Fibrations of Tree Automata

Colin Riba

ENS de Lyon, Université de Lyon, LIP*
colin.riba@ens-lyon.fr http://perso.ens-lyon.fr/colin.riba/

Abstract

We propose a notion of morphisms between tree automata based on game semantics. Morphisms are winning strategies on a synchronous restriction of the linear implication between acceptance games. This leads to split indexed categories, with substitution based on a suitable notion of synchronous tree function. By restricting to tree functions issued from maps on alphabets, this gives a fibration of tree automata. We then discuss the (fibrewise) monoidal structure issued from the synchronous product of automata. We also discuss how a variant of the usual projection operation on automata leads to an existential quantification in the fibered sense. Our notion of morphism is correctin the sense that it respects language inclusion, and in a weaker sense also complete.

1998 ACM Subject Classification F.3.2 Semantics of Programming Languages. F.4.1 Mathematical Logic. F.4.2 Formal Languages.

Keywords and phrases Tree automata, Game semantics, Categorical logic.

1 Introduction

This paper proposes a notion of morphism between tree automata based on game semantics. We follow the Curry-Howard-like slogan: *Automata as objects, Executions as morphisms.*

We consider general alternating automata on infinite ranked trees. These automata encompass Monadic Second-Order Logic (MSO) and thus most of the logics used in verification [6]. Tree automata are traditionally viewed as positive objects: one is primarily interested in satisfaction or satisfiability, and the primitive notion of quantification is existential. In contrast, Curry-Howard approaches tend to favor proof-theoretic oriented and negative approaches, *i.e.* approaches in which the predominant logical connective is the implication, and where the predominant form of quantification is universal.

We consider full infinite ranked trees, built from a non-empty finite set of directions D and labeled in non-empty finite alphabets Σ. The base category **Tree** has alphabets as objects and morphisms from Σ to Γ are $(\Sigma \to \Gamma)$-labeled D-ary trees.

The fibre categories are based on a generalization of the usual acceptance games, where for an automaton $\mathcal{A}$ on alphabet Γ (denoted $\Gamma \vdash \mathcal{A}$), input characters can be precomposed with a tree morphism $M \in \mathbf{Tree}[\Sigma, \Gamma]$, leading to substituted acceptance games of type $\Sigma \vdash \mathcal{G}(\mathcal{A}, M)$. Usual acceptance games, which correspond to the evaluation of $\Sigma \vdash \mathcal{A}$ on a Σ-labeled input tree, are substituted acceptance games $\mathbf{1} \vdash \mathcal{G}(\mathcal{A}, t)$ with $t \in \mathbf{Tree}[\mathbf{1}, \Sigma]$. Games of the form $\Sigma \vdash \mathcal{G}(\mathcal{A}, M)$ are the objects of the fibre category over Σ.

For morphisms, we introduce a notion of "synchronous" simple game between acceptance games. We rely on Hyland & Schalk's functor (denoted HS) from simple games to **Rel** [9]. A synchronous strategy $\Sigma \vdash \sigma : \mathcal{G}(\mathcal{A}, M) \multimap_{\circledast} \mathcal{G}(\mathcal{B}, N)$ is a strategy in the simple game $\mathcal{G}(\mathcal{A}, M) \multimap \mathcal{G}(\mathcal{B}, N)$ required to satisfy (in **Set**) a diagram of the form of (1) below,

* UMR 5668 CNRS ENS Lyon UCBL INRIA

expressing that $\mathcal{A}$ and $\mathcal{B}$ are evaluated along the same path of the tree and read the same input characters:

$$\begin{array}{ccc} \mathrm{HS}(\sigma) & \longrightarrow & \mathcal{G}(\mathcal{B}, N) \\ \downarrow & & \downarrow \\ \mathcal{G}(\mathcal{A}, M) & \longrightarrow & (D+\Sigma)^* \end{array} \tag{1}$$

This gives a split fibration game of tree automata and acceptance games. When restricting the base to *alphabet* morphisms (*i.e.* functions $\Sigma \to \Gamma$), substitution can be internalized in automata. By change-of-base of fibrations, this leads to a split fibration aut. In the fibers of aut, the substituted acceptance games have finite-state winning strategies, whose existence can be checked by trivial adaptation of usual algorithms.

Each of these fibrations is monoidal in the sense of [17], by using a natural synchronous product of tree automata. We also investigate a linear negation, as well as existential quantifications, obtained by adapting the usual projection operation on non-deterministic automata to make it a left-adjoint to weakening, the adjunction satisfying the usual Beck-Chevalley condition.

Our linear implication of acceptance games seems to provide a natural notion of prenex universal quantification on automata not investigated before. As expected, if there is a synchronous winning strategy $\sigma \Vdash \mathcal{A} \multimap\!\!\circledast\, \mathcal{B}$, then $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$ (*i.e.* each input tree accepted by $\mathcal{A}$ is also accepted by $\mathcal{B}$). Under some assumptions on $\mathcal{A}$ and $\mathcal{B}$ the converse holds: $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$ implies $\sigma \Vdash \mathcal{A} \multimap\!\!\circledast\, \mathcal{B}$ for some σ.

At the categorical level, thanks to (1), the constructions mimic relations in slices categories $\mathbf{Set}/(D+\Sigma)^*$ of the co-domain fibration: substitution is given by a (well chosen) pullback, and the monoidal product of automata is issued from the Cartesian product of plays in $\mathbf{Set}/(D+\Sigma)^*$ (*i.e.* also by a well chosen pullback).

The paper is organized as follows. Section 2 presents notations for trees and tree automata. Our notions of substituted acceptance games and synchronous arrow games are then discussed in Sect. 3. Substitution functors and the corresponding fibrations are presented in Sect. 4, and Section 5 overviews the monoidal structure. We then state our main correctness results in Sect. 6. Section 7 presents existential quantifications and quickly discusses non-deterministic automata. A short Appendix A gives some definitions on simple games, and a long version of the paper with full proofs [16] can be found on the webpage of the author.

2 Preliminaries

Fix a singleton set $\mathbf{1} = \{\bullet\}$ and a finite non-empty set D of (tree) *directions*.

Alphabets and Trees. We write $\Sigma, \Gamma, \ldots$ for *alphabets*, *i.e.* finite non-empty sets. We let $\mathbf{Alph}$ be the category whose objects are alphabets and whose morphisms $\beta \in \mathbf{Alph}[\Sigma, \Gamma]$ are functions $\beta : \Sigma \to \Gamma$.

We let $\mathbf{Tree}[\Sigma]$ be the set of Σ-labeled full D-ary trees, *i.e.* the set of maps $T : D^* \to \Sigma$. Let $\mathbf{Tree}$ be the category with *alphabets* as objects and with morphisms $\mathbf{Tree}[\Sigma, \Gamma] := \mathbf{Tree}[(\Sigma \to \Gamma)]$, *i.e.* $(\Sigma \to \Gamma)$-labeled trees. Maps $M \in \mathbf{Tree}[\Sigma, \Gamma]$ and $L \in \mathbf{Tree}[\Gamma, \Delta]$ are composed as

$$L \circ M \quad : \quad p \in D^* \quad \mapsto \quad (a \in \Sigma \quad \mapsto \quad L(p)(M(p)(a)))$$

and the identity $\mathrm{Id}_\Sigma \in \mathbf{Tree}[\Sigma, \Sigma]$ is defined as $\mathrm{Id}_\Sigma(p)(a) := a$. Note that $\mathbf{Tree}[\mathbf{1}, \Sigma]$ is in bijection with $\mathbf{Tree}[\Sigma]$.

There is a faithful functor from **Alph** to **Tree**, mapping $\beta \in \mathbf{Alph}[\Sigma, \Gamma]$ to the constant tree morphism $(\_ \mapsto \beta) \in \mathbf{Tree}[\Sigma, \Gamma]$ that we simply write β.

Tree Automata. Alternating tree automata [14] are finite state automata running on full infinite Σ-labeled D-ary trees. Their distinctive feature is that transitions are given by *positive Boolean* formulas with atoms pairs (q, d) of a state q and a tree direction $d \in D$ ((q, d) means that one copy of the automaton should start in state q from the d-th son of the current tree position).

Acceptance for alternating tree automata can be defined either *via* run trees or *via* the existence of winning strategies in *acceptance games* [14]. In both cases, we can w.l.o.g. restrict to transitions given by formulas in (irredundant) disjunctive normal form [15]. In our setting, it is quite convenient to follow the presentation of [19], in which disjunctive normal forms with atoms in $Q \times D$ are represented as elements of $\mathcal{P}(\mathcal{P}(Q \times D))$.

An alternating tree automaton $\mathcal{A}$ on alphabet Σ has the form $(Q, q^\imath, \delta, \Omega)$ where Q is the finite set of states, $q^\imath \in Q$ is the *initial state*, the *acceptance condition* is $\Omega \subseteq Q^\omega$ and following [19], the *transition function* δ has the form

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

We write $\Sigma \vdash \mathcal{A}$ if $\mathcal{A}$ is a tree automaton on Σ. Usual acceptance games are described in Sec. 3.1. It is customary to put restrictions on the acceptance condition $\Omega \subseteq Q^\omega$, typically by assuming it is generated from a *Muller* family $\mathcal{F} \in \mathcal{P}(\mathcal{P}(Q))$ as the set of $\pi \in Q^\omega$ such that $\mathrm{Inf}(\pi) \in \mathcal{F}$. We call such automata *regular*[1]. They have decidable emptiness checking and the same expressive power as MSO on D-ary trees (see e.g. the survey [18]).

