Probabilistic Dynamic Input Output Automata

Pierre Civit

Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, F-75005 Paris, France
pierre.civit@lip6.fr

Maria Potop-Butucaru

Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, F-75005 Paris, France
maria.potop-butucaru@lip6.fr

Abstract

We present *probabilistic dynamic I/O automata*, a framework to model dynamic probabilistic systems. Our work extends *dynamic I/O Automata* formalism [1] to probabilistic setting. The original dynamic I/O Automata formalism included operators for parallel composition, action hiding, action renaming, automaton creation, and behavioral sub-typing by means of trace inclusion. They can model mobility by using signature modification. They are also hierarchical: a dynamically changing system of interacting automata is itself modeled as a single automaton. Our work extends to probabilistic settings all these features. Furthermore, we prove necessary and sufficient conditions to obtain the implementation monotonicity with respect to automata creation and destruction. Our work lays down the premises for extending *composable secure-emulation* [3] to dynamic settings, an important tool towards the formal verification of protocols combining probabilistic distributed systems and cryptography in dynamic settings (e.g. blockchains, secure distributed computation, cybersecure distributed protocols etc).

2012 ACM Subject Classification C.2.4 Distributed Systems

Keywords and phrases distributed dynamic systems, probabilistic automata, foundations

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Distributed computing area faces today important challenges coming from modern applications such as cryptocurrencies and blockchains which have a tremendous impact in our society. Blockchains are an evolved form of the distributed computing concept of replicated state machine, in which multiple agents see the evolution of a state machine in a consistent form. At the core of both mechanisms there are distributed computing fundamental elements (e.g. communication primitives and semantics, consensus algorithms, and consistency models) and also sophisticated cryptographic tools. Recently, [5] stated that despite the tremendous interest about blockchains and distributed ledgers, no formal abstraction of these objects has been proposed. In particular it was stated that there is a need for the formalization of the distributed systems that are at the heart of most cryptocurrency implementations, and leverage the decades of experience in the distributed computing community in formal specification when designing and proving various properties of such systems. Therefore, an extremely important aspect of blockchain foundations is a proper model for the entities involved and their potential behavior. The formalisation of blockchain area has to combine models of underlying distributed and cryptographic building blocks under the same hood.

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The formalisation of distributed systems has been pioneered by Lynch and Tuttle [6]. They proposed the formalism of *Input/Output Automata* to model deterministic distributed system. Later, this formalism is extended with Markov decision processes [7] to give *Probabilistic Input/Output Automata* [9] in order to model randomized distributed systems. In this model each process in the system is a automaton with probabilistic transitions. The probabilistic protocol is the parallel composition of the automata modeling each participant. This framework has been further extended in [2] to *task-structured probabilistic Input/Output automata* specifically designed for the analysis of cryptographic protocols. Task-structured probabilistic Input/Output automata are Probabilistic Input/Output automata extended with tasks structures that are equivalence classes on the set of actions. They define the parallel composition for this type of automata. Inspired by the literature in security area they also define the notion of implementation. Informally, the implementation of a Task-structured probabilistic Input/Output automata should look "similar" to the specification whatever the external environment of execution. Furthermore, they provide compositional results for the implementation relation. Even thought the formalism proposed in [2] has been already used in the verification of various cryptographic protocols this formalism does not capture the dynamicity in blockchains systems such as Bitcoin or Ethereum where the set of participants dynamically changes. Moreover, this formalism does not cover blockchain systems where subchains can be created or destroyed at run time [8].

Interestingly, the modelisation of dynamic behavior in distributed systems is an issue that has been addressed even before the born of blockchain systems. The increase of dynamic behavior in various distributed applications such as mobile agents and robots motivated the *Dynamic Input Output Automata* formalism introduced in [1]. This formalisms extends the *Input/Output Automata* formalism with the ability to change their signature dynamically (i.e. the set of actions in which the automaton can participate) and to create other I/O automata or destroy existing I/O automata. The formalism introduced in [1] does not cover the case of probabilistic distributed systems and therefore cannot be used in the verification of blockchains such as Algorand [4].

Our contribution. In order to cope with dynamicity and probabilistic nature of blockchain systems we propose an extension of the formalisms introduced in [2] and [1]. Our extension use a refined definition of probabilistic configuration automata in order to cope with dynamic actions. The main result of our formalism is as follows: the implementation of probabilistic configuration automata is monotonic to automata creation and destruction. Our work is an intermediate step before defining composable secure-emulation [3] in dynamic settings.

Paper organization. The paper is organized as follow. Section 2 is dedicated to a brief introduction of the notion of probabilistic measure an recalls notations used in defining Signature I/O automata of [1]. Section 3 builds on the frameworks proposed in [1] and [2] in order to lay down the preliminaries of our formalism. More specifically, we introduce the definitions of probabilistic signed I/O automata and define their composition and implementation. In Section 4 we extend the definition of configuration automata proposed in [1] to probabilistic configuration automata then we define the composition of probabilistic configuration automata and prove its closeness. The key result of our formalisation, the monotonicity of PSIOA implementations with respect to creation and destruction, is presented in Section 5. The Appendix of the paper includes of the proofs and the intermediary results needed to the proof of our key result.

2 Preliminaries

Preliminaries on probability and measure. We assume our reader is comfortable with basic notions of probability theory, such as σ-fields and (discrete) probability measures. An extended abstract is provided in Appendix. A measurable space is denoted by $(S, \mathcal{F}_s)$, where S is a set and $\mathcal{F}_s$ is a σ-algebra over S. A measure space is denoted by $(S, \mathcal{F}_s, \eta)$ where η is a measure on $(S, \mathcal{F}_s)$. The product measure space $(S_1, \mathcal{F}_{s_1}, \eta_1) \otimes (S_2, \mathcal{F}_{s_2}, \eta_2)$ is the measure space $(S_1 \times S_2, \mathcal{F}_{s_1} \otimes \mathcal{F}_{s_2}, \eta_1 \otimes \eta_2)$, where $\mathcal{F}_{s_1} \otimes \mathcal{F}_{s_2}$ is the smallest σ-algebra generated by sets of the form $\{A \times B | A \in \mathcal{F}_{s_1}, B \in \mathcal{F}_{s_2}\}$ and $\eta_1 \otimes \eta_2$ is the unique measure s. t. for every $C_1 \in \mathcal{F}_{s_1}, C_2 \in \mathcal{F}_{s_2}$, $\eta_1 \otimes \eta_2(C_1 \times C_2) = \eta_1(C_1)\eta_2(C_2)$.

A discrete probability measure on a set S is a probability measure η on $(S, 2^S)$, such that, for each $C \subset S, \eta(S) = \sum_{c \in C} \eta(\{c\})$. We define $Disc(S)$ to be, the set of discrete probability measures on S. In the sequel, we often omit the set notation when we denote the measure of a singleton set. For a discrete probability measure η on a set S, $supp(\eta)$ denotes the support of η, that is, the set of elements $s \in X$ such that $\eta(s) \neq 0$. Given set S and a subset $C \subset S$, the Dirac measure δ_C is the discrete probability measure on S that assigns probability 1 to C. For each element $s \in S$, we note δ_s for $\delta_{\{s\}}$.

Signature I/O Automata (SIOA). Our framework builds on top of Signature I/O Automata (SIOA) introduced in [1]. We assume the existence of a countable set $Autids$ of unique signature input/output automata identifiers, an underlying universal set $Auts$ of SIOA, and a mapping $aut : Autids \rightarrow Auts$. $aut(\mathcal{A})$ is the SIOA with identifier $\mathcal{A}$. We use "the automaton $\mathcal{A}$" to mean "the SIOA with identifier $\mathcal{A}$". We use the letters $\mathcal{A}, \mathcal{B}$, possibly subscripted or primed, for SIOA identifiers. The executable actions of a SIOA $\mathcal{A}$ are drawn from a signature $sig(\mathcal{A})(q) = (in(\mathcal{A})(q), out(\mathcal{A})(q), int(\mathcal{A})(q))$, called the state signature, which is a function of the current state q of $\mathcal{A}$.

We node $in(\mathcal{A})(q), out(\mathcal{A})(q), int(\mathcal{A})(q)$ pairwise disjoint sets of input, output, and internal actions, respectively. We define $ext(\mathcal{A})(q)$, the external signature of $\mathcal{A}$ in state q, to be $ext(\mathcal{A})(q) = (in(\mathcal{A})(q), out(\mathcal{A})(q))$.

We define $local(\mathcal{A})(q)$, the local signature of $\mathcal{A}$ in state q, to be $local(\mathcal{A})(q) = (out(\mathcal{A})(q), in(\mathcal{A})(q))$. For any signature component, generally, the $\widehat{\cdot}$ operator yields the union of sets of actions within the signature, e.g., $\widehat{sig}(\mathcal{A}) : q \in Q \mapsto \widehat{sig}(\mathcal{A})(q) = in(\mathcal{A})(q) \cup out(\mathcal{A})(q) \cup int(\mathcal{A})(q)$. Also define $acts(\mathcal{A}) = \bigcup_{q \in Q} \widehat{sig}(\mathcal{A})(q)$, that is $acts(\mathcal{A})$ is the "universal" set of all actions that A could possibly execute, in any state. In the same way $UI(\mathcal{A}) = \bigcup_{q \in Q} in(\mathcal{A})(q)$, $UO(\mathcal{A}) = \bigcup_{q \in Q} out(\mathcal{A})(q)$, $UH(\mathcal{A}) = \bigcup_{q \in Q} int(\mathcal{A})(q)$, $UL(\mathcal{A}) = \bigcup_{q \in Q} \widehat{local}(\mathcal{A})(q)$, $UE(\mathcal{A}) = \bigcup_{q \in Q} \widehat{ext}(\mathcal{A})(q)$.

3 Probabilistic Signature I/O Automata

In the following we extend the definition of Signature I/O Automata introduced in [1] to probabilistic settings. We therefore, combine the formalisme in [1] with the Probabilistic I/O Automata defined in [9]. We will define the composition of PSIOA, measures for executions and traces and the notion of a environment for a PSIOA. Moreover, we extend the operators hidden and renaming to a PSIOA.

▶ **Definition 1** (probabilistic signature I/O automata)**.** A probabilistic signature I/O automata (PSIOA) $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$, where:

- (a) Q is a countable set of states, $(Q, 2^Q)$ is a measurable space called the state space, and $\bar{q}$ is the start state.

- (b) $sig(\mathcal{A}) : q \in Q \mapsto sig(\mathcal{A})(q) = (in(A)(q), out(A)(q), int(A)(q)$ is the signature function that maps each state to a triplet of countable input, output and internal set of actions.
- (d) $D \subset Q \times acts(\mathcal{A}) \times Disc(Q)$ is the set of probabilistic discrete transitions where $\forall (q, a, \eta) \in D : a \in \widehat{sig}(\mathcal{A})(q)$. If (q, a, η) is an element of D, we write $q \xrightarrow{a} \eta$ and action a is said to be *enabled* at q. The set of states in which action a is enabled is denoted by E_a. For $B \subseteq A$, we define E_B to be $\bigcup_{a \in B} E_a$. The set of actions enabled at q is denoted by $enabled(q)$. If a single action $a \in B$ is enabled at q and $q \xrightarrow{a} \eta$, then this η is denoted by $\eta_{(\mathcal{A},q,B)}$. If B is a singleton set $\{a\}$ then we drop the set notation and write $\eta_{(\mathcal{A},q,a)}$.

In addition $\mathcal{A}$ must satisfy the following conditions:

- $\mathbf{E_1}$ (Input action enabling) $\forall \mathbf{x} \in Q : \forall a \in in(\mathcal{A})(q), \exists \eta \in Disc(Q) : (q, a, \eta) \in D$.
- $\mathbf{T_1}$ Transition determinism: For every $q \in Q$ and $a \in A$ there is at most one $\eta \in Disc(Q)$ such that $(q, a, \eta) \in D$.

For every PSIOA $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$, we note $states(\mathcal{A}) = Q$, $start(\mathcal{A}) = \bar{q}$, $steps(\mathcal{A}) = D$.

▶ **Definition 2** (fragment, execution and trace of PSIOA)**.** An *execution fragment* of a PSIOA $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$ is a finite or infinite sequence $\alpha = q_0 a_1 q_1 a_2 ...$ of alternating states and actions, such that:

1. If α is finite, it ends with a state.
2. For every non-final state q_i, there is $\eta \in Disc(Q)$ and a transition $(q_i, a_{i+1}, \eta) \in D$ s. t. $q_{i+1} \in supp(\eta)$.

We write $fstate(\alpha)$ for q_0 (the first state of α), and if α is finite, we write $lstate(\alpha)$ for its last state. We use $Frags(\mathcal{A})$ (resp., $Frags^{\acute{\imath}*}(\mathcal{A})$) to denote the set of all (resp., all finite) execution fragments of $\mathcal{A}$. An *execution* of $\mathcal{A}$ is an execution fragment α with $fstate(\alpha) = \bar{q}$. $Execs(\mathcal{A})$ (resp., $Execs^*(\mathcal{A})$) denotes the set of all (resp., all finite) executions of $\mathcal{A}$. The *trace* of an execution fragment α, written $trace(\alpha)$, is the restriction of α to the external actions of $\mathcal{A}$. We say that β is a trace of $\mathcal{A}$ if there is $\alpha \in Execs(P)$ with $\beta = trace(\alpha)$. $Traces(\mathcal{A})$ (resp., $Traces^*(\mathcal{A})$) denotes the set of all (resp., all finite) traces of $\mathcal{A}$.

▶ **Definition 3** (reachable execution)**.** Let $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$ be a PSIOA. A state q is said *reachable* if it exists a finite execution that ends with q.

The aim of I/O formalism is to model distributed systems as composition of automata and prove guarantees of the composed system by composition of the guarantees of the different elements of the system. In the following we define the composition operation for PSIOA.

▶ **Definition 4** (Compatible signatures)**.** Let S be a set of signatures. Then S is compatible iff, $\forall sig, sig' \in S$, where $sig = (in, out, int), sig' = (in', out', int')$ and $sig \neq sig'$, we have: 1. $(in \cup out \cup int) \cap int' = \emptyset$, and 2. $out \cap out' = \emptyset$.

▶ **Definition 5** (Composition of Signatures)**.** Let $\Sigma = (in, out, int)$ and $\Sigma' = (in', out', int')$ be compatible signatures. Then we define their composition $\Sigma \times \Sigma = (in \cup in' - (out \cup out'), out \cup out', int \cup int')$.

Signature composition is clearly commutative and associative.

▶ **Definition 6** (partially compatible at a state)**.** Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a set of PSIOA. A *state* of $\mathbf{A}$ is an element $q = (q_1, ..., q_n) \in Q = Q_1 \times ... \times Q_n$. We say $\mathcal{A}_1, ..., \mathcal{A}_n$ are *partially-compatible* at state q (or $\mathbf{A}$ is) if $\{sig(\mathcal{A}_1)(q_1), ..., sig(\mathcal{A}_n)(q_n)\}$ is a set of compatible signatures. In this case we note $sig(\mathbf{A})(q) = sig(\mathcal{A}_1)(q_1) \times ... \times sig(\mathcal{A}_n)(q_n)$ and we note $\eta_{(\mathbf{A},q,a)} \in Disc(Q)$, s. t. for every action $a \in \widehat{sig}(\mathbf{A})(q)$, $\eta_{(\mathbf{A},q,a)} = \eta_1 \otimes ... \otimes \eta_n \in Disc(Q)$ that verifies for every $j \in [1, n]$:

- If $a \in sig(\mathcal{A}_j)(q_j)$, $\eta_j = \eta_{(\mathcal{A}_j,q_j,a)}$.
- Otherwise, $\eta_j = \delta_{q_j}$

while $\eta_{(\mathbf{A},q,a)} = \delta_q$ if $a \notin \widehat{sig}(\mathbf{A})(q)$.

▶ **Definition 7** (pseudo execution)**.** Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a set of PSIOA. A *pseudo execution fragment* of $\mathbf{A}$ is a finite or infinite sequence $\alpha = q^0 a^1 q^1 a^2 ...$ of alternating states of $\mathbf{A}$ and actions, such that:

- If α is finite, it ends with a n-uplet of state.
- For every non final state q^i, $\mathbf{A}$ is partially-compatible at q^i.
- For every action a^i, $a^i \in \widehat{sig}(\mathbf{A})(q^{i-1})$.
- For every state q^i, with $i > 0$, $q^i \in supp(\eta_{(\mathbf{A},q^{i-1},a^i)})$.

A *pseudo execution* of $\mathbf{A}$ is a pseudo execution fragment of $\mathbf{A}$ with $q^0 = (\bar{q}_{\mathcal{A}_1}, ..., \bar{q}_{\mathcal{A}_n})$.

▶ **Definition 8** (reachable state)**.** Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a set of PSIOA. A state q of $\mathbf{A}$ is *reachable* if it exists a pseudo execution α of $\mathbf{A}$ ending on state q.

▶ **Definition 9** (partially-compatible PSIOA)**.** Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a set of PSIOA. The automata $\mathcal{A}_1, ..., \mathcal{A}_n$ are *ℓ-partially-compatible* with $\ell \in \mathbb{N}$ if no pseudo-execution α of $\mathbf{A}$ with $|\alpha| \leq \ell$ ends on non-partially-compatible state q. The automata $\mathcal{A}_1, ..., \mathcal{A}_n$ are *partially-compatible* if $\mathbf{A}$ is partially-compatible at each reachable state q, i. e. $\mathbf{A}$ is *ℓ-partially-compatible* for every $\ell \in \mathbb{N}$.

