(\varphi^+) \otimes ?(\psi^+) & \multimap & ?(\varphi^+ \otimes \psi^+) & !(\varphi^- \otimes \psi^-) & \multimap & !(\varphi^-) \otimes !(\psi^-) \\
?(\varphi^+) \parr ?(\psi^+) & \multimap & ?(\varphi^+ \parr \psi^+) & !(\varphi^- \parr \psi^-) & \multimap & !(\varphi^-) \parr !(\psi^-)
\end{array}
$$

The soundness of $(-)^L$ from $\mathsf{MSO}(\mathbf{M})$ to $\mathsf{LMSO}(\mathbf{M}) + (\mathsf{DEXP})$ is easy to prove. Also, the axioms (DEXP) and (PEXP) are trivially realized, so that Adequacy (Prop. 8.2.12) and Extraction (Cor. 8.2.13) extend to $\mathsf{LMSO}(\mathbf{M}) + (\mathsf{DEXP}) + (\mathsf{PEXP})$. Note that the deterministic formula δ assumed for Extraction can in particular be of the form ψ^L for ψ an $\mathsf{MSO}(\mathbf{M})$ formula.

We can now turn to $\mathsf{LMSO}(\mathbf{M})^\omega$. We refer to [PR19, §4.2] for details. The language of $\mathsf{LMSO}(\mathbf{M})^\omega$ is that of $\mathsf{LMSO}(\mathbf{M})$ with the same atomic predicates and relations as $\mathsf{MSO}(\mathbf{M})^\omega$. The axioms of $\mathsf{LMSO}(\mathbf{M})^\omega$ have the same structure as those of $\mathsf{MSO}(\mathbf{M})^\omega$. We just indicate the counterparts of *Induction*, *Comprehension* and *Definition of Mealy Machines*:

- *Induction.*

$$
\frac{!\overline{\varphi},\ \mathsf{Z}(z)\ \vdash\ \varphi^-[z/x],\ ?\overline{\varphi}' \qquad !\overline{\varphi},\ \mathsf{S}(y,z),\ !\varphi^-[y/x]\ \vdash\ \varphi^-[z/x],\ ?\overline{\varphi}'}{!\overline{\varphi}\ \vdash\ (\forall x^\iota)\varphi^-,\ !\overline{\varphi}'}
$$

(y, z fresh and φ^- negative)

- *Comprehension.*

$$
\frac{}{\vdash\ ?(\exists y^o)!(\forall x^\iota)\,(x \mathbin{\dot\in} y\ \multimapboth\ \delta^\pm)}
$$

(y fresh and $\delta^\pm$ deterministic)

- *Definition of Mealy Machines.*

$$
\frac{}{\vdash\ (\forall \overline{y}^o)(\forall x^\iota)\left(x \mathbin{\dot\in} \mathtt{t}^o(\overline{y})\ \multimapboth\ \delta^L_{\mathtt{t}}(x,\overline{y})\right)}
$$

The soundness of $(-)^L$, as well as Adequacy and Extraction of course extend to $\mathsf{LMSO}(\mathbf{M})^\omega + (\mathsf{DEXP}) + (\mathsf{PEXP})$, interpreting the additional atomic predicates of $\mathsf{LMSO}(\mathbf{M})^\omega$ by suitable deterministic automata. But now, thanks to the completeness of $\mathsf{MSO}(\mathbf{M})^\omega$ (Thm. 8.2.15), more is true: Witness extraction from proofs in $\mathsf{LMSO}(\mathbf{M})^\omega + (\mathsf{DEXP})$ is complete w.r.t. Church's Synthesis.

Proposition 8.2.17. *If* φ *is provable in* $\mathsf{MSO}(\mathbf{M})^\omega$ *then* $\mathsf{LMSO}(\mathbf{M})^\omega + (\mathsf{DEXP})$ *proves* φ^L*.*

In particular, for a closed *deterministic* $\mathsf{LMSO}(\mathbf{M})^\omega$ formula δ, if $\lfloor\delta\rfloor$ is true then δ is provable in $\mathsf{LMSO}(\mathbf{M})^\omega + (\mathsf{DEXP})$.

Proposition 8.2.18 (Adequacy)**.** *Consider an* $\mathsf{LMSO}(\mathbf{M})^\omega$ *sequent*

$$
\psi_1, \ldots, \psi_k\ \vdash\ \varphi_1, \ldots, \varphi_m \tag{8.13}
$$

and a list of variables $x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}$ *containing all free the variables of* $\psi_1, \ldots, \psi_k$ *and* $\varphi_1, \ldots, \varphi_m$*. Each proof of (8.13) in* $\mathsf{LMSO}(\mathbf{M})^\omega + (\mathsf{DEXP}) + (\mathsf{PEXP})$ *induces a finite-state strategy in*

$$
\mathcal{A}(\psi_1)(x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}) \otimes \ldots \otimes \mathcal{A}(\psi_k)(x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}) \ \multimap_{\mathsf{DialAut}(\llbracket\sigma_1\rrbracket \times \cdots \times \llbracket\sigma_n\rrbracket)} \\ \mathcal{A}(\varphi_1)(x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}) \parr \cdots \parr \mathcal{A}(\varphi_m)(x_1^{\sigma_1}, \ldots, x_n^{\sigma_n})
$$

Corollary 8.2.19 (Extraction)**.** *Consider a closed* $\mathsf{LMSO}(\mathbf{M})^\omega$ *formula* $\varphi := (\forall x^\tau)(\exists y^\sigma)\delta(x, y)$ *with* δ *deterministic. From a proof of* φ *in* $\mathsf{LMSO}(\mathbf{M})^\omega + (\mathsf{DEXP}) + (\mathsf{PEXP})$*, one can extract a term* $\mathtt{t}(x)$ *such that* $\models (\forall x^\tau)\lfloor\delta(x, \mathtt{t}(x))\rfloor$*.*

Corollary 8.2.20 (Soundness and Completeness w.r.t. Church's Synthesis)**.** *Given a closed* $\mathsf{MSO}(\mathbf{M})^\omega$ *formula* $\varphi := (\forall x^\tau)(\exists y^\sigma)\psi(x, y)$*, the following are equivalent:*

- φ *is realizable in the sense of Church's Synthesis.*
- $(\forall x^\tau)(\exists y^\sigma)\psi^L(x, y)$ *is provable in* $\mathsf{LMSO}(\mathbf{M})^\omega + (\mathsf{DEXP})$*.*

Corollary 8.2.20 improves on Prop. 8.2.7 in that not only the *language* of $\mathsf{LMSO}(\mathbf{M})$ is well adapted to Church's Synthesis, but also its *axiomatization* provides a sound and complete method for Church's Synthesis. *In principle*, this approach is entirely syntactic and *Safraless* (as it avoids McNaughton Theorem).

We now turn to the axiomatization of the realizability model of $\mathsf{LMSO}(\mathbf{M})$. The main idea, developed in [PR19], is to use a variant of Gödel's functional "*Dialectica*" interpretation in order to internalize the (f.s. part) of the realizability model of $\mathsf{LMSO}(\mathbf{M})$. Actually, this internalized interpretation could be presented (as in [PR19]) as a purely syntactic way to obtain the soundness of witness extraction in $\mathsf{LMSO}(\mathbf{M})^\omega$ w.r.t. Church's Synthesis (Cor 8.2.19).

Gödel's Dialectica interpretation associates to a formula $\varphi(a)$ a formula $\varphi^D(a)$ of the form $(\exists u^\tau)(\forall x^\sigma)\varphi_D(u, x, a)$. In usual versions formulated in higher-types arithmetic (see e.g. [AF98, Koh08]), the formula φ_D is quantifier-free, so that φ^D is a prenex form of φ. This prenex form is constructive, and a constructive proof of φ can be turned to a proof of φ^D with an explicit (closed) witness for $\exists u$. We call such witnesses *Dialectica realizers* of φ. Even if Dialectica originally interprets intuitionistic arithmetic, it is structurally linear: in general, Dialectica realizers of contraction

$$\varphi(a) \quad\longrightarrow\quad \varphi(a) \land \varphi(a)$$

only exist when the term language can decide $\varphi_D(u, x, a)$, which is possible in arithmetic but not in all settings. Besides, linear versions of Dialectica were formulated at the very beginning of linear logic [dP87, dP89, dP91] (see also [Hyl02, Shi06]).

The essence of [PR19] is to use a variant of Dialectica as a syntactic formulation of the realizability model of $\mathsf{LMSO}(\mathbf{M})$. The formula φ_D essentially represents a deterministic automaton on ω-words and is in general not quantifier-free. Moreover, we extract f.s. causal functions, while the category $\mathbf{M}$ is not closed. As a result, a Dialectica realizer of φ with $\varphi^D(a)$ of the form $(\exists u^\tau)(\forall x^\sigma)\varphi_D(u, x, a)$ is an *open* (eager) term $\mathtt{u}(x)$ of arity $(\sigma; \tau)$ such that $\mathsf{LMSO}(\mathbf{M})^\omega$ proves $\varphi_D(\mathtt{u}(x), x)$.