3 Categories of Acceptance Games and Automata

We present in this Section the categories $\mathbf{SAG}^{(\mathrm{W})}_\Sigma$ of *substituted acceptance games*. Their objects will be *substituted acceptance games* (to be presented in Sect. 3.1) and their morphisms will be strategies in corresponding *synchronous arrow games* (to be presented in Sect. 3.2). Substituted acceptance games and synchronous arrow games are the two main notions we introduce in this paper. Our categories of $\mathbf{Aut}^{(\mathrm{W})}_\Sigma$ of automata will be full subcategories of $\mathbf{SAG}^{(\mathrm{W})}_\Sigma$, while $\mathbf{SAG}^{(\mathrm{W})}_\Sigma$ and $\mathbf{Aut}^{(\mathrm{W})}_\Sigma$ will be the total categories of our fibrations

$$\mathsf{game}^{(\mathrm{W})} \quad : \quad \mathbf{SAG}^{(\mathrm{W})} \quad \longrightarrow \quad \mathbf{Tree} \qquad\qquad \mathsf{aut}^{(\mathrm{W})} \quad : \quad \mathbf{Aut}^{(\mathrm{W})} \quad \longrightarrow \quad \mathbf{Alph}$$

to be presented in Sect. 4. Appendix A summarizes the basic notion of games we are using.

3.1 Substituted Acceptance Games

Consider a tree automaton $\mathcal{A} = (Q, q^\imath, \delta, \Omega)$ on Γ and a morphism $M \in \mathbf{Tree}[\Sigma, \Gamma]$. The *substituted acceptance game* $\Sigma \vdash \mathcal{G}(\mathcal{A}, M)$ is the positive game

$$\mathcal{G}(\mathcal{A}, M) \quad := \quad (D^* \times (A_\mathsf{P} + A_\mathsf{O}), E, *, \lambda, \xi, \mathcal{W})$$

[1] By adding states to $\mathcal{A}$ if necessary, one can describe Ω by an equivalent *parity* condition.

whose positions are given by $A_{\mathsf{P}} := Q$ and $A_{\mathsf{O}} := \Sigma \times \mathcal{P}(Q \times D)$, whose polarized root is $* := (\varepsilon, q^\imath)$ with $\xi(*) = \mathsf{P}$, whose polarized moves (E, λ) are given by

$$\begin{array}{lrcll} \text{from } (D^* \times A_{\mathsf{P}}) \text{ to } (D^* \times A_{\mathsf{O}}) : & (p, q) & \xrightarrow{\mathsf{P}} & (p, a, \gamma) & \text{iff } \gamma \in \delta(q, M(p)(a)) \\ \text{from } (D^* \times A_{\mathsf{O}}) \text{ to } (D^* \times A_{\mathsf{P}}) : & (p, a, \gamma) & \xrightarrow{\mathsf{O}} & (p.d, q) & \text{iff } (q, d) \in \gamma \end{array}$$

and whose winning condition is given by

$$(\varepsilon, q_0) \cdot (\varepsilon, a_0, \gamma_0) \cdot (p_1, q_1) \cdot \ldots \cdot (p_n, q_n) \cdot (p_n, a_n, \gamma_n) \cdot \ldots \quad \in \mathcal{W} \qquad \text{iff} \quad (q_i)_{i \in \mathbb{N}} \in \Omega$$

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

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

Correspondence with usual Acceptance Games. Usual acceptance games model the evaluation of automata $\Sigma \vdash \mathcal{A}$ on input trees $t \in \mathbf{Tree}[\Sigma]$. They correspond to games of the form $\mathbf{1} \vdash \mathcal{G}(\mathcal{A}, \dot{t})$, where $\dot{t} \in \mathbf{Tree}[\mathbf{1}, \Sigma]$ is the tree morphism corresponding to $t \in \mathbf{Tree}[\Sigma]$.

Note that in these cases, A_{O} is of the form $\mathbf{1} \times \mathcal{P}(Q \times D) \simeq \mathcal{P}(Q \times D)$, so that the games $\mathbf{1} \vdash \mathcal{G}(\mathcal{A}, \dot{t})$ are isomorphic to the acceptance games of [19].

▶ **Definition 3.1.** Let $\Sigma \vdash \mathcal{A}$.

(i) $\mathcal{A}$ *accepts* the tree $t \in \mathbf{Tree}[\Sigma]$ if there is a strategy σ such that $\mathbf{1} \vdash \sigma \Vdash \mathcal{G}(\mathcal{A}, \dot{t})$.
(ii) Let $\mathcal{L}(\mathcal{A}) \subseteq \mathbf{Tree}[\Sigma]$, the *language* of $\mathcal{A}$, be the set of trees accepted by $\mathcal{A}$.

3.2 Synchronous Arrow Games

Consider games $\Sigma \vdash \mathcal{G}(\mathcal{A}, M)$ and $\Sigma \vdash \mathcal{G}(\mathcal{B}, N)$ with $\mathcal{A} = (Q_{\mathcal{A}}, q^\imath_{\mathcal{A}}, \delta_{\mathcal{A}}, \Omega_{\mathcal{A}})$ and $\mathcal{B} = (Q_{\mathcal{B}}, q^\imath_{\mathcal{B}}, \delta_{\mathcal{B}}, \Omega_{\mathcal{B}})$. Similarly as in Sect. 3.1 above, write

$$A_{\mathsf{P}} := Q_{\mathcal{A}} \qquad A_{\mathsf{O}} := \Sigma \times \mathcal{P}(Q_{\mathcal{A}} \times D) \qquad B_{\mathsf{P}} := Q_{\mathcal{B}} \qquad B_{\mathsf{O}} := \Sigma \times \mathcal{P}(Q_{\mathcal{B}} \times D)$$

We define the *synchronous arrow game*

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

as the negative game $(V, E, *, \lambda, \xi, \mathcal{W})$ whose positions are given by

$$V \quad := \quad (D^* \times A_{\mathsf{P}}) \times (D^* \times B_{\mathsf{P}}) \; + \; (D^* \times A_{\mathsf{O}}) \times (D^* \times B_{\mathsf{P}}) \; + \; (D^* \times A_{\mathsf{O}}) \times (D^* \times B_{\mathsf{O}})$$

whose polarized root is $* := ((\varepsilon, q^\imath_{\mathcal{A}}), (\varepsilon, q^\imath_{\mathcal{B}}))$ with $\xi(*) := \mathsf{O}$, whole polarized edges (E, λ) are given in Table 1, and whose winning condition is given by

$$\begin{array}{l} ((\varepsilon, q^0_{\mathcal{A}}) \; , \; (\varepsilon, q^0_{\mathcal{B}})) \;\cdot\; \ldots \;\cdot\; ((\varepsilon, q^n_{\mathcal{A}}) \; , \; (\varepsilon, q^n_{\mathcal{B}})) \;\cdot\; \ldots \quad \in \mathcal{W} \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{iff} \quad \left((q^i_{\mathcal{A}})_{i \in \mathbb{N}} \in \Omega_{\mathcal{A}} \implies (q^i_{\mathcal{B}})_{i \in \mathbb{N}} \in \Omega_{\mathcal{B}}\right) \end{array}$$

Note that P-plays end in positions of the form

$$\begin{array}{lccc} & ((p, q_{\mathcal{A}}) \; , \; (p, q_{\mathcal{B}})) & \in & (D^* \times A_{\mathsf{P}}) \; \times \; (D^* \times B_{\mathsf{P}}) \\ \text{and} & ((p, a, \gamma_{\mathcal{A}}) \; , \; (p, a, \gamma_{\mathcal{B}})) & \in & (D^* \times A_{\mathsf{O}}) \; \times \; (D^* \times B_{\mathsf{O}}) \end{array}$$

λ	$\mathcal{G}(\mathcal{A}, M)$	$\multimap\!\!\circledast$	$\mathcal{G}(\mathcal{B}, N)$	
	$((p, q_{\mathcal{A}})$	,	$(p, q_{\mathcal{B}}))$	
O	$\downarrow$			
	$((p, a, \gamma_{\mathcal{A}})$	,	$(p, q_{\mathcal{B}}))$	if $\gamma_{\mathcal{A}} \in \delta_{\mathcal{A}}(q_{\mathcal{A}}, M(p)(a))$
P			$\downarrow$	
	$((p, a, \gamma_{\mathcal{A}})$	,	$(p, a, \gamma_{\mathcal{B}}))$	if $\gamma_{\mathcal{B}} \in \delta_{\mathcal{B}}(q_{\mathcal{B}}, N(p)(a))$
O			$\downarrow$	
	$((p, a, \gamma_{\mathcal{A}})$	,	$(p.d, q'_{\mathcal{B}}))$	if $(q'_{\mathcal{B}}, d) \in \gamma_{\mathcal{B}}$
P	$\downarrow$			
	$((p.d, q'_{\mathcal{A}})$	,	$(p.d, q'_{\mathcal{B}}))$	if $(q'_{\mathcal{A}}, d) \in \gamma_{\mathcal{A}}$

Figure 1 Moves of $\mathcal{G}(\mathcal{A}, M) \multimap\!\!\circledast \mathcal{G}(\mathcal{B}, N)$

Each of these position is of homogeneous type, and moreover in each case the D^* and Σ components coincide. On the other hand, O-plays end in positions of the form

$$\begin{array}{lllll} & ((p, a, \gamma_{\mathcal{A}}) \ , \ (p, q_{\mathcal{A}})) & \in & (D^* \times A_{\mathsf{O}}) \times (D^* \times B_{\mathsf{P}}) \\ \text{and} & ((p, a, \gamma_{\mathcal{A}}) \ , \ (p \cdot d, q_{\mathcal{B}})) & \in & (D^* \times A_{\mathsf{O}}) \times (D^* \times B_{\mathsf{P}}) \end{array}$$

Each of these intermediate position is of heterogeneous type, and in the second one, the D^* components do not coincide.

We write $\Sigma \vdash \sigma : \mathcal{G}(\mathcal{A}, M) \multimap\!\!\circledast \mathcal{G}(\mathcal{B}, N)$ if σ is a P-strategy on $\mathcal{G}(\mathcal{A}, M) \multimap\!\!\circledast \mathcal{G}(\mathcal{B}, N)$, and $\Sigma \vdash \sigma \Vdash \mathcal{G}(\mathcal{A}, M) \multimap\!\!\circledast \mathcal{G}(\mathcal{B}, N)$ if σ is moreover winning. Finally, we write

$$\Sigma \Vdash \mathcal{G}(\mathcal{A}, M) \multimap\!\!\circledast \mathcal{G}(\mathcal{B}, N)$$

if there is a winning P-strategy σ on $\mathcal{G}(\mathcal{A}, M) \multimap\!\!\circledast \mathcal{G}(\mathcal{B}, N)$.