▶ **Definition 10** (Compatible PSIOA)**.** Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a set of PSIOA with $\mathcal{A}_i = ((Q_i, \mathcal{F}_{Q_i}), sig(\mathcal{A}_i), D_i)$. We say $\mathbf{A}$ is compatible if it is partially-compatible for every state $q = (q_1, ..., q_n) \in Q_1 \times ... \times Q_n$.

Note that a set of compatible PSIOA is also a set of partially-compatible automata.

▶ **Definition 11** (PSIOAs composition)**.** If $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ is a compatible set of PSIOAs, with $\mathcal{A}_i = (Q_i, \bar{q}_i, sig(\mathcal{A}_i), D_i)$, then their composition $\mathcal{A}_1 || ... || \mathcal{A}_n$, is defined to be $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$, where:

- $Q = Q_1 \times ... \times Q_n$
- $\bar{q} = (\bar{q}_1, ..., \bar{q}_n)$
- $sig(\mathcal{A}) : q = (q_1, ..., q_n) \in Q \mapsto sig(\mathcal{A})(q) = sig(\mathcal{A}_1)(q_1) \times ... \times sig(\mathcal{A}_n)(q_n).$,
- $D \subset Q \times A \times Disc(Q)$ is the set of triples $(q, a, \eta_{(\mathbf{A},q,a)})$ so that $q \in Q$ and $a \in \widehat{sig}(\mathbf{A})(q)$

To solve the non-determinism we use schedule that allows us to chose an action in a signature. To do so, we adapt the definition of task of [2] to the dynamic setting. We assume the existence of a subset $Autids_0 \subset Autids$ that represents the "atomic ententies" that will constitute the configuration automata introduced in the next section.

▶ **Definition 12** (Constitution)**.** For every $\mathcal{A} \in Autids$, we note

$$constitution(\mathcal{A}) : \begin{cases} states(\mathcal{A}) & \rightarrow & \mathcal{P}(Autids_0) = 2^{Autids_0} \\ q & \mapsto & constitution(\mathcal{A})(q) \end{cases}$$

For every $\mathcal{A} \in Autids_0$, for every $q \in states(\mathcal{A})$, $constitution(\mathcal{A})(q) = \{\mathcal{A}\}$.

For every $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n) \in (Autids_0)^n$, $\mathcal{A} = \mathcal{A}_1 || ... || \mathcal{A}_n$ for every $q \in states(\mathcal{A})$, $constitution(\mathcal{A})(q) = \mathbf{A}$.

▶ **Definition 13** (Task). A task T is a pair $(id, actions)$ where $id \in Autids_0$ and $actions$ is a set of action labels. Let $T = (id, actions)$, we note $id(T) = id$ and $actions(T) = actions$.

▶ **Definition 14** (Enabled task). Let $\mathcal{A} \in Autids$. A task T is said *enabled* in state $q \in states(\mathcal{A})$ if :

- $id(T) \in constitution(\mathcal{A})(q)$
- It exists a unique local action $a \in \widehat{loc}(\mathcal{A})(q) \cap actions(T)$ (noted $a \in T$ to simplify) enabled at state q (that is it exists $\eta \in Disc(Q)$ s. t. $(q, a, \eta) \in D$.

In this case we say that a is *triggered* by T at state q.

We are not dealing with a schedule *of a specific automaton* anymore, which differs from [2]. However the restriction of our definition to "static" setting matches their definition.

▶ **Definition 15** (schedule). A schedule ρ is a (finite or infinite) sequence of tasks.

▶ **Definition 16.** Let $\mathcal{A}$ be a PSIOA. Given $\mu \in Disc(Frags(\mathcal{A}))$ a discrete probability measure on the execution fragments and a task schedule ρ, $apply(\mu, \rho)$ is a probability measure on $Frags(\mathcal{A})$. It is defined recursively as follows.

1. $apply_{\mathcal{A}}(\mu, \lambda) := \mu$. Here λ denotes the empty sequence.
2. For every T and $\alpha \in Frags^*(\mathcal{A})$, $apply(\mu, T)(\alpha) := p_1(\alpha) + p_2(\alpha)$, where:
   - $p_1(\alpha) = \begin{cases} \mu(\alpha')\eta_{(\mathcal{A},q',a)}(q) & \text{if } \alpha = \alpha' a q, q' = lstate(\alpha') \text{ and } a \text{ is triggered by } T \\ 0 & \text{otherwise} \end{cases}$
   - $p_2(\alpha) = \begin{cases} \mu(\alpha) & \text{if } T \text{ is not enabled after } \alpha \\ 0 & \text{otherwise} \end{cases}$
3. 3. If ρ is finite and of the form $\rho' T$, then $apply_{\mathcal{A}}(\mu, \rho) := apply_{\mathcal{A}}(apply_{\mathcal{A}}(\mu, \rho'), T)$.
4. 4. If ρ is infinite, let ρ_i denote the length-i prefix of ρ and let pm_i be $apply_{\mathcal{A}}(\mu, \rho_i)$. Then $apply_{\mathcal{A}}(\mu, \rho) := \underset{i \to \infty}{lim} pm_i$.

$tdist_{\mathcal{A}}(\mu, \rho) : Traces_{\mathcal{A}} \to [0, 1]$, is defined as $tdist_{\mathcal{A}}(\mu, \rho)(E) = apply(\delta_{\bar{q}}, \rho)(trace_{\mathcal{A}}^{-1}(E))$, for any measurable set $E \in \mathcal{F}_{Traces_{\mathcal{A}}}$.

We write $tdist_{\mathcal{A}}(\mu, \rho)$ as shorthand for $tdist_{\mathcal{A}}(apply_{\mathcal{A}}(\mu, \rho))$ and $tdist_{\mathcal{A}}(\rho)$ for $tdist_{\mathcal{A}}(apply_{\mathcal{A}}(\delta(\bar{x}), \rho))$, where $\delta(\bar{x})$ denotes the measure that assigns probability 1 to $\bar{x}$. A trace distribution of $\mathcal{A}$ is any $tdist_{\mathcal{A}}(\rho)$. We use $Tdists_{\mathcal{A}}$ to denote the set $\{tdist_{\mathcal{A}}(\rho) : \rho$ is a task schedule $\}$.

We removed the subscript $\mathcal{A}$ when this is clear in the context.

In the following we introduce the notion of a environment for a PSIOA.

▶ **Definition 17** (Environment). A probabilistic environment for PSIOA $\mathcal{A}$ is a PSIOA $\mathcal{E}$ such that $\mathcal{A}$ and $\mathcal{E}$ are partially-compatible.

▶ **Definition 18** (External behavior). The external behavior of a PSIOA $\mathcal{A}$, written as $ExtBeh_{\mathcal{A}}$, is defined as a function that maps each environment $\mathcal{E}$ for $\mathcal{A}$ to the set of trace distributions $Tdists_{\mathcal{A}||\mathcal{E}}$.

We introduce in the following the hiding and renaming operators for PSIOA.

▶ **Definition 19** (hiding on signature). Let $sig = (in, out, int)$ be a signature and $\underline{acts}$ a set of actions. We note $hide(sig, \underline{acts})$ the signature $sig' = (in', out', int')$ s. t.

- $in' = in$
- $out' = out \setminus \underline{acts}$
- $int' = int \cup (out \cap \underline{acts})$

▶ **Definition 20** (hiding on PSIOA)**.** Let $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$ be a PSIOA. Let *hiding-actions* a function mapping each state $q \in Q$ to a set of actions. We note $hide(\mathcal{A}, hiding\text{-}actions)$ the PSIOA $(Q, \bar{q}, sig'(\mathcal{A}), D)$, where $sig'(\mathcal{A}) : q \in Q \mapsto hide(sig(\mathcal{A})(q), hiding\text{-}actions(q))$.

It should be noted that hiding and composition are commutative. A formal proof can be found in the Appendix.

▶ **Definition 21.** (State renaming for PSIOA) Let $\mathcal{A}$ be a PSIOA with $Q_{\mathcal{A}}$ as set of states, let $Q_{\mathcal{A}'}$ be another set of states and let $ren : Q_{\mathcal{A}} \rightarrow Q_{\mathcal{A}'}$ be a bijective mapping. Then $ren(\mathcal{A})$ is the automaton given by:

- $start(ren(\mathcal{A})) = ren(start(Q_{\mathcal{A}}))$
- $states(ren(\mathcal{A})) = ren(states(Q_{\mathcal{A}}))$
- $\forall q_{\mathcal{A}'} \in states(ren(\mathcal{A})), sig(ren(\mathcal{A}))(q_{\mathcal{A}'}) = sig(\mathcal{A})(ren^{-1}(q_{\mathcal{A}'}))$
- $\forall q_{\mathcal{A}'} \in states(ren(\mathcal{A})), \forall a \in sig(ren(\mathcal{A}))(q_{\mathcal{A}'})$, if $(ren^{-1}(q_{\mathcal{A}'}), a, \eta) \in D_{\mathcal{A}}$, then $(q_{\mathcal{A}'}, a, \eta') \in D_{ren(\mathcal{A})}$ where $\eta' \in Disc(Q_{\mathcal{A}'}, \mathcal{F}_{Q_{\mathcal{A}'}})$ and for every $q_{\mathcal{A}''} \in states(ren(\mathcal{A}))$, $\eta'(q_{\mathcal{A}''}) = \eta(ren^{-1}(q_{\mathcal{A}''}))$.

▶ **Definition 22.** (State renaming for PSIOA execution) Let $\mathcal{A}$ and $\mathcal{A}'$ be two PSIOA s. t. $\mathcal{A}' = ren(\mathcal{A}')$. Let $\alpha = q^0 a^1 q^1 ...$ be an execution fragment of $\mathcal{A}$. We note $ren(\alpha)$ the sequence $ren(q^0)a^1 ren(q^1)....$

4 Probabilistic Configuration Automata

Towards the extension of the formalism to dynamic settings, in this section we introduce the Probabilistic Configuration Automata (PCA) that combines the PSIOA framework defined above and the notion of configuration of [1]. The main key result we prove here is the closeness of PCA closeness under composition.

▶ **Definition 23** (Configuration)**.** A configuration is a pair $(\mathbf{A}, \mathbf{S})$ where

- $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ is a finite sequence of PSIOA identifiers (lexicographically ordered [1]), and
- $\mathbf{S}$ maps each $\mathcal{A}_k \in \mathbf{A}$ to an $s_k \in states(\mathcal{A}_k)$.

In distributed computing, configuration usually refers to the union of states of **all** the automata of the system. Here, the notion is different, it captures a set of some automata $(\mathbf{A})$ in their current state $(\mathbf{S})$.

▶ **Definition 24** (Compatible configuration)**.** A configuration $(\mathbf{A}, \mathbf{S})$ is compatible iff, for all $\mathcal{A}, \mathcal{B} \in \mathbf{A}$, $\mathcal{A} \neq \mathcal{B}$: 1. $sig(\mathcal{A})(\mathbf{S}(\mathcal{A})) \cap int(\mathcal{B})(\mathbf{S}(\mathcal{B})) = \emptyset$, and 2. $out(\mathcal{A})(\mathbf{S}(\mathcal{A})) \cap out(\mathcal{B})(\mathbf{S}(\mathcal{B})) = \emptyset$

▶ **Definition 25** (Intrinsic attributes of a configuration)**.** Let $C = (\mathbf{A}, \mathbf{S})$ be a compatible task-configuration. Then we define

[1] lexicographic order will simplify projection on product of probabilistic measure for transition of composition of automata

- $auts(C) = \mathbf{A}$ represents the automata of the configuration,
- $map(C) = \mathbf{S}$ maps each automaton of the configuration with its current state,
- $out(C) = \bigcup_{\mathcal{A}\in\mathbf{A}} out(\mathcal{A})(\mathbf{S}(\mathcal{A}))$ represents the output action of the configuration,
- $in(C) = (\bigcup_{\mathcal{A}\in\mathbf{A}} in(\mathcal{A})(\mathbf{S}(\mathcal{A}))) - out(C)$ represents the input action of the configuration,
- $int(C) = \bigcup_{\mathcal{A}\in\mathbf{A}} int(\mathcal{A})(\mathbf{S}(\mathcal{A}))$ represents the internal action of the configuration,
- $ext(C) = in(C) \cup out(C)$ represents the external action of the configuration,
- $sig(C) = (in(C), out(C), int(C))$ is called the intrinsinc signature of the configuration,
- $CA(C) = (aut(\mathcal{A}_1)||...||aut(\mathcal{A}_n))$ represents the composition of all the automata of the configuration,
- $US(C) = (\mathbf{S}(\mathcal{A}_1), ..., \mathbf{S}(\mathcal{A}_n))$ represents the states of the automaton corresponding to the composition of all the automata of the configuration,

Here we define a reduced configuration as a configuration deprived of the automata that are in the very particular state where their current signatures are the empty set. This mechanism will allows us to capture the idea of destruction.

▶ **Definition 26** (Reduced configuration). $reduce(C) = (\mathbf{A}', \mathbf{S}')$, where $\mathbf{A}' = \{\mathcal{A}|\mathcal{A} \in \mathbf{A}$ and $sig(\mathcal{A})(\mathbf{S}(\mathcal{A})) \neq \emptyset\}$ and $\mathbf{S}'$ is the restriction of $\mathbf{S}$ to $\mathbf{A}'$, noted $\mathbf{S} \upharpoonright \mathbf{A}'$ in the remaining.

A configuration C is a reduced configuration iff $C = reduce(C)$.

We recall that we assume the existence of a countable set *Autids* of unique PSIOA identifiers, an underlying universal set *Auts* of *PSIOA*, and a mapping $aut : Autids \rightarrow Auts$. $aut(\mathcal{A})$ is the *PSIOA* with identifier $\mathcal{A}$. We will define a measurable space for configuration. We note for every $\varphi \in \mathcal{P}(Autids)$, $Q_\varphi = Q_{\varphi_1} \times ... \times Q_{\varphi_n}$ and $\mathcal{F}_{Q_\varphi} = \mathcal{F}_{Q_{\varphi_1}} \otimes ... \otimes \mathcal{F}_{Q_{\varphi_{|\varphi|}}}$

We note $Q_{aut} = \bigcup_{\varphi\in\mathcal{P}(Autids)} Q_\varphi$, the set of all possible state sets cartesian product for each possible family of automata. $\mathcal{F}_{Q_{aut}} = \{\bigcup_{i\in[1,k]} c_i|\phi \in \mathcal{P}(\mathcal{P}(Autids)), c_i \in \mathcal{F}_{Q_{\varphi_i}} \phi = \varphi_1, ..., \varphi_k, \varphi_i \in \mathcal{P}(Autids)\}$ $(Q_{aut}, \mathcal{F}_{Q_{aut}})$ is a measurable space.

We note $Q_{conf} = \{(\mathbf{A}, \mathbf{S})|\mathbf{A} \in \mathcal{P}(Autids), \forall\mathcal{A}_i \in \mathbf{A}, \mathbf{S}(\mathcal{A}_i) \in Q_i\}$, the set of all possible configurations.

Let $f = \begin{cases} Q_{conf} & \rightarrow & Q_{aut} \\ (\mathbf{A}, \mathbf{S}) & \mapsto & Q_{CA((\mathbf{A},\mathbf{S}))} = \mathbf{S}(\mathcal{A}_1) \times ... \times \mathbf{S}(\mathcal{A}_n) \end{cases}$

We note $\mathcal{F}_{Q_{conf}} = \{f^{-1}(P)|P \in \mathcal{F}_{Q_{aut}}\}$.

$(Q_{conf}, \mathcal{F}_{Q_{conf}})$ is a measurable space

We will define some probabilistic transition from configurations to others where some automata can be destroyed or created. To define it properly, we start by defining "preserving transition" where no automaton is neither created nor destroyed and then we define above this definition the notion of configuration transition.

▶ **Definition 27** (Preserving distribution). A *preserving distribution* $\eta_p \in Disc(Q_{conf})$ is a distribution verifying $\forall(\mathbf{A}, \mathbf{S}), (\mathbf{A}', \mathbf{S}') \in supp(\eta_p), \mathbf{A} = \mathbf{A}'$. The unique family of automata ids $\mathbf{A}$ of the configurations in the support of η_p is called the *family support* of η_p.

We define a companion distribution as the natural distribution of the corresponding family of automata at the corresponding current state. Since no creation or destruction occurs, these definitions can seem redundant, but this is only an intermediate step to define properly the "dynamic" distribution.

▶ **Definition 28** (Companion distribution). Let $C = (\mathbf{A}, \mathbf{S})$ be a compatible configuration with $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ and $\mathbf{S} : \mathcal{A}_i \in \mathbf{A} \mapsto q_i \in Q_{\mathcal{A}_i}$ (with $\mathbf{A}$ partially-compatible at state $q = (q_1, ..., q_n) \in Q_{\mathbf{A}} = Q_{\mathcal{A}_1} \times ... \times Q_{\mathcal{A}_n}$). Let η_p be a preserving distribution with $\mathbf{A}$ as family support. The probabilistic distribution $\eta_{(\mathbf{A},q,a)}$ is a *companion distribution* of η_p if for every $q' = (q'_1, ..., q'_n) \in Q_{\mathbf{A}}$, for every $\mathbf{S}'' : \mathcal{A}_i \in \mathbf{A} \mapsto q''_i \in Q_{\mathcal{A}_i}$,

$$\eta_{(\mathbf{A},q,a)}(q') = \eta_p((\mathbf{A}, \mathbf{S}'')) \iff \forall i \in [1, n], q''_i = q'_i,$$

that is the distribution $\eta_{(\mathbf{A},q,a)}$ corresponds exactly to the distribution η_p.