Our Dialectica-like interpretation of the multiplicative (*i.e.* exponential free) fragment of $\mathsf{LMSO}(\mathbf{M})^\omega$ roughly follows the **DC** interpretation of [dP87, dP91]. In order to keep notations simple, we reason modulo the Cartesian structure on terms (8.3) and assume formulae to have at most one free variable (which is thus in general of product type). For atomic formulae we let $\varphi^D(a) := \varphi_D(a) := \varphi(a)$. The inductive cases are given in Fig. 8.7, where $\varphi^D(a) = (\exists u)(\forall x)\varphi_D(u, x, a)$ and $\psi^D(a) = (\exists v)(\forall y)\psi_D(v, y, a)$. Note that we leaved the types implicit in Fig. 8.7. They are however easy to reconstruct from the corresponding constructions on automata of §7.2. The soundness statement of $(-)^D$ is that from a proof of a *closed* φ, one can extract a Dialectica realizer of φ. In the case of an open formula, say $\varphi(a)$, we extract realizers of $(\forall a)\varphi(a)$. In general, given an $\mathsf{LMSO}(\mathbf{M})^\omega$ sequent

$$\psi_1, \ldots, \psi_k \;\vdash\; \varphi_1, \ldots, \varphi_m \tag{8.14}$$

and a list of variables $x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}$ containing all free the variables of $\psi_1, \ldots, \psi_k$ and $\varphi_1, \ldots, \varphi_m$, from a proof of (8.14) we extract a realizer of

$$(\forall x_1^{\sigma_1}) \cdots (\forall x_n^{\sigma_n}) \big(\psi_1 \otimes \cdots \otimes \psi_k \quad \multimap \quad \varphi_1 \parr \cdots \parr \varphi_m\big)$$

Having in mind our specific extraction of *open eager* terms, note that the interpretation of the linear arrow follows the representation of strategies as pairs of functions (Prop. 6.2.2 and §6.5.2). Indeed, consider closed exponential free formulae ψ and φ, with $\varphi^D = (\exists u^\tau)(\forall x^\sigma)\varphi_D(u,x)$ and $\psi^D = (\exists v^\kappa)(\forall y^\upsilon)\psi_D(v,y)$. Then, modulo (8.3), $(\psi \multimap \varphi)^D$ is of the form

$$(\exists f^{(\kappa)\tau})(\exists F^{(\sigma)\tau\upsilon})(\forall u^\tau)(\forall y^\upsilon)\big(\psi_D(u,(F)uy) \;\multimap\; \varphi_D((f)u,y)\big)$$

Hence, a Dialectica realizer of $\psi \multimap \varphi$ is a pair of eager terms $\mathtt{v}(u^\tau, y^\upsilon)$ of sort $(\kappa)\tau$ and $\mathtt{x}(u^\tau, y^\upsilon)$ of sort $(\kappa)\tau\upsilon$. But these terms induces eager f.s. functions

$$[\![\mathtt{v}]\!] \;:\; [\![\tau]\!] \times [\![\upsilon]\!] \;\longrightarrow_{\mathbf{EM}}\; [\![\kappa]\!]^{[\![\tau]\!]} \qquad \text{and} \qquad [\![\mathtt{x}]\!] \;:\; [\![\tau]\!] \times [\![\upsilon]\!] \;\longrightarrow_{\mathbf{EM}}\; [\![\kappa]\!]^{[\![\tau]\!] \times [\![\upsilon]\!]}$$

that is, a finite-state P strategy in the game

$$([\![\tau]\!], [\![\sigma]\!]) \quad \multimap_{\mathbf{DZ}} \quad ([\![\kappa]\!], [\![\upsilon]\!])$$

in the sense of §6.5.2. As a consequence, the adequacy of $(-)^D$ essentially amounts to a formalization within $\mathsf{LMSO}(\mathbf{M})$ of the adequacy of the automata-based realizability model. The only non-trivial case is that of the cut rule, which amounts to a formal treatment of composition of finite state strategies (see §6.3.3). We refer to [PR19, App. B] for details.

More generally, for an exponential free closed formula φ with $\varphi^D = (\exists u^\tau)(\forall x^\sigma)\varphi_D(u,x)$, $[\![\tau]\!]$ and $[\![\sigma]\!]$ are respectively the sets of P and O moves of the uniform automaton $\mathcal{A}(\varphi)$. In particular, the polarities of $\mathsf{LMSO}(\mathbf{M})^\omega$ can be read off from the interpretation $(-)^D$, as for a formula φ as above:

- φ is positive iff $[\![\sigma]\!] \simeq \mathbf{1}$,
- φ is negative iff $[\![\tau]\!] \simeq \mathbf{1}$, and
- φ is deterministic iff $[\![\tau]\!] \simeq [\![\sigma]\!] \simeq \mathbf{1}$.

We now turn to the exponentials of $\mathsf{LMSO}(\mathbf{M})^\omega$. Actually, the reason for the restriction to polarized exponentials comes from the fact that we do not have a good syntactic interpretation of the general exponentials in our Dialectica-like interpretation. But let us begin by remarking that for closed formulae, the Büchi-Landweber Theorem implies, via Completeness of $\mathsf{MSO}(\mathbf{M})^\omega$ and Prop. 8.2.17 that contraction for closed formulae can be Dialectica realized.

Example 8.2.21. *Realizers of $\varphi \multimap \varphi \otimes \varphi$ for a closed φ are given by eager terms $\mathtt{U}_1(u,x_1,x_2)$, $\mathtt{U}_2(u,x_1,x_2)$, $\mathtt{X}(u,x_1,x_2)$ representing P-strategies in the game $\mathcal{G}(\Phi)(\langle U_1, U_2, X\rangle, \langle u, x_1, x_2\rangle)$, where Φ is*

$$\lfloor \varphi_D(u,(X)ux_1x_2) \rfloor \quad \longrightarrow \quad \lfloor \varphi_D((U_1)u,x_1) \rfloor \;\;\wedge\;\; \lfloor \varphi_D((U_2)u,x_2) \rfloor$$

By the Büchi-Landweber Theorem 8.2.3, either there is an eager term $\mathtt{U}(x)$ such that $\lfloor \varphi_D(\mathtt{U}(x),x) \rfloor$ holds, so that

$$\lfloor \varphi_D(u,x_1) \rfloor \quad \longrightarrow \quad \lfloor \varphi_D(\mathrm{ev}(\mathtt{U})(x_1),x_1) \rfloor \;\;\wedge\;\; \lfloor \varphi_D(\mathrm{ev}(\mathtt{U})(x_2),x_2) \rfloor$$

$$(\varphi \otimes \psi)^D(a) \;:=\; \exists\langle u,v\rangle\forall\langle x,y\rangle.\ (\varphi \otimes \psi)_D(\langle u,v\rangle,\langle x,y\rangle,a) \;:=\; \exists\langle u,v\rangle\forall\langle x,y\rangle.\ \varphi_D(u,x,a) \otimes \psi_D(v,y,a)$$

$$(\varphi \parr \psi)^D(a) \;:=\; \exists\langle u,v\rangle\forall\langle x,y\rangle.\ (\varphi \parr \psi)_D(\langle u,v\rangle,\langle x,y\rangle,a) \;:=\; \exists\langle u,v\rangle\forall\langle x,y\rangle.\ \varphi_D(u,x,a) \parr \psi_D(v,y,a)$$

$$(\varphi \multimap \psi)^D(a) \;:=\; \exists\langle f,F\rangle\forall\langle u,y\rangle.\ (\varphi \multimap \psi)_D(\langle f,F\rangle,\langle u,y\rangle,a) \;:=\; \exists\langle f,F\rangle\forall\langle u,y\rangle.\ \varphi_D(u,(F)uy,a) \multimap \psi_D((f)u,y,a)$$

$$(\exists w.\varphi)^D(a) \;:=\; \exists\langle u,w\rangle\forall x.\ (\exists w.\varphi)_D(\langle u,w\rangle,x,a) \;:=\; \exists\langle u,w\rangle\forall x.\ \varphi_D(u,x,\langle a,w\rangle)$$

$$(\forall w.\varphi)^D(a) \;:=\; \exists f\,\forall\langle x,w\rangle.\ (\forall w.\varphi)_D(f,\langle x,w\rangle,a) \;:=\; \exists f\,\forall\langle x,w\rangle.\ \varphi_D((f)w,x,\langle a,w\rangle)$$

Figure 8.7.: A Dialectica-like Interpretation of the Exponential Free Fragment of $\mathsf{LMSO}(\mathbf{M})^\omega$.

or there is an eager term $\mathtt{X}(u)$ *such that* $\neg\lfloor\varphi_D(u,\mathrm{ev}(\mathtt{X})(u))\rfloor$ *holds, so that*

$$\lfloor\varphi_D(u,\mathrm{ev}(\mathtt{X})(u))\rfloor \quad\longrightarrow\quad \lfloor\varphi_D(u,x_1)\rfloor \;\wedge\; \lfloor\varphi_D(u,x_2)\rfloor$$

Since $\varphi_D(u,\mathrm{ev}(\mathtt{X})(u)) \multimap \varphi_D(u,x_1) \otimes \varphi_D(u,x_2)$ *is deterministic, it follows that* $\varphi \multimap \varphi\otimes\varphi$ *is Dialectica realized provably in* $\mathsf{LMSO}(\mathbf{M})^\omega$.

However, contraction is in general not realized for *open* formulae.