▶ Remark. Recall that if $\Omega_{\mathcal{A}}$ and $\Omega_{\mathcal{B}}$ are Borel sets, then $\mathcal{W}$ is a Borel set and by Martin's Theorem [12], either P or O has a winning strategy. Moreover, if the automata $\mathcal{A}$ and $\mathcal{B}$ are regular (in the sense of Sect. 2), then $\mathcal{W}$ is an ω-regular language. If in addition the trees M and N are regular (in the usual sense), then the game is equivalent to a finite regular game. By Büchi-Landweber Theorem, the existence of a winning strategy for a given player is decidable, and the winning player has *finite state* winning strategies (see e.g. [18]).

3.3 Characterization of the Synchronous Arrow Games

We now give a characterization of synchronous arrow games in traditional games semantics. Our characterization involve relations in slices categories $\mathbf{Set}/J$, that will give rise to a strong analogy between our fibrations $\mathsf{game}^{(\mathrm{W})}$ and $\mathsf{aut}^{(\mathrm{W})}$ and substitution (a.k.a *change-of-base*) in the codomain fibration $\mathsf{cod} : \mathbf{Set}^{\to} \to \mathbf{Set}$.

Simple Games. Recall the usual notion of *simple games* (see e.g. [1, 7]). Simple games are usually negative, but given positive games A and B, their *negative* linear arrow $A \multimap B$ can still be defined. Moreover, simple games, with linear arrows $A \multimap B$ between games A and B of the same polarity, form a category that we write $\mathbf{SGG}$. When equipped with winning conditions, winning strategies compose, giving rise to a category that we write $\mathbf{SGG}^{\mathrm{W}}$.

A P-strategy $\Sigma \vdash \sigma : \mathcal{G}(\mathcal{A}, M) \multimap\!\!\circledast \mathcal{G}(\mathcal{B}, N)$ is a morphism of $\mathbf{SGG}$ from the substituted acceptance game $\mathcal{G}(\mathcal{A}, M)$ to the substituted acceptance game $\mathcal{G}(\mathcal{B}, N)$. If σ is moreover winning, then it is a morphism of $\mathbf{SGG}^{\mathrm{W}}$.

The Hyland & Schalk Functor. Hyland & Schalk have presented in [9] a faithful functor, that we denote HS, from simple games to the category **Rel** of sets and relations. This functor can easily be extended to a functor $\mathrm{HS} : \mathbf{SGG}^{(\mathrm{W})} \to \mathbf{Rel}$.

Given a play $s \in \wp(A \multimap B)$ we let $s{\upharpoonright}A \in \wp(A)$ be its projection on A and similarly for B,[2] so that $\mathrm{HS}(s) := (s{\upharpoonright}A, s{\upharpoonright}B)$. Given a P-strategy $\sigma : A \multimap B$ we have $\sigma \subseteq \wp^{\mathsf{P}}(A \multimap B)$ and thus

$$\mathrm{HS}(\sigma) \quad := \quad \{\mathrm{HS}(s) \mid s \in \sigma\} \quad \subseteq \quad \wp(A) \times \wp(B)$$

We write $\wp_\Sigma(\mathcal{A}, M)$ for the plays of the substituted acceptance game $\Sigma \vdash \mathcal{G}(\mathcal{A}, M)$. Given $\Sigma \vdash \sigma : \mathcal{G}(\mathcal{A}, M) \mathrel{-\!\!\circledast} \mathcal{G}(\mathcal{B}, N)$, we thus have

$$\mathrm{HS}(\sigma) \quad \subseteq \quad \wp_\Sigma(\mathcal{A}, M) \times \wp_\Sigma(\mathcal{B}, N)$$

Synchronous Relations. We will now see that P-strategies on a synchronous arrow game can be seen as relations in slice categories $\mathbf{Set}/J$. We call such relations *synchronous*.

Given a set J, define the category $\mathbf{Rel}(\mathbf{Set}/J)$ as follows:

Objects are indexed sets $A \xrightarrow{g} J$, written simply A when g is understood from the context.

Morphisms from $A \xrightarrow{g} J$ to $B \xrightarrow{h} J$ are given by relations $\mathring{R} : A \mathrel{-\!\!\!+\!\!\!\rightarrow} B$ such that the following commutes:

$$\begin{array}{ccccc} & & \mathring{R} & & \\ & \swarrow^{\pi_1} & & \searrow^{\pi_2} & \\ A & & & & B \\ & \searrow_{g} & & \swarrow_{h} & \\ & & J & & \end{array}$$

Traces. For the synchronous arrow games, synchronization is performed using the following notion of *trace*. Given $\Gamma \vdash \mathcal{A}$ and $M \in \mathbf{Tree}[\Gamma, \Sigma]$, define

$$\mathrm{tr} \quad : \quad \wp_\Sigma(\mathcal{A}, M) \quad \longrightarrow \quad (D + \Sigma)^*$$

inductively as follows

$$\mathrm{tr}(\varepsilon) \ := \ \varepsilon \qquad \mathrm{tr}(s \to (p, a, \gamma)) \ := \ \mathrm{tr}(s) \cdot a \qquad \mathrm{tr}(s \to (p \cdot d, q)) \ := \ \mathrm{tr}(s) \cdot d$$

The image of tr is the set $\mathrm{Tr}_\Sigma \ := \ (\Sigma \cdot D)^* + (\Sigma \cdot D)^* \cdot \Sigma$.

Characterization of the Synchronous Arrow. We can now characterize the synchronous arrow games. First, *via* the functor HS, synchronous strategies are synchronous relations.

▶ **Proposition 3.2.** *Strategies on the synchronous arrow game $\mathcal{G}(\mathcal{A}, M) \mathrel{-\!\!\circledast} \mathcal{G}(\mathcal{B}, N)$ are exactly the strategies $\sigma : \mathcal{G}(\mathcal{A}, M) \multimap \mathcal{G}(\mathcal{B}, N)$ such that*

$$\begin{array}{ccc} \mathrm{HS}(\sigma) & \longrightarrow & \wp_\Sigma(\mathcal{B}, N) \\ \downarrow & & \downarrow{\scriptstyle \mathrm{tr}} \\ \wp_\Sigma(\mathcal{A}, M) & \xrightarrow[\mathrm{tr}]{} & \mathrm{Tr}_\Sigma \end{array} \qquad (2)$$

Second, plays on the synchronous arrow can be obtained in a canonical way from plays on its components.

---

[2] We write $\wp(A)$ for the set of plays on A, and $\wp^{\mathsf{P}}(A)$ for the set of P-plays.

▶ **Proposition 3.3.** *Let* $\Sigma \vdash \mathcal{G}(\mathcal{A}, M)$ *and* $\Sigma \vdash \mathcal{G}(\mathcal{B}, N)$*. The following is a pullback in* $\mathbf{Set}$*:*

$$\begin{array}{ccc}
\wp^{\mathsf{P}}_{\Sigma}(\mathcal{G}(\mathcal{A}, M) \mathrel{-\!\!\circledast} \mathcal{G}(\mathcal{B}, N)) & \xrightarrow{(-)\restriction\mathcal{G}(\mathcal{B},N)} & \wp_{\Sigma}(\mathcal{B}, N) \\
{\scriptstyle (-)\restriction\mathcal{G}(\mathcal{A},M)}\big\downarrow \quad \lrcorner & & \big\downarrow{\scriptstyle \mathrm{tr}} \\
\wp_{\Sigma}(\mathcal{A}, M) & \xrightarrow[\mathrm{tr}]{} & \mathrm{Tr}_{\Sigma}
\end{array}$$

We write $\mathrm{tr}^{-\circledast}$ for any of two equal maps

$$\mathrm{tr} \circ (-)\restriction\mathcal{G}(\mathcal{A}, M)\ ,\ \mathrm{tr} \circ (-)\restriction\mathcal{G}(\mathcal{B}, N) \quad : \quad \wp^{\mathsf{P}}_{\Sigma}(\mathcal{G}(\mathcal{A}, M) \mathrel{-\!\!\circledast} \mathcal{G}(\mathcal{B}, N)) \quad \longrightarrow \quad \mathrm{Tr}_{\Sigma}$$

3.4 Categories of Substituted Acceptance Games and Automata

We now define our categories $\mathbf{SAG}^{(\mathrm{W})}_{\Sigma}$ of substituted acceptance games and their full subcategories $\mathbf{Aut}^{(\mathrm{W})}_{\Sigma}$ of tree automata. That they indeed form categories follows from the characterization Prop. 3.2, together with the fact that $\mathbf{Rel}(\mathbf{Set}/J)$ and $\mathbf{SGG}^{(\mathrm{W})}$ are categories, and the fact that the identity strategies $\mathrm{id} : \mathcal{G}(\mathcal{A}, M) \multimap \mathcal{G}(\mathcal{B}, N)$ are synchronous.

Objects of $\mathbf{SAG}_{\Sigma}$ and $\mathbf{SAG}^{\mathrm{W}}_{\Sigma}$ are games $\Sigma \vdash \mathcal{G}(\mathcal{A}, M)$,
Morphisms of $\mathbf{SAG}_{\Sigma}$ are synchronous strategies $\Sigma \vdash \sigma : \mathcal{G}(\mathcal{A}, M) \mathrel{-\!\!\circledast} \mathcal{G}(\mathcal{B}, N)$,
Morphisms of $\mathbf{SAG}^{\mathrm{W}}_{\Sigma}$ are synchronous winning strategies $\Sigma \vdash \sigma \Vdash \mathcal{G}(\mathcal{A}, M) \mathrel{-\!\!\circledast} \mathcal{G}(\mathcal{B}, N)$.

Objects of $\mathbf{Aut}_{\Sigma}$ and $\mathbf{Aut}^{\mathrm{W}}_{\Sigma}$ are automata $\Sigma \vdash \mathcal{A}$,
Morphisms of $\mathbf{Aut}_{\Sigma}$ are synchronous strategies $\Sigma \vdash \sigma : \mathcal{G}(\mathcal{A}, \mathrm{Id}_{\Sigma}) \mathrel{-\!\!\circledast} \mathcal{G}(\mathcal{B}, \mathrm{Id}_{\Sigma})$,
Morphisms of $\mathbf{Aut}^{\mathrm{W}}_{\Sigma}$ are synchronous winning strategies $\Sigma \vdash \sigma \Vdash \mathcal{G}(\mathcal{A}, \mathrm{Id}_{\Sigma}) \mathrel{-\!\!\circledast} \mathcal{G}(\mathcal{B}, \mathrm{Id}_{\Sigma})$.