This is "a" and not "the" companion distribution since η_p does not explicit the start configuration.

Now, we can naturally define a preserving transition (C, a, η_p) from a configuration C via an action a with a companion transition of η_p. It allows us to say what is the "static" probabilistic transition from a configuration C via an action a if no creation or destruction occurs.

▶ **Definition 29** (preserving transition). Let $C = (\mathbf{A}, \mathbf{S})$ be a compatible configuration, $q = US(C)$ and $\eta_p \in P(Q_{conf}, \mathcal{F}_{Q_{conf}})$ be a preserving transition with $\mathbf{A}_s$ as family support.

Then say that (C, a, η_p) is a *preserving configuration transition*, noted $C \xrightarrow{a} \eta_p$ if

- $\mathbf{A}_s = \mathbf{A}$
- $\eta_{(\mathbf{A},q,a)}$ is a companion distribution of η_p

For every preserving configuration transition (C, a, η_p), we note $\eta_{(C,a),p} = \eta_p$.

The preserving transition of a configuration corresponds to the transition of the composition of the corresponding automata at their corresponding current states.

Now we are ready to define our "dynamic" transition, that allows a configuration to create or destroy some automata.

At first, we define reduced distribution that leads to reduced configurations only, where all the automata that reach a state with an empty signature are destroyed.

▶ **Definition 30** (reduced distribution). A *reduced* distribution $\eta_r \in Disc(Q_{conf}, \mathcal{F}_{Q_{conf}})$ is a probabilistic distribution verifying that for every configuration $C \in supp(\eta_r)$, $C = reduced(C)$.

Now, we generate reduced distribution with a preserving distribution that describes what happen to the automata that already exist and a family of new automata that are created.

▶ **Definition 31** (Generation of reduced distribution). Let $\eta_p \in Disc(Q_{conf})$ be a preserving distribution with $\mathbf{A}$ as family support. Let $\varphi \subset Autids$. We say the reduced distribution $\eta_r \in Disc(Q_{conf})$ is generated by η_p and φ if it exists a non-reduced distribution $\eta_{nr} \in Disc(Q_{conf})$, s. t.

- (φ is created with probability 1)
  $\forall (\mathbf{A}'', \mathbf{S}'') \in Q_{conf}$, if $\mathbf{A}'' \neq \mathbf{A} \cup \varphi$, then $\eta_{nr}((\mathbf{A}'', \mathbf{S}'')) = 0$
- (freshly created automata start at start state)
  $\forall (\mathbf{A}'', \mathbf{S}'') \in Q_{conf}$, if $\exists \mathcal{A}_i \in \varphi - \mathbf{A}$ so that, $\mathbf{S}''(\mathcal{A}_i) \neq \bar{q}_i$, then $\eta_{nr}((\mathbf{A}'', \mathbf{S}'')) = 0$
- (The non-reduced transition match the preserving transition)
  $\forall (\mathbf{A}'', \mathbf{S}'') \in Q_{conf}$, s. t. $\mathbf{A}'' = \mathbf{A} \cup \varphi$ and $\forall \mathcal{A}_j \in \varphi, \mathbf{S}''(\mathcal{A}_j = \overline{\mathbf{x}}_j)$, $\eta_{nr}((\mathbf{A}'', \mathbf{S}'')) = \eta_p(\mathbf{A}, \mathbf{S}''\lceil\mathbf{A}))$

- (The reduced transition match the non-reduced transition) $\forall c' \in Q_{conf}$, if $c' = reduce(c'), \eta_r(c') = \Sigma_{(c'',c'=reduce(c''))}\eta_{nr}(c'')$, if $c' \neq reduce(c')$, then $\eta_r(c') = 0$

▶ **Definition 32** (Intrinsic transition). Let $(\mathbf{A}, \mathbf{S})$ be arbitrary reduced compatible configuration, let $\eta \in Disc(Q_{conf})$, and let $\varphi \subseteq Autids$, $\varphi \cap \mathbf{A} = \emptyset$. Then $\langle \mathbf{A}, \mathbf{S}\rangle \stackrel{a}{\Longrightarrow}_{\varphi} \eta$ if η is generated by η_p and φ with $(\mathbf{A}, \mathbf{S}) \stackrel{a}{\rightharpoonup} \eta_p$.

The assumption of deterministic creation is not restrictive, nothing prevents from flipping a coin at state s_0 to reach s_1 with probability p or s_2 with probability $1-p$ and only create a new automaton in state s_2 with probability 1, while the action create is not enabled in state s_1.

▶ **Definition 33** (Probabilistic Configuration Automaton). A probabilistic configuration automaton (PCA) K consists of the following components:

- 1. A probabilistic signature I/O automaton $psioa(K)$. For brevity, we define $states(K) = states(psioa(K)), start(K) = start(psioa(K)), sig(K) = sig(psioa(K)), steps(K) = steps(psioa(K))$, and likewise for all other (sub)components and attributes of $psioa(K)$.
- 2. A configuration mapping $config(K)$ with domain $states(K)$ and such that $config(K)(x)$ is a reduced compatible configuration for all $q_K \in states(K)$.
- 3. For each $q_K \in states(K)$, a mapping $created(K)(\mathbf{x})$ with domain $sig(K)(\mathbf{x})$ and such that $\forall a \in sig(K)(q), created(K)(q)(a) \subseteq Autids$
- 4. A hidden-actions mapping $hidden\text{-}actions(K)$ with domain $states(K)$ and such that $hidden\text{-}actions(K)(q_K) \subseteq out(config(K)(q_K))$.

and satisfies the following constraints

- 1. If $config(K)(\bar{q}_K) = (\mathbf{A}, \mathbf{S})$, then $\forall \mathcal{A}_i \in \mathbf{A}, \mathbf{S}(\mathcal{A}_i) = \bar{q}_i$
- 2. If $(q_K, a, \eta) \in steps(K)$ then $config(K)(q_K) \stackrel{a}{\Longrightarrow}_{\varphi} \eta'$, where $\varphi = created(K)(q_K)(a)$ and $\eta(y) = \eta'(config(K)(y))$ for every $\mathbf{y} \in states(K)$
- 3. If $q_K \in states(K)$ and $config(K)(q_K) \stackrel{a}{\Longrightarrow}_{\varphi} \eta'$ for some action a, $\varphi = created(K)(x)(a)$, and reduced compatible probabilistic measure $\eta' \in P(Q_{conf}, \mathcal{F}_{Q_{conf}})$, then $(q_K, a, \eta) \in steps(K)$ with $\eta(\mathbf{y}) = \eta'(config(K)(\mathbf{y}))$ for every $\mathbf{y} \in states(K)$.
- 4. For all $q_K \in states(K)$, $sig(K)(q_K) = hide(sig(config(K)(q_K)), hidden\text{-}actions(q_K))$, which implies that
  - (a) $out(K)(q_K) \subseteq out(config(K)(q_K))$,
  - (b) $in(K)(q_K) = in(config(K)(q_K))$,
  - (c) $int(K)(q_K) \supseteq int(config(K)(q_K))$, and
  - (d) $out(K)(q_K) \cup int(X)(q_K) = out(config(K)(q_K)) \cup int(config(K)(q_K))$

4 (d) states that the signature of a state q_K of K must be the same as the signature of its corresponding configuration $config(K)(q_K)$, except for the possible effects of hiding operators, so that some outputs of $config(K)(q_K)$ may be internal actions of K in state q_K.

Additionally, we can define the current constitution of a PCA, which is the union of the current constitution of the element of its current corresponding configuration.

▶ **Definition 34** (Constitution of a PCA). Let K be a PCA. For every $q \in states(K)$,

$$constitution(K)(q) = constitution(psioa(K))(q) = \bigcup_{\mathcal{A} \in auts(config(K)(q))} constitution(\mathcal{A})(map(config(K)(q))$$

We note $UA(K) = \bigcup_{q \in K} constitution(K)(q)$ the universal set of atomic components of K.

In the following we lay down the formalism needed to prove that probabilistic configuration automata are closed under composition.

▶ **Definition 35** (Union of configurations). Let $C_1 = (\mathbf{A}_1, \mathbf{S}_1)$ and $C_2 = (\mathbf{A}_2, \mathbf{S}_2)$ be configurations such that $\mathbf{A}_1 \cap \mathbf{A}_2 = \emptyset$. Then, the union of C_1 and C_2, denoted $C_1 \cup C_2$, is the configuration $C = (\mathbf{A}, \mathbf{S})$ where $\mathbf{A} = \mathbf{A}_1 \cup \mathbf{A}_2$ (lexicographically ordered) and $\mathbf{S}$ agrees with $\mathbf{S}_1$ on $\mathbf{A}_1$, and with $\mathbf{S}_2$ on $\mathbf{A}_2$. It is clear that configuration union is commutative and associative. Hence, we will freely use the n-ary notation $C_1 \cup ... \cup C_n$ (for any $n \geq 1$) whenever $\forall i, j \in [1:n], i \neq j, auts(C_i) \cap auts(C_j) = \emptyset$.

▶ **Definition 36** (PCA partially-compatible at a state). Let $\mathbf{X} = (X_1, ..., X_n)$ be a family of PCA. We note $psioa(\mathbf{X}) = (psioa(X_1), ..., psioa(X_n))$. The PCA $X_1, ..., X_n$ are partially-compatible at state $q_{\mathbf{X}} = (q_{X_1}, ..., q_{X_n}) \in states(X_1) \times ... \times states(X_n)$ iff:

1. $\forall i, j \in [1:n], i \neq j : auts(config(X_i)(q_{X_i})) \cap auts(config(X_j)(q_{X_j})) = \emptyset$.
2. $\{sig(X_1)(q_{X_1}), ..., sig(X_n)(q_{X_n})\}$ is a set of compatible signatures.
3. $\forall i, j \in [1:n], i \neq j : \forall a \in \widehat{sig}(X_i)(q_{X_i}) \cap \widehat{sig}(X_j)(q_{X_j}) : created(X_i)(q_{X_i})(a) \cap created(X_j)(q_{X_j})(a) = \emptyset$.
4. $\forall i, j \in [1:n], i \neq j : constitution(X_i)(q_{X_i}) \cap constitution(X_j)(q_{X_j}) = \emptyset$

We can remark that if $\forall i, j \in [1:n], i \neq j : auts(config(X_i)(q_{X_i})) \cap auts(config(X_j)(q_{X_j})) = \emptyset$ and $\{sig(X_1)(q_{X_1}), ..., sig(X_n)(q_{X_n})\}$ is a set of compatible signatures, then $config(X_1)(q_{X_1}) \cup ... \cup config(X_n)(q_{X_n})$ is a reduced compatible configuration.

If $\mathbf{X}$ is partially-compatible at state $q_{\mathbf{X}}$, for every action $a \in \widehat{sig}(psioa(\mathbf{X}))(q_{\mathbf{X}})$, we note $\eta_{(\mathbf{X}, q_{\mathbf{X}}, a)} = \eta_{(psioa(\mathbf{X}), q_{\mathbf{X}}, a)}$ and we extend this notation with $\eta_{(\mathbf{X}, q_{\mathbf{X}}, a)} = \delta_{q_{\mathbf{X}}}$ if $a \notin \widehat{sig}(psioa(\mathbf{X}))(q_{\mathbf{X}})$.

▶ **Definition 37** (pseudo execution). Let $\mathbf{X} = (X_1, ..., X_n)$ be a set of PCA. A *pseudo execution fragment* of $\mathbf{X}$ is a pseudo execution fragment of $psioa(\mathbf{A})$, s. t. for every non final state q^i, $\mathbf{X}$ is partially-compatible at state q^i (namely the conditions (1) and (3) need to be satisfied)

A *pseudo execution* α of $\mathbf{X}$ is a pseudo execution fragment of $\mathbf{X}$ with $fstate(\alpha) = (\bar{q}_{X_1}, ..., \bar{q}_{X_n})$.

▶ **Definition 38** (reachable state). Let $\mathbf{X} = (X_1, ..., X_n)$ be a set of PSIOA. A state q of $\mathbf{X}$ is *reachable* if it exists a pseudo execution α of $\mathbf{X}$ ending on state q.

▶ **Definition 39** (partially-compatible PCA). Let $\mathbf{X} = (X_1, ..., X_n)$ be a set of PCA. The automata $X_1, ..., X_n$ are *ℓ-partially-compatible* with $\ell \in \mathbb{N}$ if no pseudo-execution α of $\mathbf{X}$ with $|\alpha| \leq \ell$ ends on non-partially-compatible state q. The automata $X_1, ..., X_n$ are *partially-compatible* if $\mathbf{X}$ is partially-compatible at each reachable state q, i. e. $\mathbf{X}$ is *ℓ-partially-compatible* for every $\ell \in \mathbb{N}$.

▶ **Definition 40** (compatible PCA). Let $\mathbf{X} = (X_1, ..., X_n)$ be a set of PCA. The automata $X_1, ..., X_n$ are *compatible* if the automata $X_1, ..., X_n$ are partially-compatible for each state of $states(X_1) \times ... \times states(X_n)$.

▶ **Definition 41** (Composition of configuration automata). Let $X_1, ..., X_n$, be compatible (resp. partially-compatible) configuration automata. Then $X = X_1 || ... || X_n$ is the state machine consisting of the following components:

1. $psioa(X) = psioa(X_1) || ... || psioa(X_n)$ (where the composition can be the one dedicated to only partially-compatible PCA).

2. A configuration mapping $config(X)$ given as follows. For each $x = (x_1, ..., x_n) \in states(X)$, $config(X)(x) = config(X_1)(x_1) \cup ... \cup config(X_n)(x_n)$.
3. For each $x = (x_1, ..., x_n) \in states(X)$, a mapping $created(X)(x)$ with domain $\widehat{sig}(X)(x)$ and given as follows. For each $a \in \widehat{sig}(X)(x)$, $created(X)(x)(a) = \bigcup_{a \in \widehat{sig}(X_i)(x_i), i \in [1:n]} created(X_i)(x_i)(a)$.
4. A hidden-action mapping $hidden\text{-}actions(X)$ with domain $states(X)$ and given as follows. For each $x = (x_1, ..., x_n) \in states(X)$, $hidden\text{-}actions(x) = \bigcup_{i \in [1:n]} hidden\text{-}actions(x_i)$

We define $states(X) = states(sioa(X)), start(X) = start(sioa(X)), sig(X) = sig(sioa(X)), steps(X) = steps(sioa(X))$, and likewise for all other (sub)components and attributes of $sioa(X)$.

▶ **Theorem 42** (PCA closeness under composition). *Let $X_1, ..., X_n$, be compatible or partially-compatible PCA. Then $X = X_1||...||X_n$ is a PCA.*

5 Monotonicity of implementations with respect to automata creation and destruction

This section lays down the formalism to prove the key notion of our framework: the monotonicity of implementations with respect to automata creation and destruction. We will introduce the equivalence classes of executions, the notion of schedule and implementation and finally our key result.

▶ **Definition 43** (Execution correspondence relation, $S_{\mathcal{ABE}}$)). Let $\mathcal{A}, \mathcal{B}$ be PSIOA, let $\mathcal{E}$ be an environment for both $\mathcal{A}$ and $\mathcal{B}$. Let α, π be executions of automata $\mathcal{A}||\mathcal{E}$ and $\mathcal{B}||\mathcal{E}$ respectively.

Then $\alpha S_{(AB\mathcal{E})} \pi$ if

1. $\mathcal{A}$ is permanently off in $\alpha \iff \mathcal{B}$ is permanently off in π. $\mathcal{A}$ is permanently on in $\alpha \iff \mathcal{B}$ is permanently on in π.
2. (*) $\mathcal{A}$ is turned off in $\alpha \iff \mathcal{B}$ is turned off in π. If (*), we can note $\alpha = \alpha_1 ^\frown \alpha_2$ and $\alpha_1 = \alpha_1' ^\frown a q_1$, where $\widehat{sig}(\mathcal{A})(lstate(\alpha_1) \upharpoonright \mathcal{A}) = \emptyset$, $\widehat{sig}(\mathcal{A})(lstate(\alpha_1') \upharpoonright \mathcal{A}) \neq \emptyset$ and we can note $\pi = \pi_1 ^\frown \pi_2$ similarly.
3. $\pi \upharpoonright \mathcal{E} = \alpha \upharpoonright \mathcal{E}$. If (*), $\pi_i \upharpoonright \mathcal{E} = \alpha_i \upharpoonright \mathcal{E}$ for $i \in \{1, 2\}$.
4. $trace_{\mathcal{B}||\mathcal{E}}(\pi) = trace_{\mathcal{A}||\mathcal{E}}(\alpha)$. If (*) $trace_{\mathcal{B}||\mathcal{E}}(\pi_i) = trace_{\mathcal{A}||\mathcal{E}}(\alpha_i)$ for $i \in \{1, 2\}$.
5. $ext(\mathcal{A})(fstate(\alpha) \upharpoonright \mathcal{A}) = ext(\mathcal{B})(fstate(\pi) \upharpoonright \mathcal{B})$; $ext(\mathcal{A})(lstate(\alpha) \upharpoonright \mathcal{A}) = ext(\mathcal{B})(lstate(\pi) \upharpoonright \mathcal{B})$.