Example 8.2.22. *Consider the open formula* $\varphi(a^o) := (\forall x^o)(\mathtt{t}(x,a) \doteq 0^\omega)$ *where* $[\![\mathtt{t}]\!](B,C) = 0^{n+1}1^\omega$ *for the first* $n\in\mathbb{N}$ *with* $C(n+1) = B(0)$ *if such* n *exists, and such that* $[\![\mathtt{t}]\!](B,C) = 0^\omega$ *otherwise. The game induced by* $((\forall a)(\varphi \multimap \varphi\otimes\varphi))_D$ *is* $\mathcal{G}(\Phi)(X,\langle x_1,x_1,a\rangle)$*, where* Φ *is*

$$\mathtt{t}((X)x_1x_2a,a) \doteq 0^\omega \quad\longrightarrow\quad \mathtt{t}(x_1,a)\doteq 0^\omega \;\wedge\; \mathtt{t}(x_2,a)\doteq 0^\omega$$

In this game, P *begins by playing a function* $\mathbf{2}^3 \to \mathbf{2}$*,* O *replies in* $\mathbf{2}^3$*, and then* P *and* O *keep on alternatively playing moves of the expected type. A finite-state winning strategy for* O *is easy to find. Let* P *begin with the function* $\mathtt{X}$*. Fix some* $\mathtt{a}\in\mathbf{2}$ *and let* $i := \mathtt{X}(0,1,\mathtt{a})$*.* O *replies* $(0,1,\mathtt{a})$ *to* $\mathtt{X}$*. The further moves of* P *are irrelevant, and* O *keeps on playing* $(-,-,1-i)$ *(the values of* x_1 *and* x_2 *are irrelevant after the first round). This strategy ensures*

$$\mathtt{t}((X)x_1x_2a,a) \doteq 0^\omega \quad\wedge\quad \neg(\mathtt{t}(x_1,a)\doteq 0^\omega \;\wedge\; \mathtt{t}(x_2,a)\doteq 0^\omega)$$

Consider now exponential formulae $!\varphi$ and $?\psi$ of $\mathsf{LMSO}(\mathbf{M})^\omega$. In the case of φ positive and ψ negative, we can let

$$\begin{array}{rclcl} (!(\varphi^+))^D(a) & := & (\exists u)(!(\varphi^+))_D(u,-,a) & := & (\exists u)!\varphi_D(u,-,a) \\ (?(\psi^-))^D(a) & := & (\forall y)(?(\psi^-))_D(-,y,a) & := & (\forall x)?\psi_D(-,y,a) \end{array} \tag{8.15}$$

Note that this respects the correspondence between the types in Dialectica and moves in automata, and so in particular the polarities of formulae.

The interesting case for exponentials is that of $?\varphi$ and $!\psi$ for φ positive and ψ negative. Assume $\varphi^D(a) = (\exists u)\varphi_D(u,-,a)$ and $\psi^D(a) = (\forall y)\psi_D(-,y,a)$. Recall that φ^D (resp. ψ^D) is thought about as representing the non-deterministic (resp. universal) automaton $\mathcal{A}(\varphi)$ (resp.

$\mathcal{A}(\psi)$). Then the Dialectica interpretation of the operation $?\varphi$ (resp. $!\psi$) should correspond to McNaughton's Determinization (resp. to co-determinization). We let

$$\begin{array}{rcccl} (?(\varphi^+))^D(a) & := & (?(\varphi^+))_D(-,-,a) & := & ?(\exists u)\varphi_D(u,-,a) \\ (!(\psi^-))^D(a) & := & (!(\psi^-))_D(-,-,a) & := & !(\forall y)\psi_D(-,y,a) \end{array} \tag{8.16}$$

The interpretation of exponentials given in (8.15 and (8.16) completes the definition of $(-)^D$.

As usual, Dialectica is such that φ^D is equivalent to φ via possibly non-intuitionistic but constructive principles. The tricky connectives are implication and universal quantification. Similarly as in the intuitionistic case (see e.g. [Koh08, AF98, Tro73]), $(\psi \multimap \varphi)^D$ is a prenex form of $\psi^D \multimap \varphi^D$ obtained using (LAC) together with linear variants of the *Markov* and *Independence of premises* principles. In our case, the equivalence $\varphi \multimapboth \varphi^D$ also requires the following additional axioms (LSIP), with polarities as displayed:

$$\begin{array}{rcl} (\forall a)(\varphi^-(a) \otimes \psi^-) & \multimap & (\forall a)\varphi^-(a) \otimes \psi^- \\ (\forall a)(\varphi^-(a) \parr \psi^-) & \multimap & (\forall a)\varphi^-(a) \parr \psi^- \\ (\exists a)\varphi^-(a) \parr \psi & \multimap & (\exists a)(\varphi^-(a) \parr \psi) \\ (\psi^- \multimap (\exists a)\varphi^-(a)) & \multimap & (\exists a)(\psi^- \multimap \varphi^-(a)) \\ ((\forall a)\varphi^{\pm}(a) \multimap \psi^{\pm}) & \multimap & (\exists a)(\varphi^{\pm}(a) \multimap \psi^{\pm}) \end{array} \tag{LSIP}$$

Proposition 8.2.23 (Characterization). *For $\varphi(a)$ an* $\mathsf{LMSO}(\mathbf{M})^\omega$ *formula,* $\mathsf{LMSO}(\mathbf{M})^\omega$+(LAC)+(LSIP) + (DEXP) *proves* $\varphi(a) \multimapboth \varphi^D(a)$

We write $\mathsf{LMSO}(\mathfrak{C})$ for the system $\mathsf{LMSO}(\mathbf{M})^\omega$ + (LAC) + (LSIP) + (DEXP) + (PEXP). It is clear the Adequacy and Extraction hold for this system, so that the theory of $\mathsf{LMSO}(\mathfrak{C})$ is realized and is thus coherent, in the sense that $\bot$ is not provable. What is remarkable is that $\mathsf{LMSO}(\mathfrak{C})$ is complete in the following sense:

Theorem 8.2.24 (Completeness of $\mathsf{LMSO}(\mathfrak{C})$). *For each closed* $\mathsf{LMSO}(\mathbf{M})^\omega$ *formula φ, either* $\vdash_{\mathsf{LMSO}(\mathfrak{C})} \varphi$ *or* $\vdash_{\mathsf{LMSO}(\mathfrak{C})} \varphi \multimap \bot$.

Hence $\mathsf{LMSO}(\mathfrak{C})$ completely axiomatizes the theory if its realizability model. Note that since φ and $!\varphi$ are equiprovable, we in particular have the alternative $\vdash_{\mathsf{LMSO}(\mathfrak{C})} \varphi$ or $\vdash_{\mathsf{LMSO}(\mathfrak{C})} !\varphi \multimap \bot$.

Theorem 8.2.24 follows from a couple of facts. First, elimination linear double negation (Ex. 8.2.9) lifts from the realizability model to $\mathsf{LMSO}(\mathfrak{C})$. Combined with (LAC), this in particular gives a form of classical linear choice:

$$\vdash_{\mathsf{LMSO}(\mathfrak{C})} \quad (\forall f)(\exists x)\varphi(x,(f)x) \quad \multimap \quad (\exists x)(\forall y)\varphi(x,y)$$

Second, thanks to the existence of fixpoints of eager machines (Prop. 6.5.5, §6.5.1), we then obtain the following quantifier inversion.

Lemma 8.2.25. $\vdash_{\mathsf{LMSO}(\mathfrak{C})} \quad (\forall x^\sigma)\varphi(\mathtt{t}^\tau(x),x) \quad \multimap \quad (\exists u^\tau)(\forall x^\sigma)\varphi(u,x)$, *where* $\mathtt{t}(x)$ *is eager.*

The assumption that $\mathtt{t}(x)$ is eager in Lem. 8.2.25 is crucial. Completeness of $\mathsf{LMSO}(\mathfrak{C})$ then follows via $(-)^D$, Proposition 8.2.16, completeness of $\mathsf{MSO}(\mathbf{M})$ and the Büchi-Landweber Theorem. We give the full argument as it is a simple combination of the preceding results.

Proof of Thm. 8.2.24. Let φ be a closed $\mathsf{LMSO}(\mathbf{M})^\omega$ formula and write $\varphi^D = (\exists u^\tau)(\forall x^\sigma)\varphi_D(u,x)$. We apply Büchi-Landweber Theorem in the form of Thm. 8.2.3 to the $\mathsf{MSO}(\mathbf{M})^\omega$ formula $\lfloor \varphi_D(u^\tau, x^\sigma) \rfloor$. There are two cases.

- Either there exists an eager term $\mathtt{u}(x)$ of sort (σ, τ) such that $\models (\forall x^\sigma)\lfloor \varphi_D(\mathtt{t}(x), x)\rfloor$. We then proceed as follows.

$$
\begin{array}{lll}
\mathsf{MSO}(\mathbf{M})^\omega & \vdash \lfloor \varphi_D(\mathtt{u}(x), x)\rfloor & \text{(Completeness of } \mathsf{MSO}(\mathbf{M})^\omega) \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash \lfloor \varphi_D(\mathtt{u}(x), x)\rfloor^L & \text{(Proposition 8.2.17)} \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash \varphi_D(\mathtt{u}(x), x) & \text{(Proposition 8.2.16)} \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash (\forall x^\sigma)\varphi_D(\mathtt{u}(x), x) & (\forall\text{-right}) \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash (\exists u^\tau)(\forall x^\sigma)\varphi_D(u, x) & \text{(Lemma 8.2.25, since } \mathtt{u}(x) \text{ is eager)} \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash \varphi^D & \text{(Definition of } \varphi^D) \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash \varphi & \text{(Characterization)}
\end{array}
$$

- Otherwise, there exists a term $\mathtt{x}(u)$ of sort $(\tau; \sigma)$ such that $\models (\forall u^\tau)\neg\lfloor \varphi_D(\mathtt{x}(u), u)\rfloor$. Note that

$$
\neg\lfloor \varphi_D(\mathtt{x}(u), u)\rfloor \quad = \quad \lfloor \varphi_D(\mathtt{x}(u), u) \multimap \bot\rfloor
$$

  We then conclude as follows.

$$
\begin{array}{lll}
\mathsf{MSO}(\mathbf{M})^\omega & \vdash \lfloor \varphi_D(u, \mathtt{x}(u)) \multimap \bot\rfloor & \text{(Completeness of } \mathsf{MSO}(\mathbf{M})^\omega) \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash \lfloor \varphi_D(u, \mathtt{x}(u)) \multimap \bot\rfloor^L & \text{(Proposition 8.2.17)} \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash \varphi_D(u, \mathtt{x}(u)) \multimap \bot & \text{(Proposition 8.2.16)} \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash (\exists x^\sigma)(\varphi_D(u, x) \multimap \bot) & (\exists\text{-right}) \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash (\forall u^\tau)(\exists x^\sigma)(\varphi_D(u, x) \multimap \bot) & (\forall\text{-right}) \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash (\exists g^{(\sigma)\tau})(\forall u^\tau)(\varphi_D(u, (g)u)) \multimap \bot) & (\mathsf{LAC}) \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash (\varphi \multimap \bot)^D & \text{(Definition of } (\varphi \multimap \bot)^D) \\
\mathsf{LMSO}(\mathfrak{C}) & \vdash \varphi \multimap \bot & \text{(Characterization)}
\end{array}
$$

□

8.3. Back to Infinite Trees

We now come back to the case of infinite trees. In §8.1 we have discussed the systems $\mathsf{MSO}(\mathbf{T})$ and $\mathsf{LMSO}(\mathbf{T})$ from the point of view of soundness and adequacy. In this Section, which is purely exploratory, we briefly discuss what could lead to complete systems, in the same spirit as the systems $\mathsf{MSO}(\mathbf{M})^\omega$ and $\mathsf{LMSO}(\mathbf{M})^\omega$ of §8.2.6 for ω-words.