A Lifting Property. Among the useful consequences of Prop. 3.3, we state the following lifting property.

▶ **Proposition 3.4.** *Consider* $\Sigma \vdash \mathcal{G}(\mathcal{A}, M)$ *and* $\Sigma \vdash \mathcal{G}(\mathcal{B}, N)$*. Assume that, in* $\mathbf{Rel}(\mathbf{Set}/\mathrm{Tr}_{\Sigma})$ *we have an isomorphism* $\mathring{R} : (\wp_{\Sigma}(\mathcal{A}, M) \xrightarrow{\mathrm{tr}} \mathrm{Tr}_{\Sigma}) \mathrel{+\!\!\!\rightarrow}_{/\mathrm{Tr}_{\Sigma}} (\wp_{\Sigma}(\mathcal{B}, N) \xrightarrow{\mathrm{tr}} \mathrm{Tr}_{\Sigma})$*.*
There is a (unique, total) isomorphism $\sigma : \mathcal{G}(\mathcal{A}, M) \longrightarrow_{\mathbf{SAG}_{\Sigma}} \mathcal{G}(\mathcal{B}, N)$ *s.t.* $\mathrm{HS}(\sigma) = R$*.*

In general we can not ask σ to be winning, and in particular to be a morphism of $\mathbf{SAG}^{\mathrm{W}}_{\Sigma}$.

4 Fibrations of Acceptance Games and Automata

A tree morphism $L \in \mathbf{Tree}[\Sigma, \Gamma]$ defines a map L^* from the objects of $\mathbf{SAG}_{\Gamma}$ to the objects of $\mathbf{SAG}_{\Sigma}$: we let $L^*(\Gamma \vdash \mathcal{G}(\mathcal{A}, M)) := \Sigma \vdash \mathcal{G}(\mathcal{A}, M \circ L)$.

In this Section, we show that L^* extends to functors $L^* : \mathbf{SAG}^{(\mathrm{W})}_{\Gamma} \longrightarrow \mathbf{SAG}^{(\mathrm{W})}_{\Sigma}$ and that the operation $(-)^*$ is itself functorial and thus leads to split indexed categories $(-)^* : \mathbf{Tree}^{\mathrm{op}} \longrightarrow \mathbf{Cat}$. By applying Groethendieck completion, we obtain our split fibrations of acceptance games $\mathsf{game}^{(\mathrm{W})} : \mathbf{SAG}^{(\mathrm{W})} \longrightarrow \mathbf{Tree}$.

On the other hand, by restricting substitution to tree morphisms generated by alphabet morphisms $\beta \in \mathbf{Alph}[\Sigma, \Gamma]$, we obtain functors $\beta^* : \mathbf{Aut}^{(\mathrm{W})}_{\Gamma} \longrightarrow \mathbf{Aut}^{(\mathrm{W})}_{\Sigma}$ giving rise to split fibrations of tree automata $\mathsf{aut}^{(\mathrm{W})} : \mathbf{Aut}^{(\mathrm{W})} \longrightarrow \mathbf{Alph}$.

Our substitution functors L^* are build in strong analogy with change-of-base functors $\mathbf{Set}/\mathrm{Tr}_{\Gamma} \to \mathbf{Set}/\mathrm{Tr}_{\Sigma}$ of the codomain fibration $\mathsf{cod} : \mathbf{Set}^{\to} \to \mathbf{Set}$. We refer to [10] for basic material about fibrations.

4.1 Substitution Functors

Change-of-Base in $\mathbf{Set}^{\to}$. A morphism $L \in \mathbf{Tree}[\Sigma, \Gamma]$ induces a map $\mathrm{Tr}(L) : \mathrm{Tr}_\Sigma \longrightarrow \mathrm{Tr}_\Gamma$ inductively defined as follows (where $(-)_D$ is the obvious projection $\mathrm{Tr}_\Sigma \to D^*$):

$$\mathrm{Tr}(L)(\varepsilon) \ := \ \varepsilon \qquad \mathrm{Tr}(L)(w \cdot a) \ := \ \mathrm{Tr}(L)(w) \cdot L(w_D)(a) \qquad \mathrm{Tr}(L)(w \cdot d) \ := \ \mathrm{Tr}(L)(w) \cdot d$$

The map $\mathrm{Tr}(L)$ gives rise to the usual change-of-base functor $L^\bullet : \mathbf{Set}/\mathrm{Tr}_\Gamma \to \mathbf{Set}/\mathrm{Tr}_\Sigma$, defined using chosen pullbacks in $\mathbf{Set}$:

$$\begin{array}{ccc} L^\bullet(\wp_\Gamma(\mathcal{A}, M)) & \longrightarrow & \wp_\Gamma(\mathcal{A}, M) \\ {\scriptstyle L^\bullet(\mathrm{tr})}\downarrow \quad \lrcorner & & \downarrow{\scriptstyle \mathrm{tr}} \\ \mathrm{Tr}_\Sigma & \xrightarrow{\ \mathrm{Tr}(L)\ } & \mathrm{Tr}_\Gamma \end{array}$$

Substitution on Plays. The action of the substitution L^* on plays can be described, similarly as the action of $L^\bullet$ on objects of $\mathbf{Set}/\mathrm{Tr}_\Gamma$, by a pullback property.

Consider $\Gamma \vdash \mathcal{G}(\mathcal{A}, M)$, so that $\Sigma \vdash \mathcal{G}(\mathcal{A}, M \circ L)$. A position $(p, a, \gamma_\mathcal{A})$ of the game $\Sigma \vdash \mathcal{G}(\mathcal{A}, M \circ L)$ can be mapped to the position $(p, L(p)(a), \gamma_\mathcal{A})$ of the game $\Gamma \vdash \mathcal{G}(\mathcal{A}, M)$. Moreover, since $\delta_\mathcal{A}(q_\mathcal{A}, (M \circ L)(p)(a)) \ = \ \delta_\mathcal{A}(q_\mathcal{A}, M(p)(L(p)(a)))$, we have

$$(p, q_\mathcal{A}) \to (p, a, \gamma_\mathcal{A}) \qquad \text{if and only if} \qquad (p, q_\mathcal{A}) \to (p, L(p)(a), \gamma_\mathcal{A})$$

This gives a map

$$\wp(L) \quad : \quad \wp_\Sigma(\mathcal{A}, M \circ L) \quad \longrightarrow \quad \wp_\Gamma(\mathcal{A}, M)$$

If we are also given $\Gamma \vdash \mathcal{G}(\mathcal{B}, N)$, then we similarly obtain

$$\wp(L)_{-\!\circledast} \quad : \quad \wp_\Sigma(\mathcal{G}(\mathcal{A}, M \circ L) -\!\!\circledast\ \mathcal{G}(\mathcal{B}, N \circ L)) \quad \longrightarrow \quad \wp_\Gamma((\mathcal{G}(\mathcal{A}, M) -\!\!\circledast\ \mathcal{G}(\mathcal{B}, N))$$

These two maps are related *via* HS as expected: $\mathrm{HS} \circ \wp(L)_{-\!\circledast} = (\wp(L) \times \wp(L)) \circ \mathrm{HS}$. Moreover,

▶ **Proposition 4.1.** *We have, in* $\mathbf{Set}$*:*

$$\begin{array}{ccc} \wp_\Sigma(\mathcal{A}, M \circ L) & \xrightarrow{\ \wp(L)\ } & \wp_\Gamma(\mathcal{A}, M) \\ {\scriptstyle \mathrm{tr}}\downarrow \quad \lrcorner & & \downarrow{\scriptstyle \mathrm{tr}} \\ \mathrm{Tr}_\Sigma & \xrightarrow{\ \mathrm{Tr}(L)\ } & \mathrm{Tr}_\Gamma \end{array} \qquad \begin{array}{ccc} \wp^{\mathsf{P}}_\Sigma(\mathcal{G}(\mathcal{A}, M \circ L) -\!\!\circledast\ \mathcal{G}(\mathcal{B}, N \circ L)) & \xrightarrow{\ \wp(L)_{-\circledast}\ } & \wp^{\mathsf{P}}_\Gamma(\mathcal{G}(\mathcal{A}, M) -\!\!\circledast\ \mathcal{G}(\mathcal{B}, N)) \\ {\scriptstyle \mathrm{tr}^{-\circledast}}\downarrow \quad \lrcorner & & \downarrow{\scriptstyle \mathrm{tr}^{-\circledast}} \\ \mathrm{Tr}_\Sigma & \xrightarrow{\ \mathrm{Tr}(L)\ } & \mathrm{Tr}_\Gamma \end{array}$$

Substitution on Strategies. The action of L^* on strategies is defined using Prop. 4.1: Given $\Gamma \vdash \sigma : \mathcal{G}(\mathcal{A}, M) -\!\!\circledast\ \mathcal{G}(\mathcal{B}, N)$, so that $\sigma \subseteq \wp^{\mathsf{P}}_\Gamma(\mathcal{G}(\mathcal{A}, M) -\!\!\circledast\ \mathcal{G}(\mathcal{B}, N))$, we define

$$L^*(\sigma) \quad := \quad \wp(L)^{-1}_{-\circledast}(\sigma) \quad \subseteq \quad \wp^{\mathsf{P}}_\Sigma(\mathcal{G}(\mathcal{A}, M \circ L) -\!\!\circledast\ \mathcal{G}(\mathcal{B}, N \circ L))$$

▶ **Proposition 4.2.** $L^*(\sigma)$ *is a strategy. If moreover* σ *is winning, then* $L^*(\sigma)$ *is also winning.*