$S_{\mathcal{ABE}}$ is sometimes written $S_{\mathcal{AB}}$ hen the environment is clear in the context.

▶ **Definition 44** (equivalence class). Let $\mathcal{A}$ be a PSIOA. Let $\mathcal{E}$ be an environment of $\mathcal{A}$. Let α be an execution fragment of $\mathcal{A}||\mathcal{E}$. We note $\underline{\alpha}_{\mathcal{AE}} = \{\alpha' | \alpha' S_{\mathcal{A}} \alpha\}$

When this is clear in the context, we note $\underline{\alpha}_{\mathcal{A}}$ or even $\underline{\alpha}$ for $\underline{\alpha}_{\mathcal{AE}}$ and $\underline{\tilde{\alpha}}$ for $\underline{\tilde{\alpha}}_{\mathcal{A}}$.

In the following we introduce the notion of schedule.

▶ **Definition 45** (simple schedule notation). Let $\rho = T^{\ell}, T^{\ell+1}, ..., T^h$ be a schedule, i. e. a sequence of tasks. For every $q, q' \in [\ell, h], q \leq q'$, we note:

- $hi(\rho) = h$ the highest index in ρ
- $li(\rho) = \ell$ the lowest index in ρ
- $\rho|_q = T^{\ell}...T^q$
- $_q|\rho = T^q...T^h$
- $_q|\rho|_{q'} = T^q...T^{q'}$

By doing so, we implicitly assume an indexation of ρ, $ind(\rho) : ind \in [li(\rho), hi(\rho)] \mapsto T^{ind} \in \rho$. Hence if $\rho = T^1, T^2, ..., T^k, T^{k+1}, ..., T^q, T^{q+1} ..., T^h$, $\rho' =_k |\rho$, $\rho'' =_q |\rho'$, then $\rho'' =_q |\rho$.

▶ **Definition 46** (Schedule partition and index)**.** Let ρ be a schedule. A partition p of ρ is a sequence of schedules (finite or infinite) $p = (\rho^m, \rho^{m+1}, ..., \rho^n, ...)$ so that ρ can be written $\rho = \rho^m, \rho^{m+1}, ..., \rho^n, ...$. We note $min(p) = m$ and $max(p) = card(p) + m - 1$.

A total ordered set $(ind(\rho, p), \prec) \subset \mathbb{N}^2$ is defined as follows :

$ind(\rho, p) = \{(k, q) \in (\mathbb{N}^*)^2 | k \in [min(p), max(p)], q \in [li(\rho^k), hi(\rho^k)]\}$ For every $\ell = (k, q), \ell' = (k', q') \in ind(\rho, p)$:

- If $k < k'$, then $\ell \prec \ell'$
- If $k = k', q < q'$, then $\ell \prec \ell'$
- If $k = k'$ and $q = q'$, then $\ell = \ell'$. If either $\ell \prec \ell'$ or $\ell = \ell'$, we note $\ell \preceq \ell'$.

▶ **Definition 47** (Schedule notation)**.** Let ρ be a schedule. Let p be a partition of ρ. For every $\ell = (k, q), \ell' = (k', q') \in ind(\rho, p)^2, \ell \preceq \ell'$, we note (when this is allowed):

- $\rho|_{(p,\ell)} = \rho^1, ..., \rho^k|_q$
- $_{(p,\ell)}|\rho = (_q|\rho^k), ...$
- $_\ell|\rho|_{(p,\ell')} = (_q|\rho^k), ..., (\rho^{k'}|_q)$

The symbol p of the partition is removed when it is clear in the context.

▶ **Definition 48** ($\mathcal{A}$-partition of a schedule)**.** Let $\mathcal{A}$ be a PCA or a PSIOA. Let $\rho_{\mathcal{A}\mathcal{E}}$ be a schedule. Since each task of $\rho_{\mathcal{A}\mathcal{E}}$ is either a task of $UA(\mathcal{A})$ or not. It is always possible to build the unique partition of $\rho_{\mathcal{A}\mathcal{E}}$: $(\rho^1_{\mathcal{A}}, \rho^2_{\mathcal{E}}, \rho^3_{\mathcal{A}}, \rho^4_{\mathcal{E}} ...)$ where $\rho^k_{\mathcal{A}}$ is a sequence of tasks of $UA(\mathcal{A})$ only and $\rho^{2k}_{\mathcal{E}}$ does not contain any task of $UA(\mathcal{A})$. We call such a partition, the $\mathcal{A}$-partition of $\rho_{\mathcal{A}\mathcal{E}}$.

▶ **Definition 49** (Environment corresponding schedule)**.** Let $\mathcal{A}$ and $\mathcal{B}$ be two PCA or two PSIOA. Let $\rho_{\mathcal{A}\mathcal{E}}$ and $\rho_{\mathcal{B}\mathcal{E}}$ be two schedules. Let $(\rho^1_{\mathcal{A}}, \rho^2_{\mathcal{E}}, \rho^3_{\mathcal{A}}, \rho^4_{\mathcal{E}} ...)$ (resp. $\rho_{\mathcal{B}\mathcal{E}}$: $(\rho^1_{\mathcal{B}}, \rho^{2'}_{\mathcal{E}}, \rho^3_{\mathcal{B}}, \rho^{4'}_{\mathcal{E}}, ...)$) be the $\mathcal{A}$-partition (resp. $\mathcal{B}$-partition) of $\rho_{\mathcal{A}\mathcal{E}}$ (resp. $\rho_{\mathcal{B}\mathcal{E}}$). We say that $\rho_{\mathcal{A}\mathcal{E}}$ and $\rho_{\mathcal{B}\mathcal{E}}$ are *$\mathcal{A}\mathcal{B}$-environment-corresponding* if for every k, $\rho^{2k}_{\mathcal{E}} = \rho^{2k'}_{\mathcal{E}}$.

In the following we introduce the notions of implementation and tenacious implementation and the conditions under which the monotonicity theorem holds.

▶ **Definition 50** ($S^s_{\mathcal{A}\mathcal{B}}$)**.** Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $\mathcal{E}$ be an environment of both $\mathcal{A}$ and $\mathcal{B}$. Let ρ and ρ' be two schedule. We say that $\rho S^s_{(\mathcal{A},\mathcal{B},\mathcal{E})} \rho'$ if :

for every executions α, π of $\mathcal{A}||\mathcal{E}$ and $\mathcal{B}||\mathcal{E}$ respectively, s. t. $\alpha S_{\mathcal{A}\mathcal{B}\mathcal{E}} \pi$, then

$apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_{\mathcal{E}})}, \rho)(\underline{\alpha}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B, \bar{q}_{\mathcal{E}})}, \rho')(\underline{\pi})$.

▶ **Definition 51** (Tenacious implementation)**.** Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. We say that $\mathcal{A}$ *tenaciously implements* $\mathcal{B}$, noted $\mathcal{A} \leq^{ten} \mathcal{B}$, iff for every schedule ρ, it exists a $\mathcal{A}\mathcal{B}$-environment-corresponding schedule ρ' s. t. for every environment $\mathcal{E}$ of both $\mathcal{A}$ and $\mathcal{B}$, for every $\ell = (2k, q)$, $\ell' = (2k', q') \in ind(\rho, p) \cap ind(\rho', p')$, $(_\ell|\rho|_{\ell'}) S^s_{(\mathcal{A},\mathcal{B},\mathcal{E})} (_\ell|\rho'|_{\ell'})$

▶ **Definition 52** ($\lhd_{\mathcal{A}\mathcal{B}}$-corresponding configurations)**.** (see figure **??**) Let $\Phi \subseteq Autids$, and $\mathcal{A}, \mathcal{B}$ be SIOA identifiers. Then we define $\Phi[\mathcal{B}/\mathcal{A}] = (\Phi \setminus \mathcal{A}) \cup \{\mathcal{B}\}$ if $\mathcal{A} \in \Phi$, and $\Phi[\mathcal{B}/\mathcal{A}] = \Phi$ if $\mathcal{A} \notin \Phi$. Let C, D be configurations. We define $C \lhd_{\mathcal{A}\mathcal{B}} D$ iff (1) $auts(D) = auts(C)[\mathcal{B}/\mathcal{A}]$, (2) for every $\mathcal{A}' \notin auts(C) \setminus \{\mathcal{A}\} : map(D)(\mathcal{A}') = map(C)(\mathcal{A}')$, and (3) $ext(\mathcal{A})(s) = ext(\mathcal{B})(t)$

where $s = map(C)(\mathcal{A}), t = map(D)(\mathcal{B})$. That is, in $\lhd_{\mathcal{AB}}$-corresponding configurations, the SIOA other than $\mathcal{A}, \mathcal{B}$ must be the same, and must be in the same state. $\mathcal{A}$ and $\mathcal{B}$ must have the same external signature. In the sequel, when we write $\Psi = \Phi[\mathcal{B}/\mathcal{A}]$, we always assume that $\mathcal{B} \notin \Phi$ and $\mathcal{A} \notin \Psi$.

▶ **Definition 53** (Creation corresponding configuration automata)**.** Let X, Y be configuration automata and $\mathcal{A}, \mathcal{B}$ be SIOA. We say that X, Y are creation-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$ iff

1. X never creates $\mathcal{B}$ and Y never creates $\mathcal{A}$.
2. Let $\beta \in traces^*(X) \cap traces^*(Y)$, and let $\alpha \in execs^*(X), \pi \in execs^*(Y)$ be such that $trace_{\mathcal{A}}(\alpha) = trace_{\mathcal{A}}(\pi) = \beta$. Let $x = last(\alpha), y = last(\pi)$, i.e., x, y are the last states along α, π, respectively. Then $\forall a \in \widehat{sig}(X)(x) \cap \widehat{sig}(Y)(y) : created(Y)(y)(a) = created(X)(x)(a)[\mathcal{B}/\mathcal{A}]$.

▶ **Definition 54** (Hiding corresponding configuration automata)**.** Let X, Y be configuration automata and $\mathcal{A}, \mathcal{B}$ be PSIOA. We say that X, Y are hiding-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$ iff

1. X never creates $\mathcal{B}$ and Y never creates $\mathcal{A}$.
2. Let $\beta \in traces^*(X) \cap traces^*(Y)$, and let $\alpha \in execs^*(X), \pi \in execs^*(Y)$ be such that $trace_{\mathcal{A}}(\alpha) = trace_{\mathcal{A}}(\pi) = \beta$. Let $x = last(\alpha), y = last(\pi)$, i.e., x, y are the last states along α, π, respectively. Then $hidden\text{-}actions(Y)(y) = hidden\text{-}actions(X)(x)$.

▶ **Definition 55** ($\mathcal{A}$-fair PCA)**.** Let $\mathcal{A} \in Autids$. Let X be a PCA. We say that X is $\mathcal{A}$-fair if for every states q_X, q'_X, s. t. $config(X)(q_X) \setminus \mathcal{A} = config(X)(q'_X) \setminus \mathcal{A}$, then $created(X)(q_X) = created(X)(q'_X)$ and $hidden\text{-}actions(X)(q_X) = hidden\text{-}actions(X)(q'_X)$.

▶ **Definition 56** ($\mathcal{A}$-conservative PCA)**.** Let X be a PCA, $\mathcal{A} \in Autids$. We say that X is $\mathcal{A}$-conservative if it is $\mathcal{A}$-fair and for every state q_X, $C_x = config(X)(q_X)$ s. t. $\mathcal{A} \in aut(C_X)$ and $map(C_X)(\mathcal{A}) \triangleq q_{\mathcal{A}}$, $hidden\text{-}actions(X)(q_X) = hidden\text{-}actions(X)(q_X) \setminus \widehat{ext}(\mathcal{A})(q_{\mathcal{A}})$.

▶ **Definition 57** (corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$)**.** Let $\mathcal{A}, \mathcal{B} \in Autids$, $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ be PCA we say that $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ are corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$, if they verify:

- $config(X_{\mathcal{A}})(\bar{q}_{X_{\mathcal{A}}}) \lhd_{AB} config(X_{\mathcal{B}})(\bar{q}_{X_{\mathcal{B}}})$.
- X, Y are creation-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$
- X, Y are hiding-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$
- $X_{\mathcal{A}}$ (resp. $X_{\mathcal{B}}$) is a $\mathcal{A}$-conservative (resp. $\mathcal{B}$-conservative) PCA.
- (No creation from $\mathcal{A}$ and $\mathcal{B}$)
  - $\forall q_{X_{\mathcal{A}}}, \in states(X_{\mathcal{A}}), \forall act$ verifying $act \notin sig(config(X_{\mathcal{A}})(q_{X_{\mathcal{A}}}) \backslash \{\mathcal{A}\}) \wedge act \in sig(config(X_{\mathcal{A}})(q_{X_{\mathcal{A}}}))$, $created(X_{\mathcal{A}})(q_{X_{\mathcal{A}}})(act) = \emptyset$ and similarly
  - $\forall q_{X_{\mathcal{B}}}, \in states(X_{\mathcal{B}}), \forall act'$ verifying $act' \notin sig(config(X_{\mathcal{B}})(q_{X_{\mathcal{B}}}) \backslash \{\mathcal{B}\}) \wedge act' \in sig(config(X_{\mathcal{B}})(q_{X_{\mathcal{B}}}))$, $created(X_{\mathcal{B}})(q_{X_{\mathcal{B}}})(act') = \emptyset$

▶ **Theorem 58** (Implementation monotonicity wrt creation/destruction)**.** *Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $X_{\mathcal{A}}$, $X_{\mathcal{B}}$ be PCA corresponding w.r.t. $\mathcal{A}$, $\mathcal{B}$.*

If $\mathcal{A}$ tenaciously implements $\mathcal{B}$ ($\mathcal{A} \leq^{ten} \mathcal{B}$) then $X_{\mathcal{A}}$ tenaciously implements $X_{\mathcal{B}}$ ($X_{\mathcal{A}} \leq^{ten} X_{\mathcal{B}}$).

References

1 Paul C. Attie and Nancy A. Lynch. Dynamic input/output automata: A formal and compositional model for dynamic systems. 249:28–75.

2 Ran Canetti, Ling Cheung, Dilsun Kaynar, Moses Liskov, Nancy Lynch, Olivier Pereira, and Roberto Segala. Task-Structured Probabilistic {I/O} Automata. *Journal of Computer and System Sciences*, 94:63—-97, 2018.

3 Ran Canetti, Ling Cheung, Dilsun Kaynar, Nancy Lynch, and Olivier Pereira. Compositional security for task-PIOAs. *Proceedings - IEEE Computer Security Foundations Symposium*, pages 125–139, 2007.

4 Jing Chen and Silvio Micali. Algorand: A secure and efficient distributed ledger. *Theor. Comput. Sci.*, 777:155–183, 2019.

5 Maurice Herlihy. Blockchains and the future of distributed computing. In Elad Michael Schiller and Alexander A. Schwarzmann, editors, *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, page 155. ACM, 2017.

6 Nancy Lynch, Michael Merritt, William Weihl, and Alan Fekete. A theory of atomic transactions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 326 LNCS:41–71, 1988.

7 Martin L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. Wiley series in probability and mathematical statistics. John Wiley & Sons, 1 edition, 1994.

8 Alejandro Ranchal-Pedrosa and Vincent Gramoli. Platypus: Offchain protocol without synchrony. In Aris Gkoulalas-Divanis, Mirco Marchetti, and Dimiter R. Avresky, editors, *18th IEEE International Symposium on Network Computing and Applications, NCA 2019, Cambridge, MA, USA, September 26-28, 2019*, pages 1–8. IEEE, 2019.

9 Roberto Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusettes Institute of technology, 1995.

Probabilistic Dynamic Input Output Automata

Pierre Civit
Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, F-75005 Paris, France
pierre.civit@lip6.fr

Maria Potop-Butucaru
Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, F-75005 Paris, France
maria.potop-butucaru@lip6.fr

Abstract

We present *probabilistic dynamic I/O automata*, a framework to model dynamic probabilistic systems. Our work extends *dynamic I/O Automata* formalism [1] to probabilistic setting. The original dynamic I/O Automata formalism included operators for parallel composition, action hiding, action renaming, automaton creation, and behavioral sub-typing by means of trace inclusion. They can model mobility by using signature modification. They are also hierarchical: a dynamically changing system of interacting automata is itself modeled as a single automaton. Our work extends to probabilistic settings all these features. Furthermore, we prove necessary and sufficient conditions to obtain the implementation monotonicity with respect to automata creation and destruction. Our work lays down the premises for extending *composable secure-emulation* [3] to dynamic settings, an important tool towards the formal verification of protocols combining probabilistic distributed systems and cryptography in dynamic settings (e.g. blockchains, secure distributed computation, cybersecure distributed protocols etc).