We assume an extension $\mathsf{MSO}(\mathbf{T})^+$ of the language of $\mathsf{MSO}(\mathbf{T})$ which syntactically contains the language of $\mathsf{MSO}(\mathfrak{D})$ (§3.1), similarly as we did for $\mathsf{MSO}(\mathbf{M})^\omega$ v.s. $\mathsf{MSO}(\mathbf{M})$. We assume the language of $\mathsf{LMSO}(\mathbf{T})$ to be similarly extended to $\mathsf{LMSO}(\mathbf{T})^+$. The atomic predicates of $\mathsf{MSO}(\mathbf{T})^+$ and $\mathsf{LMSO}(\mathbf{T})^+$ can be interpreted as deterministic automata, so that the *Tree Axioms* of $\mathsf{MSO}(\mathfrak{D})$ (Fig. 3.1, §3.1.2) are trivially realized. Similarly as with $\mathsf{MSO}(\mathbf{M})^\omega$ and $\mathsf{LMSO}(\mathbf{M})^\omega$, one has to devise appropriate versions of *Induction*, *Comprehension* (as well as of the counterpart of the defining axiom for Mealy machines see §8.2.6). We can ultimately rely here on the Weak Completeness of the realizability model (Prop. 7.4.4, §7.4) and assume for $\mathsf{LMSO}(\mathbf{T})^+$ the $(-)^T$ or $?(-)^Q$ translations of these axioms (see §8.1.7).

However, as discussed in Chap. 3, it is unlikely that the resulting axiomatization of $\mathsf{MSO}(\mathbf{T})^+$ is complete, and we might have, in some way or the other, to extend $\mathsf{MSO}(\mathbf{T})^+$ (and $\mathsf{LMSO}(\mathbf{T})^+$) with means of proving the positional determinacy of parity games (see §3.2.2). It is not clear to us what could be the best option. We nevertheless mention here possibility, that we unfortunately did not explore yet. As discussed in §3.4, the main difficulty for the positional determinacy of parity games in MSO is the Uniformization Lemma 3.4.1, and in particular the "*merging*" of (potentially infinitely many) strategies (see §3.4.2). Taking a point of view similar to that of §8.2.6, we might target an axiomatization of the realizability model of $\mathsf{LMSO}(\mathbf{T})^+$. In the realizability model, positionality of strategies is in general not assumed, but for the adequacy of the rule

$$(\textsc{Promotion})\ \frac{!\varphi_1,\ \ldots,\ !\varphi_n\ \vdash\ \varphi}{!\varphi_1,\ \ldots,\ !\varphi_n\ \vdash\ !\varphi}$$

As discussed in §7.4, the adequacy of (Promotion) relies on the fact that if the premise of the rule is realized, then it is realizable by a positional strategy (realization of the conclusion is then easy). A possibility that we did not explore yet, but which seems natural to us, would be to extend $\mathsf{LMSO}(\mathbf{T})^+$ with an axiom stating that there are "enough positional P-strategies for (Promotion)". This could be formulated with a non-standard rule expressing a form of *reflection*, as e.g. the following (in the language of LMSO):

$$\frac{\varphi_1, \ldots, \varphi_n \vdash \varphi}{\vdash (\exists \sigma)\big(\sigma \Vdash \mathcal{A}(\varphi_1) \otimes \cdots \otimes \mathcal{A}(\varphi_n) \multimap \mathcal{A}(\varphi)\big)} \qquad (\varphi_1, \ldots, \varphi_n \text{ (weakly) positive})$$

9. Conclusion

This document proposed what we see as a possible first step toward a Curry-Howard approach to MSO on infinite words and trees. To our opinion the result is contrasted.

On the positive side, we were surprised that, starting from a simple operational model as that of Chap. 5 and Chap. 7, linear logic so easily enters the picture as a natural setting to reason on automata, at least from the point of view of translations from MSO. In other words, there *is* a logical world at the level of automata, which is *different* from the classical world of MSO, and for which linear logic might provide a general setting. This is, to our opinion, stressed by the fact that the natural polarities of (a linear variant of) the Dialectica interpretation correspond to the usual polarities of automata (§7.5 and §8.2.6). While this has been worked out only in the case of ω words for now, we see no reasons why the Dialectica interpretation of LMSO($\mathbf{M}$) could not lift to LMSO($\mathbf{T}$) (excepted for completeness matters). As other examples we could mention the Weak Completeness of the realizability model (Prop. 7.4.4), which reduces (via Prop. 7.3.4) to a syntactic reasoning on linear implication, or the fact that natural translations of MSO formulae to tree automata are easily adapted to fit close to the usual *call-by-name* vs *call-by-value* setting (§8.1.7).

On the negative side, to our opinion the present work cannot be considered as a definitive answer to the initial objective. In the case of infinite trees, on the one hand we do not properly understand the axiomatics of MSO on infinite trees, while on the other hand we do not properly understand the notion of MSO positionality, at least when it comes to the adequacy of the (Promotion) rule of linear logic. In the case of ω-word, this results in a lack of structural results on game solving for Church's Synthesis. Some of the perspectives given below concern these questions.

9.1. Perspectives. First of all, it is clear to us that when it comes to linear logic, we pay the price of having the additive connectives, which are natural connectives on automata, with the expected polarities. While we plan to remedy this at some moment or the other, the reason for our initial choice was to have a setting which is sufficiently simple to give a broad overview.

Concerning the axiomatic part, while we conjecture that MSO($\mathbf{2}$) is incomplete, we have yet no any concrete idea on how to attack the problem, maybe by lack of knowledge on model construction for second order logic. The incompleteness of MSO($\mathfrak{D}$) may also follow from some simple argument that we haven't seen yet... It seems however that some improvements are possible regarding the axiom (PosDet). We think that the axiom (Def) of §3.4, which may look at bit rough, may lead to something reasonable with suitable reworking and refinement. Besides, it is also possible that when thought about in the setting of the realizability model, some specific but natural rules can give a complete (linear) system. In the direction of the reflection rule mentioned in §8.3, let us recall that [Möl02] has analyzed, in the setting of second-order arithmetic, the levels of the fixpoint hierarchy as reflection schemes. On the positive side, we may say that the formalization of Rabin's Tree Theorem in FSO($\mathfrak{D}$) may be valuable as it gives some ground on what we know to be provable in FSO($\mathfrak{D}$) without requiring further axioms.

As for the compositionality problem with the (Promotion) rule, there are more or less obvious things to look at from the perspective of exponentials in game semantics, even if this leads outside of MSO. The conclusion of [Rib18] contains some ideas in this direction. We do not repeat them here, but simply mention that a natural infinitary exponential in the setting of zigzag games is used in an interesting way in Pierre Pradic's PhD Thesis [Pra19]. Of course, the fundamental question is to gain a better understanding of MSO positionality, and in particular as to whether it is related to innocence (as reformulated in [Mel06]). The slight structure added by uniform automata to the usual setting, while pertinent to have good quantifications to reflect the usual polarities of automata, seems unrelated to the problem of positionality, which rather lives at the level of states of automata and propositional logic. As a result, it seems natural to think that a possibility would be to add structure on states of automata, so as to recover some form of arena games. This might lead to an odd setting from the point of view of automata theory, since one feature of automata is precisely to assume no structure on states. But from the point of view of MSO, maybe some insight can be gained from such approaches, in particular in presence of a logical setting at the level of automata. Concerning questions related to positionality, we must however say that from our point of view, one of the big mysteries remains the structure of the acceptance conditions resulting from McNaughton Determinization Theorem, as computed with any of the constructions we are aware of. On the other hand, the notion of *Split Tree* [Zie98] may provide some hints on how to manipulate the logical structure of acceptance conditions w.r.t. positionality matters.

Besides, note that uniform automata could very well be looked at from the usual perspective of the usual algebraic setting for recognizability, in the following sense. For a uniform automaton $\mathcal{A}$ with set of P moves U and set of O moves X, we have

$$\partial_{\mathcal{A}} \quad : \quad Q_{\mathcal{A}} \times \Sigma \longrightarrow U \times X \longrightarrow (\mathfrak{D} \longrightarrow Q_{\mathcal{A}}) \quad \simeq \quad \Sigma \times U \times X \times \mathfrak{D} \longrightarrow (Q_{\mathcal{A}} \longrightarrow Q_{\mathcal{A}})$$

so that the transition structure of $\mathcal{A}$ might described as an action on the monoid of functions $Q_{\mathcal{A}} \to Q_{\mathcal{A}}$.