Functoriality of Substitution. Proposition 4.1 can be formulated by saying that the maps $\langle \mathrm{tr}, \wp(L)\rangle$ and $\langle \mathrm{tr}^{-\circledast}, \wp(L)_{-\circledast}\rangle$ are bijections, respectively:

$$\begin{array}{rcl} \wp_\Sigma(\mathcal{A}, M \circ L) & \xrightarrow{\simeq} & \mathrm{Tr}_\Sigma \times_{\mathrm{Tr}_\Sigma} \wp_\Gamma(\mathcal{A}, M) \\ \wp^{\mathsf{P}}_\Sigma(\mathcal{G}(\mathcal{A}, M \circ L) \multimap\!\!\circledast\ \mathcal{G}(\mathcal{B}, N \circ L)) & \xrightarrow{\simeq} & \mathrm{Tr}_\Sigma \times_{\mathrm{Tr}_\Sigma} \wp^{\mathsf{P}}_\Gamma(\mathcal{G}(\mathcal{A}, M) \multimap\!\!\circledast\ \mathcal{G}(\mathcal{B}, N)) \end{array}$$

These bijections are crucial to prove that

▶ **Proposition 4.3.** *L^* is a functor from $\mathbf{SAG}^{(\mathrm{W})}_\Gamma$ to $\mathbf{SAG}^{(\mathrm{W})}_\Sigma$.*

4.2 Fibrations of Acceptance Games

Consider now $L \in \mathbf{Tree}[\Sigma, \Gamma]$ and $K \in \mathbf{Tree}[\Gamma, \Delta]$. Since $\mathrm{Tr}(K \circ L) = \mathrm{Tr}(K) \circ \mathrm{Tr}(L)$ and $\wp(K \circ L)_{(-\circledast)} = \wp(K)_{(-\circledast)} \circ \wp(L)_{(-\circledast)}$ we immediately get

▶ **Proposition 4.4.** *The operations $(-)^* : \mathbf{Tree}^{\mathrm{op}} \to \mathbf{Cat}$, mapping Σ to $\mathbf{SAG}^{(\mathrm{W})}_\Sigma$, and mapping $L \in \mathbf{Tree}[\Sigma, \Gamma]$ to $L^* : \mathbf{SAG}^{(\mathrm{W})}_\Gamma \to \mathbf{SAG}^{(\mathrm{W})}_\Sigma$ are functors.*

By using Groethendieck completion (see e.g. [10, §1.10]), this gives us split fibrations of acceptance games $\mathsf{game}^{(\mathrm{W})} : \mathbf{SAG}^{(\mathrm{W})} \longrightarrow \mathbf{Tree}$ that we do not detail here by lack of space.

4.3 Fibrations of Automata

In order to obtain fibrations of automata, we restrict substitution to tree morphisms generated by alphabet morphisms $\beta \in \mathbf{Alph}[\Sigma, \Gamma]$. The crucial point is that these restricted substitutions can be internalized in automata.

Given $\Gamma \vdash \mathcal{A}$ with $\mathcal{A} = (Q, q^\imath, \delta, \Omega)$, and $\beta \in \mathbf{Alph}[\Sigma, \Gamma]$, define the automaton $\Sigma \vdash \mathcal{A}[\beta]$ as $\mathcal{A}[\beta] := (Q, q^\imath, \delta_\beta, \Omega)$ where $\delta_\beta(q, a) := \delta(q, \beta(a))$.

▶ **Proposition 4.5.** $\Sigma \vdash \mathcal{G}(\mathcal{A}[\beta], \mathrm{Id}_\Sigma) \;=\; \Sigma \vdash \mathcal{G}(\mathcal{A}, \beta)$.

It is easy to see that $(-)^*$ restricts to a functor from $\mathbf{Alph}^{\mathrm{op}}$ to $\mathbf{Cat}$, so that we get fibrations

$$\mathsf{aut}^{(\mathrm{W})} \quad : \quad \mathbf{Aut}^{(\mathrm{W})} \quad \longrightarrow \quad \mathbf{Alph}$$

5 Symmetric Monoidal Structure

We now consider a synchronous product of automata. When working on *complete* automata (to be defined in Sect. 5.1 below), it gives rise to split symmetric monoidal fibrations, in the sense of [17].

According to [17, Thm. 12.7], split symmetric monoidal fibrations can equivalently be obtained from split symmetric monoidal indexed categories. In our context, this means that the functors $(-)^*$ extend to

$$(-)^* \;:\; \mathbf{Tree}^{\mathrm{op}} \longrightarrow \mathbf{SymMonCat} \qquad (-)^* \;:\; \mathbf{Alph}^{\mathrm{op}} \longrightarrow \mathbf{SymMonCat}$$

where $\mathbf{SymMonCat}$ is the category of symmetric monoidal categories and strong monoidal functors. Hence, we equip our categories of (complete) acceptance games and automata with a symmetric monoidal structure. Substitution turns out to be *strict* symmetric monoidal.

We refer to [13] for background on symmetric monoidal categories.

5.1 Complete Tree Automata

An automaton $\mathcal{A}$ is *complete* if for every $(q, a) \in Q \times \Sigma$, the set $\delta(q, a)$ is not empty and moreover for every $\gamma \in \delta(q, a)$ and everydirection $d \in D$, we have $(q', d) \in \gamma$ for some $q' \in Q$.

Given an automaton $\mathcal{A} = (Q, q^\imath, \delta, \Omega)$ its *completion* is the automaton $\widehat{\mathcal{A}} := (\widehat{Q}, q^\imath, \widehat{\delta}, \widehat{\Omega})$ with states $\widehat{Q} := Q + \{\mathsf{true}, \mathsf{false}\}$, with acceptance condition $\widehat{\Omega} := \Omega + Q^* \cdot \mathsf{true} \cdot \widehat{Q}^\omega$, and with transition function $\widehat{\delta}$ defined as

$$\begin{array}{rcll} \widehat{\delta}(\mathsf{true}, q) & := & \{\{(\mathsf{true}, d) \mid d \in D\}\} & \widehat{\delta}(\mathsf{false}, q) \;:=\; \{\{(\mathsf{false}, d) \mid d \in D\}\} \\ \widehat{\delta}(q, a) & := & \{\{(\mathsf{false}, d) \mid d \in D\}\} & \text{if } q \in Q \text{ and } \delta(q, a) = \emptyset \\ \widehat{\delta}(q, a) & := & \{\widehat{\gamma} \mid \gamma \in \delta(q, a)\} & \text{otherwise} \end{array}$$

where, given $\gamma \in \delta(q, a)$, we let $\widehat{\gamma} := \gamma \cup \{(\mathsf{true}, d) \mid \text{there is no } q \in Q \text{ s.t. } (q, d) \in \gamma\}$.

▶ **Proposition 5.1.** $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\widehat{\mathcal{A}})$.

Restricting to complete automata gives rise to full subcategories $\widehat{\mathbf{SAG}}_\Sigma^{(\mathrm{W})}$ and $\widehat{\mathbf{Aut}}_\Sigma^{(\mathrm{W})}$ of resp. $\mathbf{SAG}_\Sigma^{(\mathrm{W})}$ and $\mathbf{Aut}_\Sigma^{(\mathrm{W})}$, and thus induces fibrations

$$\widehat{\mathsf{game}} \;:\; \widehat{\mathbf{SAG}}^{(\mathrm{W})} \longrightarrow \mathbf{Tree} \qquad\qquad \widehat{\mathsf{aut}} \;:\; \widehat{\mathbf{Aut}}^{(\mathrm{W})} \longrightarrow \mathbf{Alph}$$

5.2 The Synchronous Product

Assume given complete automata $\Sigma \vdash \mathcal{A}$ and $\Sigma \vdash \mathcal{B}$. Define $\Sigma \vdash \mathcal{A} \circledast \mathcal{B}$ as

$$\mathcal{A} \circledast \mathcal{B} \quad := \quad (Q_\mathcal{A} \times Q_\mathcal{B}, (q^\imath_\mathcal{A}, q^\imath_\mathcal{B}), \delta_{\mathcal{A}\circledast\mathcal{B}}, \Omega_{\mathcal{A}\circledast\mathcal{B}})$$

where $(q^n_\mathcal{A}, q^n_\mathcal{B})_{n\in\mathbb{N}} \in \Omega_{\mathcal{A}\circledast\mathcal{B}}$ iff $((q^n_\mathcal{A})_{n\in\mathbb{N}} \in \Omega_\mathcal{A}$ and $(q^n_\mathcal{B})_{n\in\mathbb{N}} \in \Omega_\mathcal{B})$, and where we let $\delta_{\mathcal{A}\circledast\mathcal{B}}((q_\mathcal{A}, q_\mathcal{B}), a)$ be the set of all the $\gamma_\mathcal{A} \circledast \gamma_\mathcal{B}$ for $\gamma_\mathcal{A} \in \delta_\mathcal{A}(q_\mathcal{A}, a)$ and $\gamma_\mathcal{B} \in \delta_\mathcal{B}(q_\mathcal{B}, a)$, with $\gamma_\mathcal{A} \circledast \gamma_\mathcal{B} := \{((q'_\mathcal{A}, q'_\mathcal{B}), d) \mid d \in D \text{ and } (q'_\mathcal{A}, d) \in \gamma_\mathcal{A} \text{ and } (q'_\mathcal{B}, d) \in \gamma_\mathcal{B}\}$.

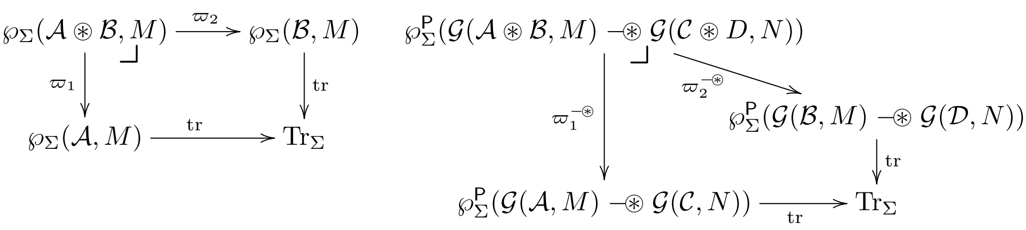
Note that since $\mathcal{A}$ and $\mathcal{B}$ are complete, each $\gamma_{\mathcal{A}\circledast\mathcal{B}} \in \delta_{\mathcal{A}\circledast\mathcal{B}}((q_\mathcal{A}, q_\mathcal{B}), a)$ uniquely decomposes as $\gamma_{\mathcal{A}\circledast\mathcal{B}} = \gamma_\mathcal{A} \circledast \gamma_\mathcal{B}$.