2012 ACM Subject Classification C.2.4 Distributed Systems

Keywords and phrases distributed dynamic systems, probabilistic automata, foundations

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Distributed computing area faces today important challenges coming from modern applications such as cryptocurrencies and blockchains which have a tremendous impact in our society. Blockchains are an evolved form of the distributed computing concept of replicated state machine, in which multiple agents see the evolution of a state machine in a consistent form. At the core of both mechanisms there are distributed computing fundamental elements (e.g. communication primitives and semantics, consensus algorithms, and consistency models) and also sophisticated cryptographic tools. Recently, [5] stated that despite the tremendous interest about blockchains and distributed ledgers, no formal abstraction of these objects has been proposed. In particular it was stated that there is a need for the formalization of the distributed systems that are at the heart of most cryptocurrency implementations, and leverage the decades of experience in the distributed computing community in formal specification when designing and proving various properties of such systems. Therefore, an extremely important aspect of blockchain foundations is a proper model for the entities involved and their potential behavior. The formalisation of blockchain area has to combine models of underlying distributed and cryptographic building blocks under the same hood.

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The formalisation of distributed systems has been pioneered by Lynch and Tuttle [6]. They proposed the formalism of *Input/Output Automata* to model deterministic distributed system. Later, this formalism is extended with Markov decision processes [7] to give *Probabilistic Input/Output Automata* [9] in order to model randomized distributed systems. In this model each process in the system is a automaton with probabilistic transitions. The probabilistic protocol is the parallel composition of the automata modeling each participant. This framework has been further extended in [2] to *task-structured probabilistic Input/Output automata* specifically designed for the analysis of cryptographic protocols. Task-structured probabilistic Input/Output automata are Probabilistic Input/Output automata extended with tasks structures that are equivalence classes on the set of actions. They define the parallel composition for this type of automata. Inspired by the literature in security area they also define the notion of implementation. Informally, the implementation of a Task-structured probabilistic Input/Output automata should look "similar" to the specification whatever the external environment of execution. Furthermore, they provide compositional results for the implementation relation. Even thought the formalism proposed in [2] has been already used in the verification of various cryptographic protocols this formalism does not capture the dynamicity in blockchains systems such as Bitcoin or Ethereum where the set of participants dynamically changes. Moreover, this formalism does not cover blockchain systems where subchains can be created or destroyed at run time [8].

Interestingly, the modelisation of dynamic behavior in distributed systems is an issue that has been addressed even before the born of blockchain systems. The increase of dynamic behavior in various distributed applications such as mobile agents and robots motivated the *Dynamic Input Output Automata* formalism introduced in [1]. This formalisms extends the *Input/Output Automata* formalism with the ability to change their signature dynamically (i.e. the set of actions in which the automaton can participate) and to create other I/O automata or destroy existing I/O automata. The formalism introduced in [1] does not cover the case of probabilistic distributed systems and therefore cannot be used in the verification of blockchains such as Algorand [4].

Our contribution. In order to cope with dynamicity and probabilistic nature of blockchain systems we propose an extension of the formalisms introduced in [2] and [1]. Our extension use a refined definition of probabilistic configuration automata in order to cope with dynamic actions. The main result of our formalism is as follows: the implementation of probabilistic configuration automata is monotonic to automata creation and destruction. Our work is an intermediate step before defining composable secure-emulation [3] in dynamic settings.

Paper organization. The paper is organized as follow. Section 2 is dedicated to a brief introduction of the notion of probabilistic measure an recalls notations used in defining Signature I/O automata of [1]. Section 3 builds on the frameworks proposed in [1] and [2] in order to lay down the preliminaries of our formalism. More specifically, we introduce the definitions of probabilistic signed I/O automata and define their composition and implementation. In Section 4 we extend the definition of configuration automata proposed in [1] to probabilistic configuration automata then we define the composition of probabilistic configuration automata and prove its closeness. The key result of our formalisation, the monotonicity of PSIOA implementations with respect to creation and destruction, is presented in Section 8. This result is based on intermediate results presented in sections 5, 6 and 7.

2 Preliminaries on probability and measure

We assume our reader is comfortable with basic notions of probability theory, such as σ-fields and (discrete) probability measures. A measurable space is denoted by $(S, \mathcal{F}_s)$, where S is a set and $\mathcal{F}_s$ is a σ-algebra over S that is $\mathcal{F}_s \subseteq \mathcal{P}(S)$, is closed under countable union and complementation and its members are called measurable sets ($\mathcal{P}(S)$ denotes the power set of S). A measure over $(S, \mathcal{F}_s)$ is a function $\eta : \mathcal{F}_s \rightarrow \mathbb{R}^{\geq 0}$, such that $\eta(\emptyset) = 0$ and for every countable collection of disjoint sets $\{S_i\}_{i \in I}$ in $\mathcal{F}_s$, $\eta(\bigcup_{i \in I} S_i) = \Sigma_{i \in I} \eta(S_i)$. A probability measure (resp. sub-probability measure) over $(S, \mathcal{F}_s)$ is a measure η such that $\eta(S) = 1$ (resp. $\eta(S) < 1$). A measure space is denoted by $(S, \mathcal{F}_s, \eta)$ where η is a measure on $(S, \mathcal{F}_s)$.

The product measure space $(S_1, \mathcal{F}_{s_1}, \eta_1) \otimes (S_2, \mathcal{F}_{s_2}, \eta_2)$ is the measure space $(S_1 \times S_2, \mathcal{F}_{s_1} \otimes \mathcal{F}_{s_2}, \eta_1 \otimes \eta_2)$, where $\mathcal{F}_{s_1} \otimes \mathcal{F}_{s_2}$ is the smallest σ-algebra generated by sets of the form $\{A \times B | A \in \mathcal{F}_{s_1}, B \in \mathcal{F}_{s_2}\}$ and $\eta_1 \otimes \eta_2$ is the unique measure s. t. for every $C_1 \in \mathcal{F}_{s_1}, C_2 \in \mathcal{F}_{s_2}$, $\eta_1 \otimes \eta_2(C_1 \times C_2) = \eta_1(C_1)\eta_2(C_2)$. If S is countable, we note $\mathcal{P}(S) = 2^S$. If S_1 and S_2 are countable, we note have $2^{S_1} \otimes 2^{S_2} = 2^{S_1 \times S_2}$.

A discrete probability measure on a set S is a probability measure η on $(S, 2^S)$, such that, for each $C \subset S, \eta(S) = \sum_{c \in C} \eta(\{c\})$. We define $Disc(S)$ to be, the set of discrete probability measures on S. In the sequel, we often omit the set notation when we denote the measure of a singleton set. For a discrete probability measure η on a set S, $supp(\eta)$ denotes the support of η, that is, the set of elements $s \in X$ such that $\eta(s) \neq 0$. Given set S and a subset $C \subset S$, the Dirac measure δ_C is the discrete probability measure on S that assigns probability 1 to C. For each element $s \in S$, we note δ_s for $\delta_{\{s\}}$.

If $\{m_i\}_{i \in I}$ is a countable family of measures on $(S, \mathcal{F}_s)$, and $\{p_i\}_{i \in I}$ is a family of non-negative values, then the expression $\sum_{i \in I} p_i m_i$ denotes a measure m on $(S, \mathcal{F}_s)$ such that, for each $C \in \mathcal{F}_s, m(C) = \sum_{i \in I} m_i f_i(C)$. A function $f : X \rightarrow Y$ is said to be measurable from $(X, \mathcal{F}_X) \rightarrow (Y, \mathcal{F}_Y)$ if the inverse image of each element of $\mathcal{F}_Y$ is an element of $\mathcal{F}_X$, that is, for each $C \in \mathcal{F}_Y$, $f^{-1}(C) \in \mathcal{F}_X$. In such a case, given a measure η on $(X, \mathcal{F}_X)$, the function $f(\eta)$ defined on $\mathcal{F}_Y$ by $f(\eta)(C) = \eta(f^{-1}(C))$ for each $C \in Y$ is a measure on $(Y, \mathcal{F}_Y)$ and is called the image measure of η under f.

3 PSIOA

3.1 Action Signature

We use the signature approach from [1].

We assume the existence of a countable set $Autids$ of unique probabilistic signature input/output automata (PSIOA) identifiers, an underlying universal set $Auts$ of PSIOA, and a mapping $aut : Autids \rightarrow Auts$. $aut(\mathcal{A})$ is the PSIOA with identifier $\mathcal{A}$. We use "the automaton $\mathcal{A}$" to mean "the PSIOA with identifier $\mathcal{A}$".. We use the letters $\mathcal{A}, \mathcal{B}$, possibly subscripted or primed, for PSIOA identifiers. The executable actions of a PSIOA $\mathcal{A}$ are drawn from a signature $sig(\mathcal{A})(q) = (in(\mathcal{A})(q), out(\mathcal{A})(q), int(\mathcal{A})(q))$, called the state signature, which is a function of the current state q of $\mathcal{A}$.

$in(\mathcal{A})(q), out(\mathcal{A})(q), int(\mathcal{A})(q)$ are pairwise disjoint sets of input, output, and internal actions, respectively. We define $ext(\mathcal{A})(q)$, the external signature of $\mathcal{A}$ in state q, to be $ext(\mathcal{A})(q) = (in(\mathcal{A})(q), out(\mathcal{A})(q))$.

We define $local(\mathcal{A})(q)$, the local signature of $\mathcal{A}$ in state q, to be $local(\mathcal{A})(q) = (out(\mathcal{A})(q), in(\mathcal{A})(q))$.

For any signature component, generally, the $\widehat{\cdot}$ operator yields the union of sets of actions within the signature, e.g., $\widehat{sig}(\mathcal{A}) : q \in Q \mapsto \widehat{sig}(\mathcal{A})(q) = in(\mathcal{A})(q) \cup out(\mathcal{A})(q) \cup int(\mathcal{A})(q)$. Also define $acts(\mathcal{A}) = \bigcup_{q \in Q} \widehat{sig}(\mathcal{A})(q)$, that is $acts(\mathcal{A})$ is the "universal" set of all actions that A could possibly execute, in any state. In the same way $UI(\mathcal{A}) = \bigcup_{q \in Q} in(\mathcal{A})(q)$, $UO(\mathcal{A}) = \bigcup_{q \in Q} out(\mathcal{A})(q)$, $UH(\mathcal{A}) = \bigcup_{q \in Q} int(\mathcal{A})(q)$, $UL(\mathcal{A}) = \bigcup_{q \in Q} \widehat{local}(\mathcal{A})(q)$, $UE(\mathcal{A}) = \bigcup_{q \in Q} \widehat{ext}(\mathcal{A})(q)$.

3.2 PSIOA

We combine the SIOA of [1] with the PIOA of [9]:

▶ **Definition 1.** A PSIOA $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$, where:

- (a) Q is a countable set of states, $(Q, 2^Q)$ is a measurable space called the state space, and $\bar{q}$ is the start state.
- (b) $sig(\mathcal{A}) : q \in Q \mapsto sig(\mathcal{A})(q) = (in(A)(q), out(A)(q), int(A)(q)$ is the signature function that maps each state to a triplet of countable input, output and internal set of actions.
- (d) $D \subset Q \times acts(\mathcal{A}) \times Disc(Q)$ is the set of probabilistic discrete transitions where $\forall (q, a, \eta) \in D : a \in \widehat{sig}(\mathcal{A})(q)$. If (q, a, η) is an element of D, we write $q \xrightarrow{a} \eta$ and action a is said to be *enabled* at q. The set of states in which action a is enabled is denoted by E_a. For $B \subseteq A$, we define E_B to be $\bigcup_{a \in B} E_a$. The set of actions enabled at q is denoted by $enabled(q)$. If a single action $a \in B$ is enabled at q and $q \xrightarrow{a} \eta$, then this η is denoted by $\eta_{(\mathcal{A},q,B)}$. If B is a singleton set $\{a\}$ then we drop the set notation and write $\eta_{(\mathcal{A},q,a)}$.

In addition $\mathcal{A}$ must satisfy the following conditions:

- $\mathbf{E_1}$ (Input action enabling) $\forall \mathbf{x} \in Q : \forall a \in in(\mathcal{A})(q), \exists \eta \in Disc(Q) : (q, a, \eta) \in D$.
- $\mathbf{T_1}$ Transition determinism: For every $q \in Q$ and $a \in A$ there is at most one $\eta \in Disc(Q)$ such that $(q, a, \eta) \in D$.

Notation

For every PSIOA $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$, we note $states(\mathcal{A}) = Q$, $start(\mathcal{A}) = \bar{q}$, $steps(\mathcal{A}) = D$.

3.3 Execution, Trace

We use the classic notions of execution and trace from [9].

▶ **Definition 2** (fragment, execution and trace of PSIOA)**.** An *execution fragment* of a PSIOA $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$ is a finite or infinite sequence $\alpha = q_0 a_1 q_1 a_2 ...$ of alternating states and actions, such that:

1. If α is finite, it ends with a state.
2. For every non-final state q_i, there is $\eta \in Disc(Q)$ and a transition $(q_i, a_{i+1}, \eta) \in D$ s. t. $q_{i+1} \in supp(\eta)$.

We write $fstate(\alpha)$ for q_0 (the first state of α), and if α is finite, we write $lstate(\alpha)$ for its last state. We use $Frags(\mathcal{A})$ (resp., $Frags^{Í*}(\mathcal{A})$) to denote the set of all (resp., all finite) execution fragments of $\mathcal{A}$. An *execution* of $\mathcal{A}$ is an execution fragment α with $fstate(\alpha) = \bar{q}$. $Execs(\mathcal{A})$ (resp., $Execs^*(\mathcal{A})$) denotes the set of all (resp., all finite) executions of $\mathcal{A}$. The *trace* of an execution fragment α, written $trace(\alpha)$, is the restriction of α to the external actions of $\mathcal{A}$. We say that β is a trace of $\mathcal{A}$ if there is $\alpha \in Execs(P)$ with $\beta = trace(\alpha)$. $Traces(\mathcal{A})$ (resp., $Traces^*(\mathcal{A})$) denotes the set of all (resp., all finite) traces of $\mathcal{A}$.

▶ **Definition 3** (reachable execution). Let $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$ be a PSIOA. A state q is said *reachable* if it exists a finite execution that ends with q.

3.4 Compatibility and composition

Tha main aim of IO formalism is to compose several automata $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ and obtain some guarantees of the system by composition of the guarantees of the different elements of the system. Some syntaxic rules have to be satisfied before defining the composition operation.

▶ **Definition 4** (Compatible signatures). Let S be a set of signatures. Then S is compatible iff, $\forall sig, sig' \in S$, where $sig = (in, out, int), sig' = (in', out', int')$ and $sig \neq sig'$, we have: 1. $(in \cup out \cup int) \cap int' = \emptyset$, and 2. $out \cap out' = \emptyset$.

▶ **Definition 5** (Composition of Signatures). Let $\Sigma = (in, out, int)$ and $\Sigma' = (in', out', int')$ be compatible signatures. Then we define their composition $\Sigma \times \Sigma = (in \cup in' - (out \cup out'), out \cup out', int \cup int')$.

Signature composition is clearly commutative and associative.

▶ **Definition 6** (partially compatible at a state). Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a set of PSIOA. A *state* of $\mathbf{A}$ is an element $q = (q_1, ..., q_n) \in Q = Q_1 \times ... \times Q_n$. We say $\mathcal{A}_1, ..., \mathcal{A}_n$ are *partially-compatible* at state q (or $\mathbf{A}$ is) if $\{sig(\mathcal{A}_1)(q_1), ..., sig(\mathcal{A}_n)(q_n)\}$ is a set of compatible signatures. In this case we note $sig(\mathbf{A})(q) = sig(\mathcal{A}_1)(q_1) \times ... \times sig(\mathcal{A}_n)(q_n)$ and we note $\eta_{(\mathbf{A},q,a)} \in Disc(Q)$, s. t. for every action $a \in \widehat{sig}(\mathbf{A})(q)$, $\eta_{(\mathbf{A},q,a)} = \eta_1 \otimes ... \otimes \eta_n \in Disc(Q)$ that verifies for every $j \in [1, n]$:

- If $a \in sig(\mathcal{A}_j)(q_j)$, $\eta_j = \eta_{(\mathcal{A}_j,q_j,a)}$.
- Otherwise, $\eta_j = \delta_{q_j}$

while $\eta_{(\mathbf{A},q,a)} = \delta_q$ if $a \notin \widehat{sig}(\mathbf{A})(q)$.

▶ **Definition 7** (pseudo execution). Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a set of PSIOA. A *pseudo execution fragment* of $\mathbf{A}$ is a finite or infinite sequence $\alpha = q^0 a^1 q^1 a^2 ...$ of alternating states of $\mathbf{A}$ and actions, such that:

- If α is finite, it ends with a n-uplet of state.
- For every non final state q^i, $\mathbf{A}$ is partially-compatible at q^i.
- For every action a^i, $a^i \in \widehat{sig}(\mathbf{A})(q^{i-1})$.
- For every state q^i, with $i > 0$, $q^i \in supp(\eta_{(\mathbf{A},q^{i-1},a^i)})$.

A *pseudo execution* of $\mathbf{A}$ is a pseudo execution fragment of $\mathbf{A}$ with $q^0 = (\bar{q}_{\mathcal{A}_1}, ..., \bar{q}_{\mathcal{A}_n})$.

▶ **Definition 8** (reachable state). Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a set of PSIOA. A state q of $\mathbf{A}$ is *reachable* if it exists a pseudo execution α of $\mathbf{A}$ ending on state q.