An other trend concerns the relaxation of the realizability model to more liberal notions of transductions. We have began to look at what can be said in the setting of *rational relations* (see e.g. [Ber79]). An important point here is that as soon as we give up on causality, it seems much more difficult to keep a Cartesian structure w.r.t. alphabet multiplication, on which crucially rely quantifications. So such approaches may lead to two level models, with one notion of transduction for indexing and quantification, and a different notion of transduction for the realizers, thus requiring some explicit logical control on (existential) quantifications. In this direction, let us mention that *continuous* language reductions (*i.e.* the *Wadge hierarchy*)[1] play an important role in the study of the topological complexity of tree languages.[2]

A related question, concerning the burden of Notation 8.2.1, is the fact that our categorical models do not axiomatize the membership predicate, while known notions (see e.g. [Jac01, §5.2], [Pit02]) may be pertinent w.r.t. our setting. Actually, in the case of *finite* words (and forbidding universal automata) we can have a "*generic object*", namely the DFA $\boldsymbol{\varepsilon} : \mathbf{2}$ which accepts when it reads a 1. Then for any DFA $\mathcal{A} : \Sigma$, there is an eager Moore function $f : \Sigma \to_{\mathbf{EM}} \mathbf{2}$ (see §2.3) such that there is a Cartesian map $\mathcal{A} \to f^\star(\boldsymbol{\varepsilon})$. Note that this does not directly lift to ω-words, where we may rather have generic objects $\boldsymbol{\varepsilon}_{(\iota,\kappa)} : \{\iota, \dots, \kappa\}$ for each parity condition with range $\{\iota, \dots, \kappa\}$ (and hence have one total category of deterministic automata for each pair (ι, κ)).[3] This might suggest that suitable adaptations of our setting might be interesting to look at in

[1] Continuous reductions amount to causal reductions with ω words (see e.g. [PP04]) but not with infinite trees.

[2] We shall not give a bibliography, by lack of expertise on this evolving area.

[3] The languages of such $\boldsymbol{\varepsilon}_{(\iota,\kappa)}$ are variants of the usual *game languages* (used to describe the hierarchy of parity conditions in the case of infinite trees) which form a hierarchy w.r.t. continuous reductions which exhausts ω-regular languages (see e.g. [AN07] and references therein).

the case of finite words, at least for the structure it may uncover. We didn't followed this path in the work presented in this document because we thought that the inherent non-constructivity of MSO on infinite structures could provide a pertinent guiding principle.

In the setting of ω-words, an obvious question is whether $\mathsf{LMSO}(\mathfrak{C})$ could be complete if unrestricted exponentials were allowed.

Further interesting questions concern the extraction of realizers, taking inspiration from intuitionistic arithmetic (see e.g. [Koh08]). For instance, the realizability model of $\mathsf{LMSO}(\mathfrak{C})$ validates (for trivial reasons) a *Fan Rule* of the form

$$\vdash (\forall x^o)(\exists y^\iota)\delta^\pm(x,y) \quad \Rightarrow \quad \vdash (\forall x^o)(\exists y \leq N)\delta^\pm(x,y) \text{ for some } N \in \mathbb{N} \qquad (\delta^\pm \text{ deterministic})$$

This suggests to look at variants of Dialectica allowing to extract *bounds* instead of witnesses. Further, we plan to look at whether, variants of *Safra's Construction* for McNaughton's Determinization Theorem can allow to preserve some witnessing or bounding information, in sufficiently simple cases, in order to obtain variants of *Markov Rule* or of the Fan Rule.

An important direction of future research concerns (semi)automated proof search for (L)MSO. As far as full automation is targeted, this is fairly outside our area of expertise and we refer to the conclusion of [DR19] for a discussion.

In a similar vein, one may look at polarized fragments of LMSO from a syntactic perspective. On the first hand, in the setting of automata on infinite trees, the expressive power of deterministic automata is well known (see e.g. [AJFN08]). On the other hand, in the case of ω words (and assuming parity conditions), non-deterministic, deterministic, universal and alternating automata all have the same expressive power, so that the negative fragment of $\mathsf{MSO}(\mathbf{M})$ is as expressive as the full language.[4] This fact is in particular exploited in some *Safraless* approach to Church's Synthesis [KV05, KPV06, FJR11] (see §8.2.4). In both cases, it may be interesting to look precisely at inductively defined polarized fragments of LMSO, in particular because LMSO allows for an explicit control of the uses of Simulation (and McNaughton's Determinization in the case of ω-words).

At a more concrete level, interactive proof systems for MSO (as well as LMSO) could be implemented. The experience of [DR19] shows that it might indeed be humanly feasible to formally represent and prove things in suitable versions of MSO. In this direction, an important step is to have a suitable representation of Mealy machines (both at the level of the language of $(\mathsf{L})\mathsf{MSO}(\mathbf{T})$ or $(\mathsf{L})\mathsf{MSO}(\mathbf{M})$ and for implementing extraction). To this end, an interesting way to represent Mealy machines by means of a few simple combinators is proposed in [Pra19].

Last but not least, a remaining open question on the works presented in this document is their relation to Ong's Theorem [Ong06] (the decidability of Higher-Order Model Checking, see also [SW15, Mel17a]).

[4] The $\mathbf{M}$ is important there.

Bibliography

[Abr97] S. Abramsky. Semantics of Interaction. In A. M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, volume 14 of *Publications of the Newton Institute*, page 1. Cambridge University Press, 1997. 17, 56, 57, 69, 141

[AC98] R. M. Amadio and P.-L. Curien. *Domains and Lambda-Calculi*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1998. 47, 48, 55, 56, 96, 141

[AF98] J. Avigad and S. Feferman. Gödel's functional ("Dialectica") interpretation. In S. Buss, editor, *Handbook Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 337–405. Elsevier, 1998. 91, 109, 120, 123

[AJFN08] A. Arnold, Duparc J., Murlak F., and D. Niwinski. On the topological complexity of tree languages. In *Logic and Automata*, volume 2 of *Texts in Logic and Games*. Amsterdam University Press, 2008. 129

[AL17] B. Afshari and G. E. Leigh. Cut-free Completeness for Modal Mu-Calculus. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE Computer Society, 2017. 29

[AN07] A. Arnold and D. Niwinski. Continuous Separation of Game Languages. *Fundam. Inform.*, 81(1-3):19–28, 2007. 128

[BdRV02] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2002. 14

[Ber79] J. Berstel. *Transductions and context-free languages*. Leitfäden der angewandten Mathematik und Mechanik. Teubner, 1979. 128

[BG00] A. Blumensath and E. Grädel. Automatic Structures. In *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, June 26-29, 2000*, pages 51–62. IEEE Computer Society, 2000. 34

[BGG97] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997. 14

[BH18] G. Bellin and W. Heijltjes. Proof Nets for Bi-Intuitionistic Linear Logic. In Hélène Kirchner, editor, *FSCD*, volume 108 of *LIPIcs*, pages 10:1–10:18, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 116

[Bie96] G.M. Bierman. A note on full intuitionistic linear logic. *Annals of Pure and Applied Logic*, 79(3):281 – 287, 1996. 116

[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. 112

[BK08] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008. 112

[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. 22, 26, 35, 92, 94, 109, 111

[BL19] U. Boker and K. Lehtinen. Good for Games Automata: From Nondeterminism to Alternation. In Wan Fokkink and Rob van Glabbeek, editors, *30th International Conference on Concurrency Theory (CONCUR 2019)*, volume 140 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. 115

[Blu13] A. Blumensath. An algebraic proof of Rabin's Tree Theorem. *Theor. Comput. Sci.*, 478:1–21, 2013. 14, 29, 32

[BMSS12] L. Birkedal, R. E. Møgelberg, J. Schwinghammer, and K. Støvring. First steps in synthetic guarded domain theory: step-indexing in the topos of trees. *Logical Methods in Computer Science*, 8(4), 2012. 67, 71, 72, 80

[Bor94] F. Borceux. *Handbook of Categorical Algebra: Volume 1, Basic Category Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1994. 105

[BS73] J. R. Büchi and D. Siefkes. Axiomatization of the Monadic Second Order Theory of ω_1. In J. R. Büchi and D. Siefkes, editors, *Decidable Theories II : The Monadic Second Order Theory of All Countable Ordinals*, volume 328 of *LNM*, pages 129–217. Springer, 1973. 29

[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. 21, 32, 111

[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. 110

[CL07] A. Carayol and C. Löding. MSO on the Infinite Binary Tree: Choice and Order. In *CSL*, volume 4646 of *Lecture Notes in Computer Science*, pages 161–176. Springer, 2007. 34, 36, 37, 40

[CL08] T. Colcombet and C. Löding. The Non-deterministic Mostowski Hierarchy and Distance-Parity Automata. In *ICALP 2008*, volume 5126 of *Lecture Notes in Computer Science*, pages 398–409. Springer, 2008. 51, 97, 115

[CLNW10] A. Carayol, C. Löding, D. Niwinski, and I. Walukiewicz. Choice functions and well-orderings over the infinite binary tree. *Central European Journal of Mathematics*, 8(4):662–682, Aug 2010. 36, 37, 40

[Col09] T. Colcombet. The theory of stabilisation monoids and regular cost functions. In S. Albers, A. Marchetti-Spaccamela, Y. Matias, S. Nikoletseas, and W. Thomas, editors, *Proceedings of ICALP'09*, volume 5556 of *Lecture Notes in Computer Science*, pages 139–150, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. 115

[Dav64] M. Davis. Infinite Games of Perfect Information. In *Advances in Game Theory*, number 52 in Annals of Mathematics Studies, pages 85–102. Princeton University Press, 1964. 23

[DJS97] V. Danos, J.-B. Joinet, and H. Schellinx. A New Deconstructive Logic: Linear Logic. *Journal of Symbolic Logic*, 62(3):755–807, 1997. 100, 106

[Dou17] A. Doumane. Constructive completeness for the linear-time μ-calculus. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, 2017. 29

[dP87] V. de Paiva. The Dialectica Categories. In *Proc of Categories in Computer Science and Logic, Boulder, CO, 1987*, volume 92 of *Contemporary Mathematics*. American Mathematical Society, 1987. 109, 120

[dP89] V. de Paiva. A Dialectica-like Model of Linear Logic. In *Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*, pages 341–356. Springer, 1989. 120