Action on Plays. The unique decomposition property of $\gamma_{\mathcal{A}\circledast\mathcal{B}}$ allows to define projections

$$\begin{array}{rcrcl} \varpi_i & : & \wp_\Sigma(\mathcal{A}_1 \circledast \mathcal{A}_2, M) & \longrightarrow & \wp_\Sigma(\mathcal{A}_i, M) \\ \varpi_i^{-\circledast} & : & \wp_\Sigma\left(\mathcal{G}(\mathcal{A}_1 \circledast \mathcal{B}_1, M) \multimap\!\!\circledast\; \mathcal{G}(\mathcal{A}_2 \circledast \mathcal{B}_2, N)\right) & \longrightarrow & \wp_\Sigma\left(\mathcal{G}(\mathcal{A}_i, M) \multimap\!\!\circledast\; \mathcal{G}(\mathcal{B}_i, N)\right) \end{array}$$

We write $\mathrm{SP} := \langle \varpi_1, \varpi_2 \rangle$ and $\mathrm{SP}_{-\circledast} := \langle \varpi_1^{-\circledast}, \varpi_2^{-\circledast} \rangle$.

▶ **Proposition 5.2.** *We have, in* **Set***:*



Action on Synchronous Games. The action of $\circledast$ on the objects of $\widehat{\mathbf{SAG}}_\Sigma^{(\mathrm{W})}$ is given by

$$(\Sigma \vdash \mathcal{G}(\mathcal{A}, M)) \circledast (\Sigma \vdash \mathcal{G}(\mathcal{B}, N)) \quad := \quad \Sigma \vdash \mathcal{G}(\mathcal{A}[\pi] \circledast \mathcal{B}[\pi'], \langle M, N \rangle)$$

where π and π' are suitable projections. For morphisms, let $\Sigma \vdash \sigma : \mathcal{G}(\mathcal{A}_0, M_0) \multimap_\circledast \mathcal{G}(\mathcal{A}_1, M_1)$ and $\Sigma \vdash \tau : \mathcal{G}(\mathcal{B}_0, N_0) \multimap_\circledast \mathcal{G}(\mathcal{B}_1, N_1)$. Then since $\Sigma \vdash \mathcal{G}(\mathcal{A}_i[\pi_i], \langle M_i, N_i \rangle) = \Sigma \vdash \mathcal{G}(\mathcal{A}_i, M_i)$ and $\Sigma \vdash \mathcal{G}(\mathcal{B}_i[\pi'_i], \langle M_i, N_i \rangle) = \Sigma \vdash \mathcal{G}(\mathcal{B}_i, N_i)$, thanks to Prop. 5.2 we can simply let $\sigma \circledast \tau := \mathrm{SP}^{-1}_{\multimap\circledast}(\sigma, \tau)$.

▶ **Proposition 5.3.** *The product* $\_ \circledast \_$ *gives functors* $\widehat{\mathbf{SAG}}_\Sigma^{(\mathrm{W})} \times \widehat{\mathbf{SAG}}_\Sigma^{(\mathrm{W})} \longrightarrow \widehat{\mathbf{SAG}}_\Sigma^{(\mathrm{W})}$.

5.3 Symmetric Monoidal Structure

Thanks to Prop. 5.2 and Prop. 3.4 the symmetric monoidal structure of $\circledast$ in $\widehat{\mathbf{SAG}}_\Sigma^{(\mathrm{W})}$ can be directly obtained from the symmetric monoidal structure of the tensorial product of $\mathbf{Rel}(\mathbf{Set}/\mathrm{Tr}_\Sigma)$.

Symmetric Monoidal Structure in $\mathbf{Rel}(\mathbf{Set}/J)$. We define a product $\otimes$ in $\mathbf{Rel}(\mathbf{Set}/J)$:

On Objects: for (A, g) and (B, h) objects in $\mathbf{Rel}(\mathbf{Set}/J)$ the product $A \otimes B$ is $A \times_J B$ with the corresponding map, that is

$$A \otimes B \quad := \quad \{(a, b) \in A \times B \mid g(a) = h(b)\} \quad \xrightarrow{g \circ \pi_1 = h \circ \pi_2} \quad J$$

On Morphisms: given $R \in \mathbf{Rel}(\mathbf{Set}/J)[A, C]$ and $P \in \mathbf{Rel}(\mathbf{Set}/J)[B, D]$, we define $R \otimes P \in \mathbf{Rel}(\mathbf{Set}/J)[A \otimes B, C \otimes D]$ as

$$R \otimes P \quad := \quad \{((a, b), (c, d)) \in (A \otimes B) \times (C \otimes D) \mid (a, c) \in R \text{ and } (b, d) \in P\}$$

For the unit, we *choose* some $\mathbf{I} = (\jmath : I \xrightarrow{\simeq} J)$. Note that $\jmath$ is required to be a bijection. The natural isomorphisms are given by:

$$\begin{array}{rcl}
\mathring{\alpha}_{A,B,C} & := & \{(((a, b), c) \ , \ (a, (b, c))) \mid g_A(a) = g_B(b) = g_C(c)\} \\
\mathring{\lambda}_A & := & \{((e, a) \ , \ a) \mid \jmath(e) = g_A(a)\} \\
\mathring{\rho}_A & := & \{((a, e) \ , \ a) \mid g_A(a) = \jmath(e)\} \\
\mathring{\gamma}_{A,B} & := & \{((a, b) \ , \ (b, a)) \mid g_A(a) = g_B(b)\}
\end{array}$$

We easily get:

▶ **Proposition 5.4.** *The category* $\mathbf{Rel}(\mathbf{Set}/J)$*, equipped with the above data, is symmetric monoidal.*

Unit Automata. The requirement that the monoidal unit $\jmath : I \to J$ of $\mathbf{Rel}(\mathbf{Set}/J)$ should be a bijection leads us to the following unit automata. We let $\mathcal{I} := (Q_\mathcal{I}, q_\mathcal{I}, \delta_\mathcal{I}, \Omega_\mathcal{I})$ where $Q_\mathcal{I} := \mathbf{1}$, $q_\mathcal{I} := \bullet$, $\Omega_\mathcal{I} = Q_\mathcal{I}^\omega$ and $\delta_\mathcal{I}(q_\mathcal{I}, a) := \{\{(q_\mathcal{I}, d) \mid d \in D\}\}$.

Note that since $\delta_\mathcal{I}$ is constant, we have $\Sigma \vdash \mathcal{G}(\mathcal{I}, M) = \Sigma \vdash \mathcal{G}(\mathcal{I}, \mathrm{Id})$. Moreover,

▶ **Proposition 5.5.** *Given* $M \in \mathbf{Tree}[\Sigma, \Gamma]$*, we have, in* $\mathbf{Set}$*, a bijection*

$$\mathrm{tr} \quad : \quad \wp_\Sigma(\mathcal{I}, M) \quad \xrightarrow{\simeq} \quad \mathrm{Tr}_\Sigma$$

Symmetric Monoidal Structure. Using Prop. 3.4, the structure isos of $\mathbf{Rel}(\mathbf{Set}/\mathrm{Tr}_\Sigma)$ can be lifted to $\widehat{\mathbf{SAG}}_\Sigma^{(\mathrm{W})}$ (winning is trivial). Moreover, the required equations (naturality and coherence) follows from Prop. 3.3, Prop 5.2, and the fact that $((\mathrm{SP} \times \mathrm{SP}) \circ \mathrm{HS})(\sigma \circledast \tau) = \mathrm{HS}(\sigma) \otimes \mathrm{HS}(\tau)$ (where composition on the left is in **Set**, and the expression denotes the actions of the resulting function on the set of plays $(\sigma \circledast \tau)$).

All the symmetric monoidal structure restricts from $\widehat{\mathbf{SAG}}_\Sigma^{(\mathrm{W})}$ to $\widehat{\mathbf{Aut}}_\Sigma^{(\mathrm{W})}$.

▶ **Proposition 5.6.** *The categories* $\widehat{\mathbf{SAG}}_\Sigma^{(\mathrm{W})}$ *and* $\widehat{\mathbf{Aut}}_\Sigma^{(\mathrm{W})}$ *equipped with the above data, are symmetric monoidal.*

5.4 Symmetric Monoidal Fibrations

In order to obtain symmetric monoidal fibrations, by [17, Thm. 12.7], it remains to check that substitution is strong monoidal. It is actually *strict* monoidal: it directly commutes with $\circledast$ and preserves the unit, as well as all the structure maps.

▶ **Proposition 5.7.**

(i) Given $L \in \mathbf{Tree}[\Sigma, \Gamma]$*, the functors* $L^* : \widehat{\mathbf{SAG}}_\Gamma^{(\mathrm{W})} \to \widehat{\mathbf{SAG}}_\Sigma^{(\mathrm{W})}$ *are strict monoidal.*
(ii) Given $\beta \in \mathbf{Alph}[\Sigma, \Gamma]$*, the functors* $\beta^* : \widehat{\mathbf{Aut}}_\Gamma^{(\mathrm{W})} \to \widehat{\mathbf{Aut}}_\Sigma^{(\mathrm{W})}$ *are strict monoidal.*

6 Correctness w.r.t. Language Operations

This Section gathers several properties stating the correctness of our constructions w.r.t. operations on recognized languages. We begin in Sect. 6.1 by properties on the symmetric monoidal structure, the most important one being that the synchronous arrow is *correct*, in the sense that $\Sigma \vdash \mathcal{A} \multimap\!\!\circledast\, \mathcal{B}$ implies $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$. Then, in Sect. 6.2, we discuss complementation of automata, and its relation with the synchronous arrow.

6.1 Correctness of the Symmetric Monoidal Structure

We begin by a formal correspondence between acceptance games and synchronous games of a specific form. This allows to show that the synchronous arrow is *correct*, in the sense that $\Sigma \vdash \mathcal{A} \multimap\!\!\circledast\, \mathcal{B}$ implies $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$. We then briefly discuss the correctness of the synchronous product w.r.t. language intersection.