▶ **Definition 9** (partially-compatible PSIOA). Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a set of PSIOA. The automata $\mathcal{A}_1, ..., \mathcal{A}_n$ are *ℓ-partially-compatible* with $\ell \in \mathbb{N}$ if no pseudo-execution α of $\mathbf{A}$ with $|\alpha| \leq \ell$ ends on non-partially-compatible state q. The automata $\mathcal{A}_1, ..., \mathcal{A}_n$ are *partially-compatible* if $\mathbf{A}$ is partially-compatible at each reachable state q, i. e. $\mathbf{A}$ is *ℓ-partially-compatible* for every $\ell \in \mathbb{N}$.

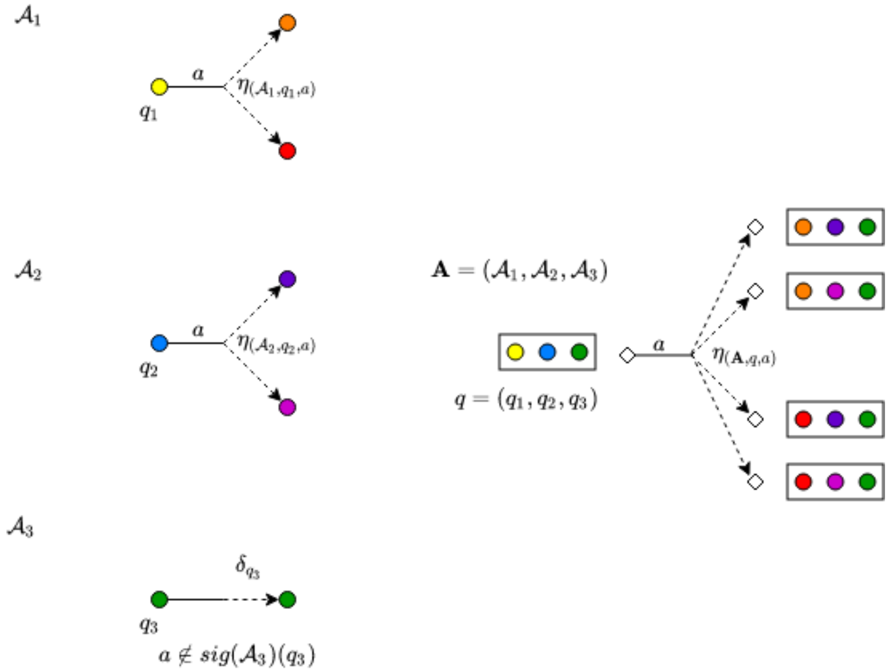


Figure 1 The family transition is obtain by the transitions of the automata of the family.

▶ **Definition 10** (Compatible PSIOA)**.** Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a set of PSIOA with $\mathcal{A}_i = ((Q_i, \mathcal{F}_{Q_i}), sig(\mathcal{A}_i), D_i)$. We say $\mathbf{A}$ is compatible if it is partially-compatible for every state $q = (q_1, ..., q_n) \in Q_1 \times ... \times Q_n$.

Of course a set of compatible PSIOA is also a set of partially-compatible automata. The latter allows us to extend the formalism of [1] which will be useful later.

▶ **Definition 11** (PSIOAs composition)**.** If $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ is a compatible set of PSIOAs, with $\mathcal{A}_i = (Q_i, \bar{q}_i, sig(\mathcal{A}_i), D_i)$, then their composition $\mathcal{A}_1||...||\mathcal{A}_n$, is defined to be $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$, where:

- $Q = Q_1 \times ... \times Q_n$
- $\bar{q} = (\bar{q}_1, ..., \bar{q}_n)$
- $sig(\mathcal{A}) : q = (q_1, ..., q_n) \in Q \mapsto sig(\mathcal{A})(q) = sig(\mathcal{A}_1)(q_1) \times ... \times sig(\mathcal{A}_n)(q_n)$. ,
- $D \subset Q \times A \times Disc(Q)$ is the set of triples $(q, a, \eta_{(\mathbf{A},q,a)})$ so that $q \in Q$ and $a \in \widehat{sig}(\mathbf{A})(q)$

▶ **Definition 12** (partially-compatible PSIOA composition)**.** If $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ is a partially-compatible set of PSIOA, with $\mathcal{A}_i = ((Q_i, \mathcal{F}_{Q_i}), sig(\mathcal{A}_i), D_i)$, then their partial-composition $\mathcal{A}_1||...||\mathcal{A}_n$, is defined to be $\mathcal{A} = ((Q, \mathcal{F}_Q), sig(\mathcal{A}), D)$, where:

- $Q = \{q \in Q_1 \times ... \times Q_n | q$ is a reachable state of $\mathbf{A}\}$.
- $\bar{q} = (\bar{q}_1, ..., \bar{q}_n)$
- $sig(\mathcal{A}) : q = (q_1, ..., q_n) \in Q \mapsto sig(\mathcal{A})(q) = sig(\mathcal{A}_1)(q_1) \times ... \times sig(\mathcal{A}_n)(q_n)$.
- $D \subset Q \times A \times Disc(Q)$ is the set of triples $(q, a, \eta_{(\mathbf{A},q,a)})$ so that $q \in Q$ and $a \in \widehat{sig}(\mathbf{A})(q)$

3.5 Measure for executions and traces

To solve the non-determinism we use schedule that allows us to chose an action in a signature. To do so, we adapt the definition of task of [2] to the dynamic setting. We assume the existence of a subset $Autids_0 \subset Autids$ that represents the "atomic ententies" that will constitute the configuration automata introduced in the next section.

▶ **Definition 13** (Constitution). For every $\mathcal{A} \in Autids$, we note

$$constitution(\mathcal{A}) : \begin{cases} states(\mathcal{A}) & \rightarrow & \mathcal{P}(Autids_0) = 2^{Autids_0} \\ q & \mapsto & constitution(\mathcal{A})(q) \end{cases}$$

For every $\mathcal{A} \in Autids_0$, for every $q \in states(\mathcal{A})$, $constitution(\mathcal{A})(q) = \{\mathcal{A}\}$.

For every $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n) \in (Autids_0)^n$, $\mathcal{A} = \mathcal{A}_1||...||\mathcal{A}_n$ for every $q \in states(\mathcal{A})$, $constitution(\mathcal{A})(q) = \mathbf{A}$.

In the next section we will define the constitution mapping for a new kind of automata, with a "dynamic" constitution that can change from one state to another one.

▶ **Definition 14** (Task). A task T is a pair $(id, actions)$ where $id \in Autids_0$ and $actions$ is a set of action labels. Let $T = (id, actions)$, we note $id(T) = id$ and $actions(T) = actions$.

▶ **Definition 15** (Enabled task). Let $\mathcal{A} \in Autids$. A task T is said *enabled* in state $q \in states(\mathcal{A})$ if :

- $id(T) \in constitution(\mathcal{A})(q)$
- It exists a unique local action $a \in \widehat{loc}(\mathcal{A})(q) \cap actions(T)$ (noted $a \in T$ to simplify) enabled at state q (that is it exists $\eta \in Disc(Q)$ s. t. $(q, a, \eta) \in D$.

In this case we say that a is *triggered* by T at state q.

We are not dealing with a schedule *of a specific automaton* anymore, which differs from [2]. However the restriction of our definition to "static" setting matches their definition.

▶ **Definition 16** (schedule). A schedule ρ is a (finite or infinite) sequence of tasks.

We use the measure of [2].

▶ **Definition 17.** Let $\mathcal{A}$ be a PSIOA. Given $\mu \in Disc(Frags(\mathcal{A}))$ a discrete probability measure on the execution fragments and a task schedule ρ, $apply(\mu, \rho)$ is a probability measure on $Frags(\mathcal{A})$. It is defined recursively as follows.

1. $apply_{\mathcal{A}}(\mu, \lambda) := \mu$. Here λ denotes the empty sequence.
2. For every T and $\alpha \in Frags^*(\mathcal{A})$, $apply(\mu, T)(\alpha) := p_1(\alpha) + p_2(\alpha)$, where:
   - $p_1(\alpha) = \begin{cases} \mu(\alpha')\eta_{(\mathcal{A},q',a)}(q) & \text{if } \alpha = \alpha' a q, q' = lstate(\alpha') \text{ and } a \text{ is triggered by } T \\ 0 & \text{otherwise} \end{cases}$
   - $p_2(\alpha) = \begin{cases} \mu(\alpha) & \text{if } T \text{ is not enabled after } \alpha \\ 0 & \text{otherwise} \end{cases}$
3. 3. If ρ is finite and of the form $\rho' T$, then $apply_{\mathcal{A}}(\mu, \rho) := apply_{\mathcal{A}}(apply_{\mathcal{A}}(\mu, \rho'), T)$.
4. 4. If ρ is infinite, let ρ_i denote the length-i prefix of ρ and let pm_i be $apply_{\mathcal{A}}(\mu, \rho_i)$. Then $apply_{\mathcal{A}}(\mu, \rho) := \lim_{i \rightarrow \infty} pm_i$.

$tdist_{\mathcal{A}}(\mu, \rho) : Traces_{\mathcal{A}} \rightarrow [0, 1]$, is defined as $tdist_{\mathcal{A}}(\mu, \rho)(E) = apply(\delta_{\bar{q}}, \rho)(trace_{\mathcal{A}}^{-1}(E))$, for any measurable set $E \in \mathcal{F}_{Traces_{\mathcal{A}}}$.

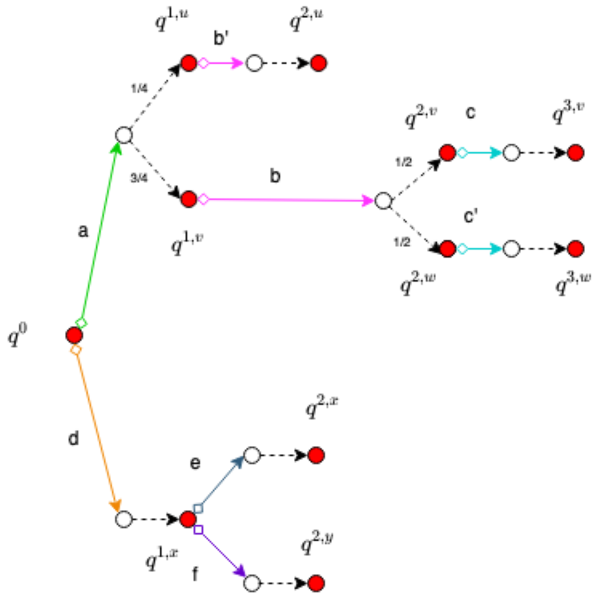


Figure 2 Non-deterministic execution: The scheduler allows us to solve the non-determinism, by triggering an action among the enabled one. We give an example with an automaton $\mathcal{A} = (Q, \bar{q} = q_0, sig(\mathcal{A}), D_{\mathcal{A}})$ and the tasks T_g, T_o, T_p, T_b (for green, orange, pink, blue) with the respective actions $\{a\}, \{d\}, \{b, b'\}, \{c, c'\}$, and the tasks T_{go}, T_{bo} with the respective actions $\{a, d\}, \{c, c', d\}$. At state q_0, $sig(\mathcal{A})(q_0) = (\emptyset, \{a\}, \{d\})$. Hence both a and d are enabled local action at q_0, which means both T_g and T_o are enabled at state q_0, but T_{go} is not enabled at state q_0 since it does not solve the non-determinism (a and d are enabled local action at q_0). At state q_1, T_p is enabled but neither T_o or T_b. We give some results: $apply(\delta_{q^0}, T_g)(q^0, a, q^{1,v}) = 1$
$apply(\delta_{q^0}, T_g T_p)(q^0, a, q^{1,v}, b, q^{2,w}) = apply(apply(\delta_{q^0}, T_g), T_p)(q^0, a, q^{1,v}, b, q^{2,w}) = 1/2$
$apply(\delta_{q^0}, T_g T_p T_b)(q^0, a, q^{1,v}, b, q^{2,w}, c, q^{3,w}) = apply(apply(\delta_{q^0}, T_g T_p), T_b)(q^0, a, q^{1,v}, b, q^{2,w}, c, q^{3,w}) = 3/8$
$apply(\delta_{q^0}, T_g T_p T_o T_b)(q^0, a, q^{1,v}, b, q^{2,w}, c, q^{3,w}) = 3/8$, since T_o is not enabled at state $q^{2,w}$.

We write $tdist_{\mathcal{A}}(\mu, \rho)$ as shorthand for $tdist_{\mathcal{A}}(apply_{\mathcal{A}}(\mu, \rho))$ and $tdist_{\mathcal{A}}(\rho)$ for $tdist_{\mathcal{A}}(apply_{\mathcal{A}}(\delta(\bar{x}), \rho))$, where $\delta(\bar{x})$ denotes the measure that assigns probability 1 to $\bar{x}$. A trace distribution of $\mathcal{A}$ is any $tdist_{\mathcal{A}}(\rho)$. We use $Tdists_{\mathcal{A}}$ to denote the set $\{tdist_{\mathcal{A}}(\rho) : \rho$ is a task schedule $\}$.

We removed the subscript $\mathcal{A}$ when this is clear in the context.

3.6 Implementation

▶ **Definition 18** (Environment). A probabilistic environment for PSIOA $\mathcal{A}$ is a PSIOA $\mathcal{E}$ such that $\mathcal{A}$ and $\mathcal{E}$ are partially-compatible.

▶ **Definition 19** (External behavior). The external behavior of a PSIOA $\mathcal{A}$, written as $ExtBeh_{\mathcal{A}}$, is defined as a function that maps each environment $\mathcal{E}$ for $\mathcal{A}$ to the set of trace distributions $Tdists_{\mathcal{A}||\mathcal{E}}$.

▶ **Definition 20** (Comparable PSIOA). Two PSIOA $\mathcal{A}_1$ and $\mathcal{A}_2$ are comparable if $UI(\mathcal{A}_1) = UI(\mathcal{A}_2)$ and $UO(\mathcal{A}_1) = UO(\mathcal{A}_2)$.

▶ **Definition 21.** If $\mathcal{A}_1$ and $\mathcal{A}_2$ are comparable then $\mathcal{A}_1$ is said to implement $\mathcal{A}_2$, written as $\mathcal{A}_1 \leq \mathcal{A}_2$ if, for every environment $\mathcal{E}$ for both $\mathcal{A}_1$ and $\mathcal{A}_2$, $ExtBeh_{\mathcal{A}_1}(\mathcal{E}) \subseteq ExtBeh_{\mathcal{A}_2}(\mathcal{E})$.

This definition of implementation as a functional map from environment automata gives us the desired compositionality result for task-PSIOAs.

▶ **Theorem 22.** *Suppose $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{B}$ are PSIOAs, where $\mathcal{A}_1$, $\mathcal{A}_2$ are comparable and $\mathcal{A}_1 \leq \mathcal{A}_2$. If $\mathcal{B}$ is compatible with $\mathcal{A}_1$, $\mathcal{A}_2$ then $\mathcal{A}_1||\mathcal{B} \leq \mathcal{A}_2||\mathcal{B}$.*

Proof. Immediate with the associativity of the parallel composition. Indeed, if $\mathcal{E}$ is an environment for both $\mathcal{A}_1||\mathcal{B}$ and $\mathcal{A}_2||\mathcal{B}$, then $\mathcal{E}' = \mathcal{B}||\mathcal{E}$ is an environment for both $\mathcal{A}_1$ and $\mathcal{A}_2$. Since $\mathcal{A}_1 \leq \mathcal{A}_2$, for any schedule ρ, it exists a corresponding schedule ρ', s. t. $tdist_{\mathcal{A}_1||\mathcal{E}'}(\rho) = tdist_{\mathcal{A}_2||\mathcal{E}'}(\rho)$. Thus, for any schedule ρ, it exists a corresponding schedule ρ' s. t. $tdist_{\mathcal{A}_1||\mathcal{B}||\mathcal{E}}(\rho) = tdist_{\mathcal{A}_2||\mathcal{B}||\mathcal{E}}(\rho)$, that is $\mathcal{A}_1||\mathcal{B} \leq \mathcal{A}_2||\mathcal{B}$. ◀

3.7 Hiding operator

We anticipate the definition of configuration automata by introducing the classic hiding operator.

▶ **Definition 23** (hiding on signature)**.** Let $sig = (in, out, int)$ be a signature and $\underline{acts}$ a set of actions. We note $hide(sig, \underline{acts})$ the signature $sig' = (in', out', int')$ s. t.

- $in' = in$
- $out' = out \setminus \underline{acts}$
- $int' = int \cup (out \cap \underline{acts})$

▶ **Definition 24** (hiding on PSIOA)**.** Let $\mathcal{A} = (Q, \bar{q}, sig(\mathcal{A}), D)$ be a PSIOA. Let $hiding\text{-}actions$ a function mapping each state $q \in Q$ to a set of actions. We note $hide(\mathcal{A}, hiding\text{-}actions)$ the PSIOA $(Q, \bar{q}, sig'(\mathcal{A}), D)$, where $sig'(\mathcal{A}) : q \in Q \mapsto hide(sig(\mathcal{A})(q), hiding\text{-}actions(q))$.