[dP91] V. de Paiva. The Dialectica categories. Technical Report 213, University of Cambridge Computer Laboratory, January 1991. 71, 120

[DR15] A. Das and C. Riba. A complete axiomatization of mso on infinite trees. In *Proceedings of LICS'15*. IEEE Computer Society, 2015. 13, 29, 30

[DR19] A. Das and C. Riba. A Functional (Monadic) Second-Order Theory of Infinite Trees . Submitted. Available on HAL (hal-02067267) `https://hal.archives-ouvertes.fr/hal-02067267`, 2019. 13, 24, 29, 30, 32, 33, 34, 36, 37, 41, 43, 129

[EJ91] E. A. Emerson and C. S. Jutla. Tree Automata, Mu-Calculus and Determinacy (Extended Abstract). In *FOCS*, pages 368–377. IEEE Computer Society, 1991. 15, 16, 22, 23, 41, 55, 65, 83, 86, 94

[Fin16] B. Finkbeiner. Synthesis of Reactive Systems. In J. Esparza, O. Grumberg, and S. Sickert, editors, *Dependable Software Systems Engineering*, volume 45 of *NATO Science for Peace and Security Series, D: Information and Communication Security*, pages 72–98. IOS Press, 2016. 111, 112

[FJR11] E. Filiot, N. Jin, and J.-F. Raskin. Antichains and compositional algorithms for LTL synthesis. *Formal Methods in System Design*, 39(3):261–296, Dec 2011. 25, 112, 129

[FL12] O. Friedmann and M. Lange. Ramsey-Based Analysis of Parity Automata. In *Proceedings of TACAS 2012*, volume 7214 of *Lecture Notes in Computer Science*, pages 64–78. Springer, 2012. 112

[GH82] Y. Gurevich and L. Harrington. Trees, Automata, and Games. In *STOC*, pages 60–65. ACM, 1982. 15

[Gir87] J.-Y. Girard. Linear Logic. *Theoretical Computer Science*, 50:1–102, 1987. 16, 53, 54, 55, 92

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989. 17

[GS83] Y. Gurevich and S. Shelah. Rabin's Uniformization Problem. *J. Symb. Log.*, 48(4):1105–1119, 1983. 34, 36

[GtC09] A. Gheerbrant and B. ten Cate. Complete Axiomatizations of MSO, FO(TC1) and FO(LFP1) on Finite Trees. In *LFCS*, volume 5407 of *LNCS*, pages 180–196. Springer, 2009. 29, 32

[GtC12] A. Gheerbrant and B. ten Cate. Complete Axiomatizations of Fragments of Monadic Second-Order Logic on Finite Trees. *Logical Methods in Computer Science*, 8(4), 2012. 29, 32

[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. 14, 21, 23, 111, 112

[HdP93] M. Hyland and V. de Paiva. Full Intuitionistic Linear Logic (Extended Abstract). *Annals of Pure and Applied Logic*, 64(3), 1993. 116

[HHM07] R. Harmer, M Hyland, and P.-A. Melliès. Categorical combinatorics for innocent strategies. In *LICS 2007*, pages 379–388, 2007. 74, 76

[Hof11] P. J. W. Hofstra. The dialectica monad and its cousins. In M. Makkai and B.T. Hart, editors, *Models, Logics, and Higher-dimensional Categories: A Tribute to the Work of Mihály Makkai*, CRM proceedings & lecture notes. American Mathematical Society, 2011. 77

[HP06] T. A. Henzinger and N. Piterman. Solving Games Without Determinization. In Z. Ésik, editor, *Proceedings of CSL'06*, volume 4207 of *Lecture Notes in Computer Science*, pages 395–410, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 113, 115

[HS99] J. M. E. Hyland and A. Schalk. Abstract Games for Linear Logic. *Electr. Notes Theor. Comput. Sci.*, 29:127–150, 1999. 57, 67, 76

[HS03] J. M. E. Hyland and A. Schalk. Glueing and orthogonality for models of linear logic. *Theoretical Computer Science*, 294(1/2):183–231, 2003. 67, 71, 75, 76

[Hyl97] J. M. E. Hyland. Game Semantics. In A. M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, volume 14 of *Publications of the Newton Institute*, page 131. Cambridge University Press, 1997. 17, 56, 57, 141

[Hyl02] J. M. E. Hyland. Proof theory in the abstract. *Ann. Pure Appl. Logic*, 114(1-3):43–78, 2002. 71, 77, 120

[Jac01] B. Jacobs. *Categorical Logic and Type Theory*. Studies in logic and the foundations of mathematics. Elsevier, 2001. 47, 48, 67, 77, 78, 93, 128

[Jec06] T. Jech. *Set Theory: The Third Millennium Edition, revised and expanded*. Springer Monographs in Mathematics. Springer, 2006. 40

[Jut97] C. S. Jutla. Determinization and Memoryless Winning Strategies. *Inf. Comput.*, 133(2):117–134, 1997. 96

[Kai95] R. Kaivola. Axiomatising Linear Time Mu-calculus. In *CONCUR*, volume 962 of *Lecture Notes in Computer Science*, pages 423–437. Springer, 1995. 29

[KK95] N. Klarlund and D. Kozen. Rabin Measures. *Chicago J. Theor. Comput. Sci.*, 1995, 1995. 96

[Kla94] N. Klarlund. Progress measures, immediate determinacy, and a subset construction for tree automata. *Annals of Pure and Applied Logic*, 69(2-3):243–268, 1994. 41, 96

[KM16] L. A. Kołodziejczyk and H. Michalewski. How Unprovable is Rabin's Decidability Theorem? In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '16, pages 788–797. ACM, 2016. 41

[KMPS16] L. A. Kolodziejczyk, H. Michalewski, P. Pradic, and M. Skrzypczak. The Logical Strength of Büchi's Decidability Theorem. In J.-M. Talbot and L. Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:16, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 32

[Koh08] U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Springer Monographs in Mathematics. Springer, 2008. 20, 91, 109, 120, 123, 129

[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, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 111, 112, 129

[Kri11] J.-L. Krivine. Realizability algebras: a program to well order R. *Logical Methods in Computer Science*, 7(3), 2011. 30

[KS15] D. Kuperberg and M. Skrzypczak. On determinisation of good-for-games automata. In M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, editors, *Proceedings of ICALP'15*, volume 9135 of *Lecture Notes in Computer Science*, pages 299–310, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg. 115

[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. 20, 112, 129

[LLW08] LLWiki. LLWiki. `http://llwiki.ens-lyon.fr/mediawiki/`, 2008. 105, 106, 113

[LR03] O. Laurent and L. Regnier. About Translations of Classical Logic into Polarized Linear Logic. In *Proceedings of LICS'03*, pages 11–20. IEEE Computer Society Press, 2003. 106

[LS86] J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*. CUP, 1986. 47, 48

[Mar75] D. A. Martin. Borel Determinacy. *The Annals of Mathematics, Second Series*, 102(2):363–371, 1975. 41

[McN66] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Inf. Control*, 9(5):521 – 530, 1966. 20, 21, 30, 35, 97, 109, 111, 113

[Mel04] P.-A. Melliès. Comparing hierarchies of types in models of linear logic. *Inf. Comput.*, 189(2):202–234, 2004. 96

[Mel05] P.-A. Melliès. Sequential algorithms and strongly stable functions. *Theoretical Computer Science*, 343(1-2):237–281, 2005. 141

[Mel06] P.-A. Melliès. Asynchronous games 2: The true concurrency of innocence. *Theor. Comput. Sci.*, 358(2-3):200–228, 2006. 128

[Mel09] P.-A. Melliès. Categorical semantics of linear logic. In *Interactive models of computation and program behaviour*, volume 27 of *Panoramas et Synthèses*. SMF, 2009. 47, 48, 55, 75, 76, 88, 93, 96

[Mel13] P.-A. Melliès. On dialogue games and coherent strategies. In *CSL*, volume 23 of *LIPIcs*, pages 540–562. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013. 16, 100

[Mel17a] P.-A. Melliès. Higher-order parity automata. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12, 2017. 129

[Mel17b] P.-A. Melliès. Une étude micrologique de la négation. Habilitation Thesis, University Paris Diderot, 2017. 16, 100

[Miq11] A. Miquel. Forcing as a Program Transformation. In *Proceedings of LICS'11*, pages 197–206. IEEE Computer Society, 2011. 30

[ML98] S. Mac Lane. *Categories for the Working Mathematician*. Springer, 2nd edition, 1998. 76

[MLM92] S. Mac Lane and I. Moerdijk. *Sheaves in geometry and logic: A first introduction to topos theory*. Springer, 1992. 71

[MM15] K. Maillard and P.-A. Melliès. A Fibrational Account of Local States. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 402–413. IEEE Computer Society, 2015. 77

[Möl02] M. Möllerfeld. *Generalized Inductive Definitions - The μ-calculus and Π^1_2-comprehension*. PhD thesis, Westfälischen Wilhelms-Universität Münster, 2002. Available at `https://miami.uni-muenster.de/Record/9dfa74b6-186b-4e95-a51f-9965d7e1e508`. 127

[Mos09] Y.N. Moschovakis. *Descriptive Set Theory*. Mathematical surveys and monographs. American Mathematical Society, 2009. 41

[MS87] D. E. Muller and P. E. Schupp. Alternating Automata on Infinite Trees. *Theor. Comput. Sci.*, 54:267–276, 1987. 15, 16, 22, 55, 94

[MS95] D. E. Muller and P. E Schupp. Simulating Alternating Tree Automata by Nondeterministic Automata: New Results and New Proofs of the Theorems of Rabin, McNaughton and Safra. *Theor. Comput. Sci.*, 141(1&2):69–107, 1995. 15, 16, 22, 35, 55, 65, 83, 86, 94, 95, 97