▶ **Proposition 6.1.** *Given* $\Sigma \vdash \mathcal{A}$ *and* $t \in \mathbf{Tree}[\Sigma]$*, there is a bijection:*

$$\{\sigma \mid \mathbf{1} \vdash \sigma \Vdash \mathcal{G}(\mathcal{A}, \dot{t})\} \qquad \simeq \qquad \{\theta \mid \mathbf{1} \vdash \theta \Vdash \mathcal{G}(\mathcal{I}, \mathrm{Id}_{\mathbf{1}}) \multimap\!\!\circledast\, \mathcal{G}(\mathcal{A}, \dot{t})\}$$

▶ Remark. The above correspondence is only possible for acceptance games over **1**:
- In $\Sigma \vdash \sigma \Vdash \mathcal{G}(\mathcal{A}, M)$, σ is a positive P-strategy, hence chooses the input characters in Σ.
- In $\Sigma \vdash \theta \Vdash \mathcal{G}(\mathcal{I}_\Sigma, \mathrm{Id}_\Sigma) \multimap\!\!\circledast\, \mathcal{G}(\mathcal{A}, M)$, the strategy θ is a negative. It plays positively in $\Sigma \vdash \mathcal{G}(\mathcal{A}, M)$, but must follow the input characters chosen by O in $\Sigma \vdash \mathcal{G}(\mathcal{I}_\Sigma, \mathrm{Id}_\Sigma)$.

We now check that the arrow $\mathcal{G}(\mathcal{A}, M) \multimap\!\!\circledast\, \mathcal{G}(\mathcal{B}, N)$ is correct w.r.t. language inclusion:

▶ **Proposition 6.2** (Correctness of the Arrow)**.** *Assume given* $\Sigma \vdash \sigma \Vdash \mathcal{G}(\mathcal{A}, M) \multimap\!\!\circledast\, \mathcal{G}(\mathcal{B}, N)$.

(i) For all $t \in \mathbf{Tree}[\Sigma]$*, we have* $\dot{t}^*(\sigma) \Vdash \mathcal{G}(\mathcal{A}, M \circ \dot{t}) \multimap\!\!\circledast\, \mathcal{G}(\mathcal{B}, N \circ \dot{t})$.
(ii) If $\mathbf{1} \Vdash \mathcal{G}(\mathcal{A}, M \circ \dot{t})$ *then* $\mathbf{1} \Vdash \mathcal{G}(\mathcal{B}, N \circ \dot{t})$.

(iii) For all tree $t \in \mathbf{Tree}[\Sigma]$, if $M(t) \in \mathcal{L}(\mathcal{A})$ then $N(t) \in \mathcal{L}(\mathcal{B})$.

The converse property will be discussed in Sect. 7. We finally check that the synchronous product is correct.

▶ **Proposition 6.3.** $\mathcal{L}(\widehat{\mathcal{A}} \circledast \widehat{\mathcal{B}}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B})$.

6.2 Complementation and Falsity

Complementation. Given an automaton $\mathcal{A} = (Q, q^\imath, \delta, \Omega)$, following [19], we let its complement be $\sim\!\mathcal{A} := (Q, q^\imath, \delta_{\sim\mathcal{A}}, \Omega_{\sim\mathcal{A}})$, where $\Omega_{\sim\mathcal{A}} := Q^\omega \setminus \Omega$ and

$$\delta_{\sim\mathcal{A}}(q,a) \quad := \quad \{\gamma_\sim \in \mathcal{P}(Q \times D) \mid \forall \gamma \in \delta(q,a),\ \gamma_\sim \cap \gamma \neq \emptyset\}$$

The idea is that P on $\sim\!\mathcal{A}$ simulates O on $\mathcal{A}$, so that the correctness of $\sim\!\mathcal{A}$ relies on determinacy of acceptance games. In particular, thanks to Borel determinacy [12], we have:

▶ **Proposition 6.4** ([19])**.** *Given $\mathcal{A}$ with $\Omega_{\mathcal{A}}$ a Borel set, we have $\mathcal{L}(\sim\!\mathcal{A}) = \mathbf{Tree}[\Sigma] \setminus \mathcal{L}(\mathcal{A})$.*

Note that if $\mathcal{A}$ is complete, then $\sim\!\mathcal{A}$ is not necessarily complete, but $\delta_{\sim\mathcal{A}}$ is always not empty and so are the γ's in its image.

The Falsity Automaton $\bot$. We let $\bot := (Q_\bot, q_\bot, \delta_\bot, \Omega_\bot)$ where $Q_\bot := \mathbf{1}$, $q_\bot := \bullet$, $\Omega_\bot = \emptyset$ and $\delta_\bot(q_\bot, a) := \{\{(q_\bot, d)\} \mid d \in D\}$. Note that $\mathcal{I} = \sim\!\bot$. In particular, it is actually P who guides the evaluation of $\bot$, by choosing the tree directions.

▶ **Proposition 6.5.** *Let $\mathcal{A}$ and $\mathcal{B}$ be complete. Then $\Sigma \Vdash \mathcal{A} \circledast \mathcal{B} \multimap_{\circledast} \widehat{\bot}$ iff $\Sigma \Vdash \mathcal{A} \multimap_{\circledast} \widehat{\sim\mathcal{B}}$.*

▶ **Corollary 6.6.** *Let $\mathcal{A}$ be a complete automaton on Σ. Then $\mathbf{1} \Vdash \widehat{\sim\mathcal{A}}$ iff $\mathbf{1} \Vdash \mathcal{A} \multimap_{\circledast} \widehat{\bot}$.*

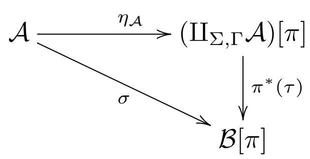
7 Projection and Fibred Simple Coproducts

We now check that automata can be equipped with existential quantifications in the fibered sense. Namely, given a projection $\pi \in \mathbf{Alph}[\Sigma \times \Gamma, \Sigma]$, the induced weakening functor $\pi^* : \widehat{\mathbf{Aut}}_\Sigma^{(\mathrm{W})} \to \widehat{\mathbf{Aut}}_{\Sigma\times\Gamma}^{(\mathrm{W})}$ has a left-adjoint $\amalg_{\Sigma,\Gamma}$, and moreover this structure is preserved by substitution, in the sense of the Beck-Chevalley condition (see e.g. [10]). This will lead to a (weak) completeness property of the synchronous arrow on *non-deterministic* automata, to be discussed below.

Recall from [11, Thm. IV.1.2.(ii)] than an adjunction $\amalg_{\Sigma,\Gamma} \dashv \pi^*$, with π^* a functor, is completely determined by the following data: To each object $\Sigma \times \Gamma \vdash \mathcal{A}$, an object $\Sigma \vdash \amalg_{\Sigma,\Gamma}\mathcal{A}$, and a map $\eta_{\mathcal{A}} : \Sigma \times \Gamma \vdash \mathcal{A} \longrightarrow \Sigma \times \Gamma \vdash (\amalg_{\Sigma,\Gamma}\mathcal{A})[\pi]$ satisfying the following universal lifting property:

For every
$\sigma \quad : \quad \Sigma \times \Gamma \vdash \mathcal{A} \quad \longrightarrow \quad \Sigma \times \Gamma \vdash \mathcal{B}[\pi]$
there is a unique
$\tau \quad : \quad \Sigma \vdash \amalg_{\Sigma,\Gamma}\mathcal{A} \quad \longrightarrow \quad \Sigma \vdash \mathcal{B}$

s.t.



(3)

In our context, the Beck-Chevalley condition amounts to the equalities

$$\Delta \vdash (\amalg_{\Sigma,\Gamma}\mathcal{A})[\beta] \;=\; \Delta \vdash \amalg_{\Delta,\Gamma}(\mathcal{A}[\beta \times \mathrm{Id}_\Gamma]) \qquad \eta_{\mathcal{A}[\beta\times\mathrm{Id}_\Gamma]} \;=\; (\beta \times \mathrm{Id}_\Gamma)^*(\eta_{\mathcal{A}}) \tag{4}$$

It turns out that the usual projection operation on automata (see e.g. [19]) is not functorial. Surprisingly, this is independent from whether automata are non-deterministic or not[3]. We devise a *lifted* projection operation, which indeed leads to a fibered existential quantification, and which is correct, on non-deterministic automata, w.r.t. the recognized languages.

The Lifted Projection. Consider $\Sigma \times \Gamma \vdash \mathcal{A}$ with $\mathcal{A} = (Q, q^\imath, \delta, \Omega)$. Define $\Sigma \vdash \amalg_{\Sigma,\Gamma}\mathcal{A}$ as $\amalg_{\Sigma,\Gamma}\mathcal{A} := (Q \times \Gamma + \{q^\imath\}, q^\imath, \delta_{\amalg\mathcal{A}}, \Omega_{\amalg\mathcal{A}})$ where

$$\begin{array}{rcl} \delta_{\amalg\mathcal{A}}(q^\imath, a) & := & \bigcup_{b\in\Gamma}\{\gamma^{+b} \mid \gamma \in \delta(q^\imath, (a, b))\} \\ \delta_{\amalg\mathcal{A}}((q, \_), a) & := & \bigcup_{b\in\Gamma}\{\gamma^{+b} \mid \gamma \in \delta(q, (a, b))\} \end{array}$$

and, given $\gamma \in \mathcal{P}(Q \times D)$ and $b \in \Gamma$, we let $\gamma^{+b} := \{((q^{+b}, d) \mid (q, d) \in \gamma\}$ with $q^{+b} := (q, b)$. For the acceptance condition, we let $q^\imath\cdot(q_0, b_0)\cdot\ldots\cdot(q_n, b_n)\cdot\ldots$ in $\Omega_{\amalg\mathcal{A}}$ iff $q^\imath\cdot q_0\cdot\ldots\cdot q_n\cdot\ldots \in \Omega$.