▶ **Lemma 25** (hiding and composition are commutative)**.** *Let $sig_a = (in_a, out_a, int_a)$, $sig_b = (in_b, out_b, int_b)$ be compatible signature and $\underline{acts}_a$, $\underline{acts}_b$ some set of actions, s. t. $(\underline{acts}_a \cap out_a) \cap \widehat{sig}_b = \emptyset$ and $(\underline{acts}_b \cap out_b) \cap \widehat{sig}_b = \emptyset$, then $sig'_a \triangleq hide(sig, \underline{act}_a) \triangleq (in'_a, out'_a, int'_a)$ and $sig'_b \triangleq hide(sig_b, \underline{act}_b) \triangleq (in'_b, out'_b, int'_b)$ are compatible. Furthermore, if $out_b \cap \underline{acts}_a = \emptyset$,and $out_a \cap \underline{acts}_b = \emptyset$ then $sig'_a \times sig'_b = hide(sig_a \times sig_b, \underline{act}_a \cup \underline{act}_b)$.*

Proof.
- compatibility: After hiding operation, we have:
  - $in'_a = in_a$, $in'_b = in_b$
  - $out'_a = out_a \setminus \underline{acts}_a$, $out'_b = out_b \setminus \underline{acts}_b$
  - $int'_a = int_a \cup (out_a \cap \underline{acts}_a)$, $int'_b = int_b \cup (out_b \cap \underline{acts}_b)$

  Since $out_a \cap out_b = \emptyset$, a fortiori $out'_a \cap out'_b = \emptyset$. $int_a \cap \widehat{sig}_b = \emptyset$, thus if $(out_a \cap \underline{acts}_a) \cap \widehat{sig}_b = \emptyset$, then $int'_a \cap \widehat{sig}_b = \emptyset$ and with the symetric argument, $int'_b \cap \widehat{sig}_a = \emptyset$. Hence, sig'_a and sig'_b are compatible.
- commutativity:

  After composition of $sig'_c = sig'_a \times sig'_b$ operation, we have:
  - $out'_c = out'_a \cup out'_b = (out_a \setminus \underline{acts}_a) \cup (out_b \setminus \underline{acts}_b)$. If $out_b \cap \underline{acts}_a = \emptyset$ and $out_a \cap \underline{acts}_b = \emptyset$, then $out'_c = (out_a \cup out_b) \setminus (\underline{acts}_a \cup \underline{acts}_b)$.
  - $in'_c = in'_a \cup in'_b \setminus out'_c = in_a \cup in_b \setminus out'_c$
  - $int'_c = int'_a \cup int'_b = int_a \cup (out_a \cap \underline{acts}_a) int_b \cup (out_b \cap \underline{acts}_b) = int_a \cup int_b \cup (out_a \cap \underline{acts}_a) \cup (out_b \cap \underline{acts}_b)$. If $out_b \cap \underline{acts}_a = \emptyset$ and $out_a \cap \underline{acts}_b = \emptyset$, then $int'_c = int_a \cup int_b \cup ((out_a \cup out_b) \cap (\underline{acts}_a \cup \underline{acts}_b)$.

  and after composition of $sig_d = sig_a \times sig_b$

- $out_d = out_a \cup out_b$
- $in_d = in_a \cup in_b \setminus out_d$
- $int_d = int_a \cup int_b$

Finally, after hiding operation $sig'_d = hide(sig_d, \underline{acts}_a \cup \underline{acts}_b)$ we have :

- $in'_d = in_d$
- $out'_d = out_d \setminus \underline{acts}_a \cup \underline{acts}_b = (out_a \cup out_b) \setminus (\underline{acts}_a \cup \underline{acts}_b)$
- $int'_d = int_d \cup (out_d \cap (\underline{acts}_a \cup \underline{acts}_b)) = (int_a \cup int_b) \cup (out_d \cap (\underline{acts}_a \cup \underline{acts}_b))$

Thus, if $out_b \cap \underline{acts}_a = \emptyset$ and $out_a \cap \underline{acts}_b = \emptyset$

- $in'_d = in'_c$
- $out'_d = out'_c$
- $int'_d = int'_c$

◀

▶ Remark. We can restrict hiding operation to set of actions include in the set of output actions of the signature ($\underline{act} \subseteq out$). In this case, since we alreay have $out_a \cap out_b = \emptyset$ by compatibility, we immediatly have $out_a \cap \underline{acts}_b = \emptyset$ and $out_b \cap \underline{acts}_a = \emptyset$. Thus to obtain compatibility, we only need $in_b \cap \underline{acts}_a = \emptyset$ and $in_a \cap \underline{acts}_b = \emptyset$. Later, the compatibility of PCA will implicitly assume this predicate (otherwise the PCA could not be compatible).

3.8 State renaming operator

We anticipate the definition of isomorphism between PSIOA that differs only syntactically.

▶ **Definition 26.** (State renaming for PSIOA) Let $\mathcal{A}$ be a PSIOA with $Q_{\mathcal{A}}$ as set of states, let $Q_{\mathcal{A}'}$ be another set of states and let $ren : Q_{\mathcal{A}} \rightarrow Q_{\mathcal{A}'}$ be a bijective mapping. Then $ren(\mathcal{A})$ is the automaton given by:

- $start(ren(\mathcal{A})) = ren(start(Q_{\mathcal{A}}))$
- $states(ren(\mathcal{A})) = ren(states(Q_{\mathcal{A}}))$
- $\forall q_{\mathcal{A}'} \in states(ren(\mathcal{A})), sig(ren(\mathcal{A}))(q_{\mathcal{A}'}) = sig(\mathcal{A})(ren^{-1}(q_{\mathcal{A}'}))$
- $\forall q_{\mathcal{A}'} \in states(ren(\mathcal{A})), \forall a \in sig(ren(\mathcal{A}))(q_{\mathcal{A}'})$, if $(ren^{-1}(q_{\mathcal{A}'}), a, \eta) \in D_{\mathcal{A}}$, then $(q_{\mathcal{A}'}, a, \eta') \in D_{ren(\mathcal{A})}$ where $\eta' \in Disc(Q_{\mathcal{A}'}, \mathcal{F}_{Q_{\mathcal{A}'}})$ and for every $q_{\mathcal{A}''} \in states(ren(\mathcal{A}))$, $\eta'(q_{\mathcal{A}''}) = \eta(ren^{-1}(q_{\mathcal{A}''}))$.

▶ **Definition 27.** (State renaming for PSIOA execution) Let $\mathcal{A}$ and $\mathcal{A}'$ be two PSIOA s. t. $\mathcal{A}' = ren(\mathcal{A}')$. Let $\alpha = q^0 a^1 q^1 ...$ be an execution fragment of $\mathcal{A}$. We note $ren(\alpha)$ the sequence $ren(q^0)a^1 ren(q^1)....$

▶ **Lemma 28.** *Let $\mathcal{A}$ and $\mathcal{A}'$ be two PSIOA s. t. $\mathcal{A}' = ren(\mathcal{A}')$. Let α be an execution fragment of $\mathcal{A}$. The sequence $ren(\alpha)$ is an execution fragment of $\mathcal{A}$.*

Proof. Let $q^j a^{j+1} q^{j+1}$ be a subsequence of α. $ren(q^j) \in states(\mathcal{A}')$ by definition, $a^j \in sig(\mathcal{A}')(ren(q^j))$ since $sig(\mathcal{A}')(ren(q^j)) = sig(\mathcal{A})(q^j)$, and $\eta_{(\mathcal{A}', ren(q^j), a^{j+1})}(ren(q^{j+1})) = \eta_{(\mathcal{A}, q^j, a^{j+1})}(q^{j+1}) > 0$. ◀

4 Probabilistic Configuration Automata

We combine the notion of configuration of [1] with the probabilistic setting of [9].

4.1 configuration

▶ **Definition 29** (Configuration). A configuration is a pair $(\mathbf{A}, \mathbf{S})$ where

- $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ is a finite sequence of PSIOA identifiers (lexicographically ordered [1]), and
- $\mathbf{S}$ maps each $\mathcal{A}_k \in \mathbf{A}$ to an $s_k \in states(\mathcal{A}_k)$.

In distributed computing, configuration usually refers to the union of states of **all** the automata of the system. Here, the notion is different, it captures a set of some automata $(\mathbf{A})$ in their current state $(\mathbf{S})$.

▶ **Definition 30** (Compatible configuration). A configuration $(\mathbf{A}, \mathbf{S})$ is compatible iff, for all $\mathcal{A}, \mathcal{B} \in \mathbf{A}$, $\mathcal{A} \neq \mathcal{B}$: 1. $sig(\mathcal{A})(\mathbf{S}(\mathcal{A})) \cap int(\mathcal{B})(\mathbf{S}(\mathcal{B})) = \emptyset$, and 2. $out(\mathcal{A})(\mathbf{S}(\mathcal{A})) \cap out(\mathcal{B})(\mathbf{S}(\mathcal{B})) = \emptyset$

▶ **Definition 31** (Intrinsic attributes of a configuration). Let $C = (\mathbf{A}, \mathbf{S})$ be a compatible task-configuration. Then we define

- $auts(C) = \mathbf{A}$ represents the automata of the configuration,
- $map(C) = \mathbf{S}$ maps each automaton of the configuration with its current state,
- $out(C) = \bigcup_{\mathcal{A} \in \mathbf{A}} out(\mathcal{A})(\mathbf{S}(\mathcal{A}))$ represents the output action of the configuration,
- $in(C) = (\bigcup_{\mathcal{A} \in \mathbf{A}} in(\mathcal{A})(\mathbf{S}(\mathcal{A}))) - out(C)$ represents the input action of the configuration,
- $int(C) = \bigcup_{\mathcal{A} \in \mathbf{A}} int(\mathcal{A})(\mathbf{S}(\mathcal{A}))$ represents the internal action of the configuration,
- $ext(C) = in(C) \cup out(C)$ represents the external action of the configuration,
- $sig(C) = (in(C), out(C), int(C))$ is called the intrinsic signature of the configuration,
- $CA(C) = (aut(\mathcal{A}_1)||...||aut(\mathcal{A}_n))$ represents the composition of all the automata of the configuration,
- $US(C) = (\mathbf{S}(\mathcal{A}_1), ..., \mathbf{S}(\mathcal{A}_n))$ represents the states of the automaton corresponding to the composition of all the automata of the configuration,

Here we define a reduced configuration as a configuration deprived of the automata that are in the very particular state where their current signatures are the empty set. This mechanism will allows us to capture the idea of destruction.

▶ **Definition 32** (Reduced configuration). $reduce(C) = (\mathbf{A}', \mathbf{S}')$, where $\mathbf{A}' = \{\mathcal{A} | \mathcal{A} \in \mathbf{A} \text{ and } sig(\mathcal{A})(\mathbf{S}(\mathcal{A})) \neq \emptyset\}$ and $\mathbf{S}'$ is the restriction of $\mathbf{S}$ to $\mathbf{A}'$, noted $\mathbf{S} \upharpoonright \mathbf{A}'$ in the remaining.

A configuration C is a reduced configuration iff $C = reduce(C)$.

We recall that we assume the existence of a countable set $Autids$ of unique PSIOA identifiers, an underlying universal set $Auts$ of $PSIOA$, and a mapping $aut : Autids \rightarrow Auts$. $aut(\mathcal{A})$ is the $PSIOA$ with identifier $\mathcal{A}$. We will define a measurable space for configuration. We note for every $\varphi \in \mathcal{P}(Autids)$, $Q_\varphi = Q_{\varphi_1} \times ... \times Q_{\varphi_n}$ and $\mathcal{F}_{Q_\varphi} = \mathcal{F}_{Q_{\varphi_1}} \otimes ... \otimes \mathcal{F}_{Q_{\varphi_{|\varphi|}}}$

We note $Q_{aut} = \bigcup_{\varphi \in \mathcal{P}(Autids)} Q_\varphi$, the set of all possible state sets cartesian product for each possible family of automata. $\mathcal{F}_{Q_{aut}} = \{\bigcup_{i \in [1,k]} c_i | \phi \in \mathcal{P}(\mathcal{P}(Autids)), c_i \in \mathcal{F}_{Q_{\varphi_i}} \phi = \varphi_1, ..., \varphi_k, \varphi_i \in \mathcal{P}(Autids)\}$ $(Q_{aut}, \mathcal{F}_{Q_{aut}})$ is a measurable space.

We note $Q_{conf} = \{(\mathbf{A}, \mathbf{S}) | \mathbf{A} \in \mathcal{P}(Autids), \forall \mathcal{A}_i \in \mathbf{A}, \mathbf{S}(\mathcal{A}_i) \in Q_i\}$, the set of all possible configurations.

[1] lexicographic order will simplify projection on product of probabilistic measure for transition of composition of automata

Let $f = \left\{ \begin{array}{lll} Q_{conf} & \rightarrow & Q_{aut} \\ (\mathbf{A}, \mathbf{S}) & \mapsto & Q_{CA((\mathbf{A},\mathbf{S}))} = \mathbf{S}(\mathcal{A}_1) \times ... \times \mathbf{S}(\mathcal{A}_n) \end{array} \right.$

We note $\mathcal{F}_{Q_{conf}} = \{f^{-1}(P) | P \in \mathcal{F}_{Q_{aut}}\}$.

$(Q_{conf}, \mathcal{F}_{Q_{conf}})$ is a measurable space

4.2 Configuration transition

We will define some probabilistic transition from configurations to others. where some automata can be destroyed or created. To define it properly, we start by defining "preserving transition" where no automaton is neither created nor destroyed and then we define above this definition the notion of configuration transition.

▶ **Definition 33** (Preserving distribution)**.** A *preserving distribution* $\eta_p \in Disc(Q_{conf})$ is a distribution verifying $\forall (\mathbf{A}, \mathbf{S}), (\mathbf{A}', \mathbf{S}') \in supp(\eta_p), \mathbf{A} = \mathbf{A}'$. The unique family of automata ids $\mathbf{A}$ of the configurations in the support of η_p is called the *family support* of η_p.

We define a companion distribution as the natural distribution of the corresponding family of automata at the corresponding current state. Since no creation or destruction occurs, these definitions can seem redundant, but this is only an intermediate step to define properly the "dynamic" distribution.

▶ **Definition 34** (Companion distribution)**.** Let $C = (\mathbf{A}, \mathbf{S})$ be a compatible configuration with $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ and $\mathbf{S} : \mathcal{A}_i \in \mathbf{A} \mapsto q_i \in Q_{\mathcal{A}_i}$ (with $\mathbf{A}$ partially-compatible at state $q = (q_1, ..., q_n) \in Q_{\mathbf{A}} = Q_{\mathcal{A}_1} \times ... \times Q_{\mathcal{A}_n}$). Let η_p be a preserving distribution with $\mathbf{A}$ as family support. The probabilistic distribution $\eta_{(\mathbf{A},q,a)}$ is a *companion distribution* of η_p if for every $q' = (q'_1, ..., q'_n) \in Q_{\mathbf{A}}$, for every $\mathbf{S}'' : \mathcal{A}_i \in \mathbf{A} \mapsto q''_i \in Q_{\mathcal{A}_i}$,

$$\eta_{(\mathbf{A},q,a)}(q') = \eta_p((\mathbf{A}, \mathbf{S}'')) \iff \forall i \in [1, n], q''_i = q'_i,$$

that is the distribution $\eta_{(\mathbf{A},q,a)}$ corresponds exactly to the distribution η_p.

This is "a" and not "the" companion distribution since η_p does not explicit the start configuration.

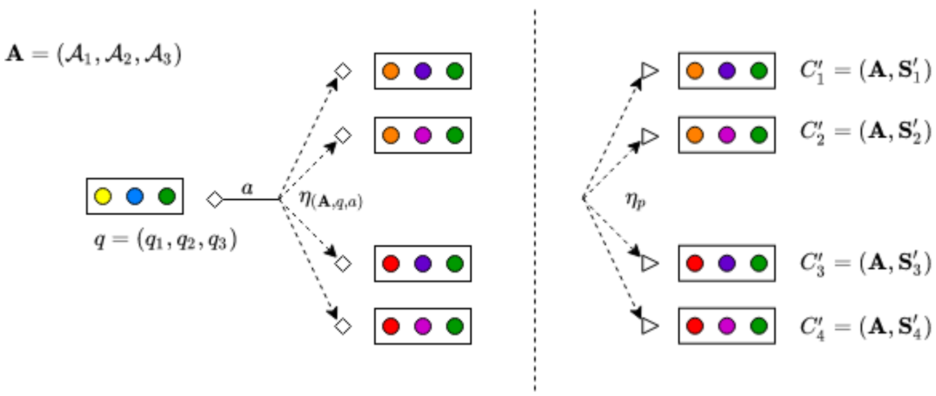


Figure 3 A preserving distribution is matching its companion distribution.