[MS11] A. Montalbán and R. A. Shore. The limits of determinacy in second-order arithmetic. *Proceedings of the London Mathematical Society*, 2011. 41

[MT07] M. O. MedSalem and K. Tanaka. Δ^0_3-determinacy, comprehension and induction. *J. Symb. Log.*, 72(2):452–462, 2007. 41

[Ong06] C.-H. L. Ong. On Model-Checking Trees Generated by Higher-Order Recursion Schemes. In *Proceedings of LICS'06*, pages 81–90. IEEE Computer Society, 2006. 53, 129

[Pit02] A. M. Pitts. Tripos theory in retrospect. *Mathematical Structures in Computer Science*, 12:265 – 279, 2002. 128

[PP04] D. Perrin and J.-É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Pure and Applied Mathematics. Elsevier, 2004. 14, 23, 32, 90, 96, 128

[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. 13, 47, 117, 137

[PR18a] P. Pradic and C. Riba. A Curry-Howard Approach to Church's Synthesis. Full version of [PR17]. Under revision. Available on HAL (hal-01780807) https://hal.archives-ouvertes.fr/hal-01780807, 2018. 13

[PR18b] P. Pradic and C. Riba. LMSO: A Curry-Howard Approach to Church's Synthesis via Linear Logic. In *LICS*. ACM, 2018. 13, 47, 53, 67, 83, 97, 100, 110, 114, 115, 117, 118

[PR19] P. Pradic and C. Riba. A Dialectica-Like Interpretation of a Linear MSO on Infinite Words. In *FoSSaCS*. Springer, 2019. 13, 47, 67, 79, 80, 91, 100, 110, 114, 117, 118, 119, 120, 121

[Pra19] P. Pradic. Proof-Theoretical Approaches to MSO. Forthcoming, 2019. 79, 83, 85, 91, 97, 100, 110, 128, 129

[Rab69] M. O. Rabin. Decidability of Second-Order Theories and Automata on Infinite Trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969. 13, 14, 15, 36, 38, 96

[Rab72] M. O. Rabin. Automata on infinite objects and Church's Problem. *Amer. Math. Soc.*, 1972. 35, 94, 111

[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. Full version available on HAL (hal-00692153) https://hal.archives-ouvertes.fr/hal-00692153. 13, 29, 30, 32

[Rib13] C. Riba. Forcing MSO on Infinite Words in Weak MSO. In *LICS*, pages 448–457. IEEE Computer Society, 2013. 13, 29

[Rib15] C. Riba. Fibrations of tree automata. In *TLCA*, volume 38 of *LIPIcs*, pages 302–316. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. Full version available on HAL (hal-01261182) https://hal.archives-ouvertes.fr/hal-01261182. 13, 47, 50, 54, 55, 57, 61, 62, 65, 67, 84, 96

[Rib18] C. Riba. Monoidal-Closed Categories of Tree Automata. Under revision. Available on HAL (hal-01261183) https://hal.archives-ouvertes.fr/hal-01261183, 2018. 13, 47, 61, 64, 65, 67, 69, 71, 75, 76, 78, 83, 85, 86, 92, 95, 97, 105, 128

[Ros92] R. Rosner. *Modular Synthesis of Reactive Systems.* PhD thesis, Weismann Institute of Science, 1992. 112

[SA05] L. Santocanale and A. Arnold. Ambiguous classes in mu-calculi hierarchies. *Theor. Comput. Sci.*, 333(1-2):265–296, 2005. 97

[Sha91] S. Shapiro. *Foundations without Foundationalism: A Case for Second-Order Logic.* Oxford University Press, 1991. 30, 32

[Shi06] M. Shirahata. The Dialectica Interpretation of First-Order Classical Affine Logic. *Theory and Applications of Categories*, 17(4):49–79, 2006. 120

[Sie70] D. Siefkes. *Decidable Theories I : Büchi's Monadic Second Order Successor Arithmetic*, volume 120 of *LNM.* Springer, 1970. 29, 30, 32, 35, 109, 118

[Sim10] S.G. Simpson. *Subsystems of Second Order Arithmetic.* Perspectives in Logic. Cambridge University Press, 2nd edition, 2010. 29, 30, 40, 41, 42

[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. 17, 20

[SV10] L. Santocanale and Y. Venema. Completeness for flat modal fixpoint logics. *Ann. Pure Appl. Logic*, 162(1):55–82, 2010. 29

[SW15] S. Salvati and I. Walukiewicz. Using models to model-check recursive schemes. *Logical Methods in Computer Science*, 11(2), 2015. 129

[Tan91] K. Tanaka. Weak axioms of determinacy and subsystems of analysis II (Σ^0_2 games). *Annals of Pure and Applied Logic*, 52:181–193, 1991. 41

[tCF10] B. ten Cate and G. Fontaine. An Easy Completeness Proof for the Modal μ-Calculus on Finite Trees. In *FOSSACS*, volume 6014 of *Lecture Notes in Computer Science*, pages 161–175. Springer, 2010. 29

[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. 14, 21, 23, 35, 41, 43, 90, 92, 94, 96, 97, 110, 111

[Tho08] W. Thomas. Solution of Church's Problem: A tutorial. *New Perspectives on Games and Interaction*, 5:23, 2008. 25, 110, 111

[Tro73] A.S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *LNM.* Springer-Verlag, 1973. 123

[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. 14

[Wal00] I. Walukiewicz. Completeness of Kozen's Axiomatisation of the Propositional μ-Calculus. *Information and Computation*, 157(1-2):142–182, 2000. 29

[Wal02] I. Walukiewicz. Monadic second-order logic on tree-like structures. *Theor. Comput. Sci.*, 275(1-2):311–346, 2002. 15, 22, 24, 35, 66, 94, 105, 110

[Zie98] Z. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998. 41, 96, 128

A. A Setting of Simple Games

We present here the basic setting of games on which most of this document (and all of Part II) is based. It consists of tree games, essentially corresponding to the games known as *simple games* in the litterature (see e.g. [Abr97, Hyl97]). We shall not provide any genral introduction to game semantics in the realm of the Curry-Howard correspondance, but jsut mention that *simple games* stem from Berry & Curien's *sequential data structures* (see e.g. [AC98, Chap. 14]) and refer to [Mel05] for further references and discussions.

Definition A.0.1 (Simple Games)**.**

- *A* simple game *A has the form*

$$A \quad = \quad (A_{\mathsf{P}},\, A_{\mathsf{O}},\, L_A)$$

where A_{P} is the set of P-moves*, A_{O} is the set of* O-moves *and $L_A \subseteq (A_{\mathsf{P}} + A_{\mathsf{O}})^*$ is a non-empty prefix-closed set of* legal plays.

We let $s, t, \ldots$ range of over plays and $m, n, \ldots$ range over moves.

We often write L_A^{even} for the set of even-length legal plays of A.

- *A* simple game with winning *is a simple game A equipped with a* set of winning plays *(or* winning condition*) $\mathcal{W}_A \subseteq (A_{\mathsf{P}} + A_{\mathsf{O}})^\omega$.*

- *The sets $\wp_A^{\mathsf{P}}$ and $\wp_A^{\mathsf{O}}$ of resp.* positive *and* negative *plays on A are*

$$\begin{array}{lcl} \wp_A^{\mathsf{P}} & := & (A_{\mathsf{P}} \cdot A_{\mathsf{O}})^* + (A_{\mathsf{P}} \cdot A_{\mathsf{O}})^* \cdot A_{\mathsf{P}} \\ \wp_A^{\mathsf{O}} & := & (A_{\mathsf{O}} \cdot A_{\mathsf{P}})^* + (A_{\mathsf{O}} \cdot A_{\mathsf{P}})^* \cdot A_{\mathsf{O}} \end{array}$$

The game A is positive *(resp.* negative*) if all its legal plays are positive (resp. negative), that is if $L_A \subseteq \wp_A^{\mathsf{P}}$ (resp. $L_A \subseteq \wp_A^{\mathsf{O}}$). So P starts in a positive game and O starts in a negative one. A game is* polarized *if it is either positive or negative.*

- *A play for player $\mathsf{J} \in \{\mathsf{P}, \mathsf{O}\}$ (also called a* J-play*) is either the empty play or a non-empty play in which J played last (i.e. which ends with a J-move).*

Example A.0.2 (Full Postitive Games)**.** Full positive games *are positives games A whose positive plays are all legal, that is such that*

$$L_A \quad = \quad \wp_A^{\mathsf{P}} \quad = \quad (A_{\mathsf{P}} \cdot A_{\mathsf{O}})^* + (A_{\mathsf{P}} \cdot A_{\mathsf{O}})^* \cdot A_{\mathsf{P}}$$

Hence, a full positive game A is completely characterized by its set of P and O-moves. We can thus simply omit legal plays in the description of full positive games. We moreover often write U (resp. X) for the set of P-moves (resp. of O-moves) of a full positive game A.

We now come to the definition of strategies in simple games. A strategy for player P or O is what one expects. The formal definition of strategy below emphasizes P (strategies for O are defined by duality), because in categories of games, composition and identities are only defined for the strategies of the negative player P. Moreover, the manipulation of strategies as morphisms is more convenient when strategies are presented as sets of plays rather than as functions on plays.