Action on Plays of The Lifted Projection. The action on plays of $\amalg_{\Sigma,\Gamma}$ is characterized by the map $\wp(\amalg) : \wp_{\Sigma\times\Gamma}(\mathcal{A}) \longrightarrow \wp_\Sigma(\amalg_{\Sigma,\Gamma}\mathcal{A})$ inductively defined as $\wp(\amalg)(\varepsilon, q^\imath) := (\varepsilon, q^\imath)$ and

$$\begin{array}{rcl} \wp(\amalg)((\varepsilon, q^\imath) \to^* (p, q) \to (p, (a, b), \gamma)) & := & \wp(\amalg)((\varepsilon, q^\imath) \to^* (p, q)) \to (p, a, \gamma^{+b}) \\ \wp(\amalg)((\varepsilon, q^\imath) \to^* (p, (a, b), \gamma) \to (p.d, q)) & := & \wp(\amalg)((\varepsilon, q^\imath) \to^* (p, (a, b), \gamma)) \to (p.d, q^{+b}) \end{array}$$

▶ **Proposition 7.1.** *If $\mathcal{A}$ is a complete automaton, then $\wp(\amalg)$ is a bijection.*

The Unit Maps $\eta_{(-)}$. Consider the injection $\iota_{\Sigma,\Gamma} : \wp_\Sigma(\amalg_{\Sigma,\Gamma}\mathcal{A}) \longrightarrow \wp_{\Sigma\times\Gamma}((\amalg_{\Sigma,\Gamma}\mathcal{A})[\pi])$ inductively defined as $\iota_{\Sigma,\Gamma}((\varepsilon, q^\imath_\mathcal{A})) := (\varepsilon, q^\imath_\mathcal{A})$ and $\iota_{\Sigma,\Gamma}(s \to (p, q^{+b})) := \iota_{\Sigma,\Gamma}(s) \to (p, q^{+b})$ and $\iota_{\Sigma,\Gamma}(s \to (p, a, \gamma^{+b})) := \iota_{\Sigma,\Gamma}(s) \to (p, (a, b), \gamma^{+b})$.

If $\Sigma \times \Gamma \vdash \mathcal{A}$ is complete, we let the unit $\eta_\mathcal{A}$ be the unique strategy of $\widehat{\mathbf{SAG}}^{\mathrm{W}}_{\Sigma\times\Gamma}$ such that $\mathrm{HS}(\eta_\mathcal{A}) = \{(t, \iota_{\Sigma,\Gamma} \circ \wp(\amalg)(t)) \mid t \in \wp_{\Sigma\times\Gamma}(\mathcal{A})\}$. We do not detail the B.-C. condition (4).

The Unique Lifting Property (3). Consider some $\Sigma \times \Gamma \vdash \sigma : \mathcal{A} \multimap\!\!\circledast\, \mathcal{B}[\pi]$ with $\mathcal{A}$ complete. We let τ be the unique strategy such that $\mathrm{HS}(\tau) = \{(\wp(\amalg)(s), \wp(\pi)(t)) \mid (s, t) \in \mathrm{HS}(\sigma)\}$. It is easy to see that τ is winning whenever σ is winning. Moreover

▶ **Lemma 7.2.** $\sigma = \pi^*(\tau) \circ \eta_\mathcal{A}$.

For the unicity part of the lifting property of $\eta_\mathcal{A}$, it is sufficient to check:

▶ **Lemma 7.3.** *If $\pi^*(\theta) \circ \eta_\mathcal{A} = \pi^*(\theta') \circ \eta_\mathcal{A}$ then $\theta = \theta'$.*

Non-Deterministic Tree Automata. An automaton $\mathcal{A}$ is *non-deterministic* if for every γ in the image of δ and every direction $d \in D$, there is at most one state q such that $(q, d) \in \gamma$.

▶ Remark. If $\mathcal{A}$ and $\mathcal{B}$ are non-deterministic, then so are $\mathcal{A} \circledast \mathcal{B}$ and $\amalg(\mathcal{A})$.

▶ **Proposition 7.4** ([4, 15, 19]). *For each regular automaton $\Sigma \vdash \mathcal{A}$ there is a complete non-deterministic automaton $\Sigma \vdash \mathrm{ND}(\mathcal{A})$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathrm{ND}(\mathcal{A}))$.*

▶ **Proposition 7.5.** *If $\Sigma \times \Gamma \vdash \mathcal{A}$ is non-deterministic and complete, then $\mathcal{L}(\amalg_{\Sigma,\Gamma}\mathcal{A}) = \pi_{\Sigma,\Gamma}(\mathcal{L}(\mathcal{A}))$ where $\pi_{\Sigma,\Gamma} \in \mathbf{Alph}[\Sigma \times \Gamma, \Sigma]$ is the first projection.*

▶ **Proposition 7.6.** *Consider complete regular automata $\Sigma \vdash \mathcal{A}$ and $\Sigma \vdash \mathcal{B}$.*
If $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$ then $\Sigma \Vdash \mathrm{ND}(\mathcal{A}) \multimap\!\!\circledast\, \widehat{\sim\mathcal{C}}$ with $\mathcal{C} := \mathrm{ND}(\sim\mathcal{B})$.

[3] It is well-known that the projection operation is correct w.r.t. the recognized languages only on *non-deterministic automata*.

8 Conclusion

We presented monoidal fibrations of tree automata and acceptance games, in which the fibre categories are based on a synchronous restriction of linear simple games.

For technical simplicity, we did not yet consider monoidal closure, but strongly expect that it holds. One of the main question is whether suitable restrictions of these categories are Cartesian closed, so as to interpret proofs from intuitionistic variants of MSO. Among other questions are the status of non-determinization (*i.e.* whether it can be made functorial, or even co-monadic), as well as relation with the Dialectica interpretation (in the vein of e.g. [8]). Our result of weak completeness (Prop. 7.6) suggests strong connections with the notion of *guidable* non-deterministic automata of [2]. On a similar vein, connections with *game automata* [3, 5] might be relevant to investigate.

Acknowledgments. This work benefited from numerous discussions with Pierre Clairambault and Thomas Colcombet.

References

1. 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.
2. 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.
3. J. Duparc, F. Facchini, and F. Murlak. Definable Operations On Weakly Recognizable Sets of Trees. In *FSTTCS*, volume 13 of *LIPIcs*, pages 363–374. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.
4. E. A. Emerson and C. S. Jutla. Tree Automata, Mu-Calculus and Determinacy (Extended Abstract). In *FOCS*, pages 368–377. IEEE Computer Society, 1991.
5. A. Facchini, F. Murlak, and M. Skrzypczak. Rabin-Mostowski Index Problem: A Step beyond Deterministic Automata. In *LICS*, pages 499–508. IEEE Computer Society, 2013.
6. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002.
7. 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.
8. J. M. E. Hyland. Proof theory in the abstract. *Ann. Pure Appl. Logic*, 114(1-3):43–78, 2002.
9. J. M. E. Hyland and A. Schalk. Abstract Games for Linear Logic. *Electr. Notes Theor. Comput. Sci.*, 29:127–150, 1999.
10. B. Jacobs. *Categorical Logic and Type Theory*. Studies in logic and the foundations of mathematics. Elsevier, 2001.
11. S. Mac Lane. *Categories for the Working Mathematician*. Springer, 2 edition, 1998.
12. D. A. Martin. Borel Determinacy. *The Annals of Mathematics, Second Series*, 102(2):363–371, 1975.
13. P.-A. Melliès. Categorical semantics of linear logic. In *Interactive models of computation and program behaviour*, volume 27 of *Panoramas et Synthèses*. SMF, 2009.
14. D. E. Muller and P. E. Schupp. Alternating Automata on Infinite Trees. *Theor. Comput. Sci.*, 54:267–276, 1987.

15 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.

16 C. Riba. Fibrations of Tree Automata. Available at http://perso.ens-lyon.fr/colin.riba/papers/fibaut.pdf, 2015.

17 M. Shulman. Framed bicategories and monoidal fibrations. *Theory and Applications of Categories*, 20(18):650–738, 2008.

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

19 I. Walukiewicz. Monadic second-order logic on tree-like structures. *Theor. Comput. Sci.*, 275(1-2):311–346, 2002.

A Simple Graph Games

We work on *simple graph games with winning*, of the form $G = (V, E, *, \lambda, \xi, \mathcal{W})$. They are played by *Opponent* (O) and *Proponent* (P) on the graph with vertices in V, edges in E, root $*$, edge labeling $\lambda : E \to \{\mathsf{O}, \mathsf{P}\}$, polarity $\xi : \{*\} \to \{\mathsf{O}, \mathsf{P}\}$ and winning condition $\mathcal{W} \subseteq V^\omega$. Vertices are game *positions*, while edges are *moves*: Opponent plays O-labeled moves and Proponent plays P-labeled moves. We write $v \to w$ if $(v, w) \in E$.

We assume that games are *alternating*, in the sense that $u \to v \to w$ implies $\lambda(u \to v) \neq \lambda(v \to w)$, and *polarized* in the sense that $\lambda(u \to v) = \lambda(u \to w)$ for all coinitial edges $u \to v$, $u \to w$, and moreover $\lambda(* \to u) = \xi(*)$ for all $* \to u$. A game is *positive* if $\xi(*) = \mathsf{P}$ and *negative* otherwise. A *play* is a finite path starting from the root $*$. It is a P-*play* (resp. an O-*play*) if it is either empty or ends with a P-move (resp. an O-move). A P-*strategy* is a non-empty set σ of P-plays which is

P-prefix-closed: if $s \to^* v \in \sigma$ and s is a P-play then $s \in \sigma$, and

P-deterministic: if $s \to w \in \sigma$ and $s \to w' \in \sigma$ then $w = w'$.

Consider a P-strategy σ and an O-play s. We say that s is an O-*interrogation* of σ if either $s = *$ and G is a positive game, or if $s = t \to u$ for some P-play $t \in \sigma$. We say that σ is *total* if for every O-interrogation s of σ, we have $s \to v \in \sigma$ for some v. A P-strategy σ is *winning* if it is total and moreover, for all infinite path $\pi \in V^\omega$, we have $\pi \in \mathcal{W}$ whenever $\pi(0) \to \ldots \to \pi(n) \in \sigma$ for infinitely many $n \in \mathbb{N}$.