Definition A.0.3 (Strategies)**.** *A* P-strategy *on A is a non-empty set of legal P-plays $\sigma \subseteq L_A$ which is*

- P***-prefix-closed:*** *if $s.t \in \sigma$ and s is a P-play then $s \in \sigma$, and*
- P***-deterministic:*** *if $s.n \in \sigma$ and $s.m \in \sigma$ then $n = m$.*

Consider now a polarized game with winning A. Given a P-strategy σ on A and an O-play $s \in L_A$, we say that s is an O-interrogation of σ *if either $s = \varepsilon$ and A is positive, or if $s = t.m$ for some $t \in \sigma$. We say that σ is* total *if for every O-interrogation s of σ, we have $s.n \in \sigma$ for some n. A* winning *(P-)strategy on A is a total strategy σ s.t. for all $\chi \in (A_\mathsf{P} + A_\mathsf{O})^\omega$, we have $\chi \in \mathcal{W}_A$ whenever $\exists^\infty k \in \mathbb{N}.\ \chi(0) \cdot \ldots \cdot \chi(k) \in \sigma$.*

The notion of (total, winning) O-strategy is defined by duality. Each game A has a dual $\overline{A} = (A_\mathsf{O}, A_\mathsf{P}, L_A)$, where we moreover let $\mathcal{W}_{\overline{A}} := (A_\mathsf{P} + A_\mathsf{O})^\omega \setminus \mathcal{W}_A$ if A is a game with winning. Note that $\overline{A}$ is polarized iff A is polarized, and that $\overline{A}$ is positive (resp. negative) iff A is negative (resp. positive). Then, we say that a (total, winning) O-strategy on A is a (total, winning) P-strategy on $\overline{A}$.

Example A.0.4 (Strategies in Full Postitive Games)**.** *Consider a total full positive game $A = (U, X)$. A P-strategy σ in A is a non-empty set of sequences of the form*

$$s \quad = \quad u_0 \cdot x_1 \cdot u_1 \cdot \ldots \cdot x_{n-1} \cdot u_{n-1}$$

such that

$$s \cdot x_n \cdot u_n \ \in \ \sigma \quad \Rightarrow \quad s \in \sigma$$

and

$$s \cdot x_n \cdot u_n \ , \ s \cdot x_n \cdot u'_n \ \in \ \sigma \quad \Rightarrow \quad u_n = u'_n$$

Hence, as expected, in a play

$$u_0 \cdot x_1 \cdot u_1 \cdot \ldots \cdot x_n \cdot u_n \ \in \ \sigma$$

the moves u_i are uniquely determined by the O-moves $x_1, \ldots, x_{i-1}$. Moreover, σ is total iff for every

$$u_0 \cdot x_1 \cdot u_1 \cdot \ldots \cdot x_n \cdot u_n \ \in \ \sigma$$

and for every O-move x_{n+1}, there is some u_{n+1} such that

$$u_0 \cdot x_1 \cdot u_1 \cdot \ldots \cdot x_n \cdot u_n \cdot x_{n+1} \cdot u_{n+1} \ \in \ \sigma$$

In other words, total P-strategies in a total full positive game $A = (U, X)$ are given by functions $X^* \to U$.

B. Proof of Proposition 3.4.2

Consider HF-sets K, L and fix MC and φ as required for axiom (Def). Let

$$\mathcal{S} \quad := \quad K^{\mathfrak{D}^* \times L}$$

Given $n \in \mathbb{N}$, let T_n be the finite tree $\mathfrak{D}^{\leq n}$. We build a sequence $(s_n)_{n\in\mathbb{N}} \in \mathcal{S}$, with for each $n \in \mathbb{N}$,

- $\varphi(s_n, T_n)$,
- $s_n =_{\mathrm{MC}(s_n,T_n)} s_{n+1}$,
- $\mathrm{MC}(s_n, T_n) \subseteq \mathrm{MC}(s_{n+1}, T_{n+1})$.

First, since $T_0 = \{\varepsilon\}$, thanks to assumption (c) of (Def), we take $s_0 \in \mathcal{S}$ such that $\varphi(s_0, T_0)$ holds. Assume now that s_n has been defined. Let $U_0, \ldots, U_k$ be finite trees with

$$U_0 = T_n \quad \text{and} \quad (\forall i < k)(\exists p \in \mathfrak{D}^*)(U_{i+1} = U_i + \{p\}) \quad \text{and} \quad U_k = T_{n+1}$$

We inductively define $u_0, \ldots, u_k \in \mathcal{S}$ with

- $\varphi(u_i, U_i)$,
- $u_0 =_{\mathrm{MC}(u_0,U_0)} u_i$,
- $\mathrm{MC}(u_0, U_0) \subseteq \mathrm{MC}(u_i, U_i)$,

For the base case, we let $u_0 := s_n$. Let now $i < k$ and assume that u_i has been defined. Since $\varphi(u_i, U_i)$ holds, by assumption on U_{i+1} it follows from assumption (d) of (Def) that there is $u_{i+1} \in \mathcal{S}$ such that $\varphi(u_{i+1}, U_{i+1})$ holds and

$$\mathrm{MC}(u_i, U_i) \subseteq \mathrm{MC}(u_{i+1}, U_{i+1}) \quad \text{and} \quad u_{i+1} =_{\mathrm{MC}(u_i,U_i)} u_i$$

We have

$$\mathrm{MC}(u_0, U_0) \quad \subseteq \quad \mathrm{MC}(u_i, U_i) \quad \subseteq \quad \mathrm{MC}(u_{i+1}, U_{i+1})$$

and furthermore

$$\begin{aligned} u_{i+1}{\restriction}\mathrm{MC}(u_0, U_0) &= (u_{i+1}{\restriction}\mathrm{MC}(u_i, U_i)){\restriction}\mathrm{MC}(u_0, U_0) \\ &= (u_i{\restriction}\mathrm{MC}(u_i, U_i)){\restriction}\mathrm{MC}(u_0, U_0) \\ &= u_i{\restriction}\mathrm{MC}(u_0, U_0) \\ &= u_0{\restriction}\mathrm{MC}(u_0, U_0) \end{aligned}$$

This completes the definition of $u_0, \ldots, u_k$.

We now take $s_{n+1} := u_k$. We therefore have $s_{n+1} \in \mathcal{S}$, as well as $\varphi(s_{n+1}, T_{n+1})$ and

$$s_{n+1} =_{\mathrm{MC}(s_n,T_n)} s_n \qquad \text{and} \qquad \mathrm{MC}(s_n, T_n) \subseteq \mathrm{MC}(s_{n+1}, T_{n+1})$$

This completes the definition of $(s_n)_{n\in\mathbb{N}}$. An easy induction shows the following claim.

Claim B.0.1. *For $k \leq n$, we have*

$$\mathrm{MC}(s_k, T_k) \quad \subseteq \quad \mathrm{MC}(s_n, T_n) \qquad \textit{and} \qquad s_n =_{\mathrm{MC}(s_k,T_k)} s_k$$

Proof. The first part can be proved by induction on $n \geq k$ (for fixed k) using the fact that $\mathrm{MC}(s_n, T_n) \subseteq \mathrm{MC}(s_{n+1}, T_{n+1})$.

Consider now the second part. Fix k and reason by induction on $n \geq k$. The base case $n = k$ is obvious. For the induction step, by induction hypothesis we have

$$s_k =_{\mathrm{MC}(s_k,T_k)} s_n$$

But by the first part, we have

$$\mathrm{MC}(s_k, T_k) \quad \subseteq \quad \mathrm{MC}(s_n, T_n)$$

Hence

$$\begin{array}{rcl} s_n{\restriction}\mathrm{MC}(s_k, T_k) & = & (s_n{\restriction}\mathrm{MC}(s_n, T_n)){\restriction}\mathrm{MC}(s_k, T_k) \\ & = & (s_{n+1}{\restriction}\mathrm{MC}(s_n, T_n)){\restriction}\mathrm{MC}(s_k, T_k) \\ & = & s_{n+1}{\restriction}\mathrm{MC}(s_k, T_k) \end{array}$$

□

We are now going to define some $s \in \mathcal{S}$ such that $\varphi(s, T)$ holds for each finite tree T. First, let

$$\tilde{s} \quad := \quad \bigcup_{n \in \mathbb{N}} s_n{\restriction}\mathrm{MC}(s_n, T_n)$$

It follows from Claim B.0.1 that given $x \in \mathrm{MC}(s_k, T_k)$, for all $n \geq k$ we have $x \in \mathrm{MC}(s_n, T_n)$ and $s_k(x) = s_n(x)$. Hence $\tilde{s}$ is a function from $\bigcup_{n \in \mathbb{N}} \mathrm{MC}(s_n, T_n)$ to K. We extend it to a total map $s \in \mathcal{S}$ by putting $s(x) := s_0(x)$ for $x \notin \bigcup_{n \in \mathbb{N}} \mathrm{MC}(s_n, T_n)$. It easily follows from the definition of s that its restriction to $\mathrm{MC}(s_n, T_n)$ agrees with s_n:

Claim B.0.2. *For all $n \in \mathbb{N}$ we have $s =_{\mathrm{MC}(s_n,T_n)} s_n$.*

Proof. Fix $n \in \mathbb{N}$ and consider $x \in \mathrm{MC}(s_n, T_n)$. Then $s(x) = \kappa \in K$ iff $s_j(x) = \kappa$ for some $j \in \mathbb{N}$ such that $x \in \mathrm{MC}(s_j, T_j)$. But for all $\ell \geq n$ we have $x \in \mathrm{MC}(s_\ell, T_\ell)$ and Claim B.0.1 implies $s_n(x) = s_\ell(x)$. Moreover, for all $i \leq n$, if $x \in \mathrm{MC}(s_i, T_i)$ then Claim B.0.1 also implies $s_n(x) = s_i(x)$. Hence for all $j \in \mathbb{N}$, if $x \in \mathrm{MC}(s_j, T_j)$ then $s_j(x) = s_n(x)$ and the result follows. □

It then follows that

Claim B.0.3. *$\varphi(s, T)$ holds for each finite tree $T \subseteq \mathfrak{D}^*$.*

Proof. Let T be a finite tree and let $T_n \supseteq T$. By assumption (b) it is sufficient to show $\varphi(s, T_n)$. But by assumption (a), we are done since $\varphi(s_n, T_n)$ holds and since $s =_{\mathrm{MC}(s_n,T_n)} s_n$ by the above Claim B.0.2. □