▶ **Lemma 35** (Joint preserving probability distribution for union of configuration)**.** *Let* $\mathbf{A}_X$, $\mathbf{A}_Y$, $\mathbf{A}_Z = \mathbf{A}_X \cup \mathbf{A}_Y$ *be family of automata. Let* $C_X = (\mathbf{A}_X, \mathbf{S}_X)$ *and* $C_Y = (\mathbf{A}_Y, \mathbf{S}_Y)$ *be* *two compatible configurations. Let* $C_Z = (\mathbf{A}_Z, \mathbf{S}_Z) = C_X \cup C_Y$ *be a compatible configuration. Let* $\mathcal{A}_X$ *(resp.* $\mathcal{A}_Y$ *and* $\mathcal{A}_Z$*) the automaton issued from the composition of automata in* $\mathbf{A}_X$ *(resp.* $\mathbf{A}_Y$ *and* $\mathbf{A}_Z$*). Let* q_X *(resp.* q_Y *and* q_Z*) be the current states of* $\mathcal{A}_X$ *at configuration* C_X *(resp.* $\mathcal{A}_Y$ *at configuration* C_Y *and* $\mathcal{A}_Z$ *at configuration* C_Z*)*

Let η_p^X *and* η_p^Y *be preserving distributions that have* $\eta_{(X,q_X,a)}$ *and* $\eta_{(Y,q_Y,a)}$ *as companion distribution. We note* η_p^Z *the preserving distributions that have* $\eta_{(Z,q_Z,a)}$ *as companion distribution.*

For every configuration $C'_Z = (\mathbf{A}_Z, \mathbf{S}'_Z) = C'_X \cup C'_Y$*, with* $C'_X = (\mathbf{A}_X, \mathbf{S}'_X)$ *and* $C'_Y = (\mathbf{A}_Y, \mathbf{S}'_Y)$*,* $\eta_p^Z(C'_Z) = (\eta_p^X \otimes \eta_p^Y)((C'_X, C'_Y))$*.*

Proof. We have $\eta_{(\mathcal{A}_Z,q_Z,a)} = \eta_{(\mathcal{A}_X,q_X,a)} \otimes \eta_{(\mathcal{A}_Y,q_Y,a)}$. Parallely, η_p^X and η_p^Y are preserving distributions that have $\eta_{(\mathcal{A}_X,q_X,a)}$ and $\eta_{(\mathcal{A}_Y,q_Y,a)}$ as companion distribution, while η_p^Z is preserving distributions that have $\eta_{(\mathcal{A}_Z,q_Z,a)}$ as companion distribution. ◀

Now, we can naturally define a preserving transition (C, a, η_p) from a configuration C via an action a with a companion transition of η_p. It allows us to say what is the "static" probabilistic transition from a configuration C via an action a if no creation or destruction occurs.

▶ **Definition 36** (preserving transition). Let $C = (\mathbf{A}, \mathbf{S})$ be a compatible configuration, $q = US(C)$ and $\eta_p \in P(Q_{conf}, \mathcal{F}_{Q_{conf}})$ be a preserving transition with $\mathbf{A}_s$ as family support.

Then say that (C, a, η_p) is a *preserving configuration transition*, noted $C \overset{a}{\rightharpoonup} \eta_p$ if

- $\mathbf{A}_s = \mathbf{A}$
- $\eta_{(\mathbf{A},q,a)}$ is a companion distribution of η_p

For every preserving configuration transition (C, a, η_p), we note $\eta_{(C,a),p} = \eta_p$.

The preserving transition of a configuration corresponds to the transition of the composition of the corresponding automata at their corresponding current states.

No we are ready to define our "dynamic" transition, that allows a configuration to create or destroy some automata.

At first, we define reduced distribution that leads to reduced configurations only, where all the automata that reach a state with an empty signature are destroyed.

▶ **Definition 37** (reduced distribution). A *reduced* distribution $\eta_r \in Disc(Q_{conf}, \mathcal{F}_{Q_{conf}})$ is a probabilistic distribution verifying that for every configuration $C \in supp(\eta_r)$, $C = reduced(C)$.

Now, we generate reduced distribution with a preserving distribution that describes what happen to the automata that already exist and a family of new automata that are created.

▶ **Definition 38** (Generation of reduced distribution). Let $\eta_p \in Disc(Q_{conf})$ be a preserving distribution with $\mathbf{A}$ as family support. Let $\varphi \subset Autids$. We say the reduced distribution $\eta_r \in Disc(Q_{conf})$ is generated by η_p and φ if it exists a non-reduced distribution $\eta_{nr} \in Disc(Q_{conf})$, s. t.

- (φ is created with probability 1)
  $\forall (\mathbf{A}'', \mathbf{S}'') \in Q_{conf}$, if $\mathbf{A}'' \neq \mathbf{A} \cup \varphi$, then $\eta_{nr}((\mathbf{A}'', \mathbf{S}'')) = 0$
- (freshly created automata start at start state)
  $\forall (\mathbf{A}'', \mathbf{S}'') \in Q_{conf}$, if $\exists \mathcal{A}_i \in \varphi - \mathbf{A}$ so that, $\mathbf{S}''(\mathcal{A}_i) \neq \bar{q}_i$, then $\eta_{nr}((\mathbf{A}'', \mathbf{S}'')) = 0$

- (The non-reduced transition match the preserving transition) $\forall (\mathbf{A}'', \mathbf{S}'') \in Q_{conf}$, s. t. $\mathbf{A}'' = \mathbf{A} \cup \varphi$ and $\forall \mathcal{A}_j \in \varphi, \mathbf{S}''(\mathcal{A}_j = \overline{\mathbf{x}}_j)$, $\eta_{nr}((\mathbf{A}'', \mathbf{S}'')) = \eta_p(\mathbf{A}, \mathbf{S}'' \lceil \mathbf{A}))$
- (The reduced transition match the non-reduced transition) $\forall c' \in Q_{conf}$, if $c' = reduce(c'), \eta_r(c') = \Sigma_{(c'', c' = reduce(c''))} \eta_{nr}(c'')$, if $c' \neq reduce(c')$, then $\eta_r(c') = 0$

▶ **Definition 39** (Intrinsic transition)**.** Let $(\mathbf{A}, \mathbf{S})$ be arbitrary reduced compatible configuration, let $\eta \in Disc(Q_{conf})$, and let $\varphi \subseteq Autids$, $\varphi \cap \mathbf{A} = \emptyset$. Then $\langle \mathbf{A}, \mathbf{S} \rangle \stackrel{a}{\Longrightarrow}_{\varphi} \eta$ if η is generated by η_p and φ with $(\mathbf{A}, \mathbf{S}) \stackrel{a}{\rightharpoonup} \eta_p$.

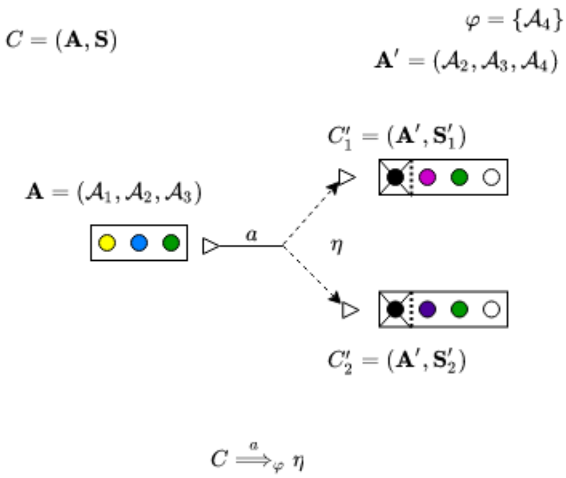


Figure 4 An intrinsinc transition where $\mathcal{A}_1$ is destroyed deterministically and $\mathcal{A}_4$ is created deterministically.

The assumption of deterministic creation is not restrictive, nothing prevents from flipping a coin at state s_0 to reach s_1 with probability p or s_2 with probability $1 - p$ and only create a new automaton in state s_2 with probability 1, while the action create is not enabled in state s_1.

4.3 Probabilistic Configuration Automata

▶ **Definition 40** (Probabilistic Configuration Automaton)**.** A probabilistic configuration automaton (PCA) K consists of the following components:

- 1. A probabilistic signature I/O automaton $psioa(K)$. For brevity, we define $states(K) = states(psioa(K)), start(K) = start(psioa(K)),\ sig(K) = sig(psioa(K)), steps(K) = steps(psioa(K))$, and likewise for all other (sub)components and attributes of $psioa(K)$.
- 2. A configuration mapping $config(K)$ with domain $states(K)$ and such that $config(K)(x)$ is a reduced compatible configuration for all $q_K \in states(K)$.
- 3. For each $q_K \in states(K)$, a mapping $created(K)(\mathbf{x})$ with domain $sig(K)(\mathbf{x})$ and such that $\forall a \in sig(K)(q), created(K)(q)(a) \subseteq Autids$
- 4. A hidden-actions mapping $hidden\text{-}actions(K)$ with domain $states(K)$ and such that $hidden\text{-}actions(K)(q_K) \subseteq out(config(K)(q_K))$.

and satisfies the following constraints

- 1. If $config(K)(\bar{q}_K) = (\mathbf{A}, \mathbf{S})$, then $\forall \mathcal{A}_i \in \mathbf{A}, \mathbf{S}(\mathcal{A}_i) = \bar{q}_i$
- 2. If $(q_K, a, \eta) \in steps(K)$ then $config(K)(q_K) \stackrel{a}{\Longrightarrow}_\varphi \eta'$, where $\varphi = created(K)(q_K)(a)$ and $\eta(y) = \eta'(config(K)(y))$ for every $\mathbf{y} \in states(K)$
- 3. If $q_K \in states(K)$ and $config(K)(q_K) \stackrel{a}{\Longrightarrow}_\varphi \eta'$ for some action a, $\varphi = created(K)(x)(a)$, and reduced compatible probabilistic measure $\eta' \in P(Q_{conf}, \mathcal{F}_{Q_{conf}})$, then $(q_K, a, \eta) \in steps(K)$ with $\eta(\mathbf{y}) = \eta'(config(K)(\mathbf{y}))$ for every $\mathbf{y} \in states(K)$.
- 4. For all $q_K \in states(K)$, $sig(K)(q_K) = hide(sig(config(K)(q_K)), hidden\text{-}actions(q_K))$, which implies that
  - (a) $out(K)(q_K) \subseteq out(config(K)(q_K))$,
  - (b) $in(K)(q_K) = in(config(K)(q_K))$,
  - (c) $int(K)(q_K) \supseteq int(config(K)(q_K))$, and
  - (d) $out(K)(q_K) \cup int(X)(q_K) = out(config(K)(q_K)) \cup int(config(K)(q_K))$

4 (d) states that the signature of a state q_K of K must be the same as the signature of its corresponding configuration $config(K)(q_K)$, except for the possible effects of hiding operators, so that some outputs of $config(K)(q_K)$ may be internal actions of K in state q_K.

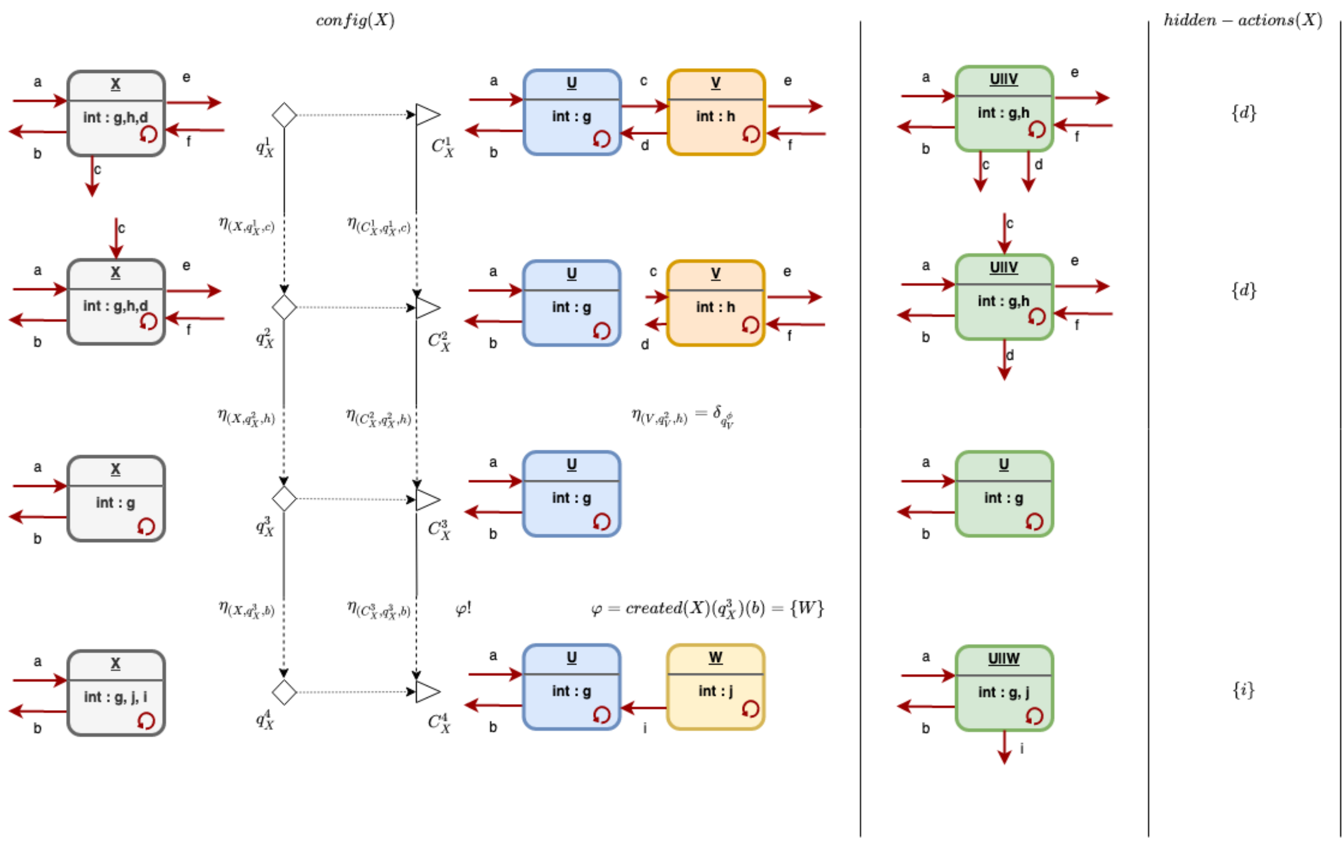


Figure 5 A PCA life cycle.

Additionnaly, we can define the current constitution of a PCA, which is the union of the current constitution of the element of its current corresponding configuration.

▶ **Definition 41** (Constitution of a PCA). Let K be a PCA. For every $q \in states(K)$,

$constitution(K)(q) = constitution(psioa(K))(q) =$

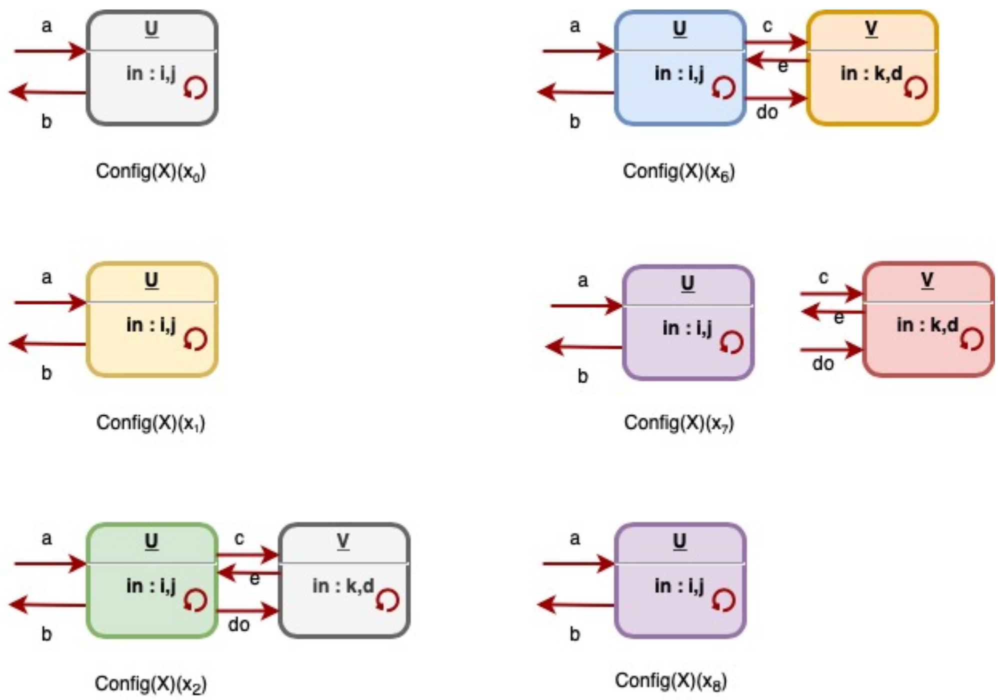


Figure 6 Example of Configuration Automaton execution. We illustrate succession of configurations mapped with the configuration automaton X. We denote $\forall x_i \in states(X), C_i = \langle \mathbf{A}_i, \mathbf{S}_i \rangle = Config(X)(x_i)$, $C_0 \stackrel{a}{\Longrightarrow}_{\emptyset} C_1 \stackrel{i}{\Longrightarrow}_{V} C_2 ... C_6 \stackrel{do}{\Longrightarrow}_{\emptyset} C_7 \stackrel{d}{\Longrightarrow}_{\emptyset} C_8 \stackrel{b}{\Longrightarrow}_{\emptyset} C_9$. The automata included in the configuration are either $\{U\}$ or $\{U, V\}$. The internal action i of U aims to create the automaton V. do represents a destruction order, while d is a destruction action. The step (s_V, d, s'_V) is so that $\widehat{sig}(V)(s'_V) = \emptyset$
, thus $\langle \mathbf{A}_8, \mathbf{S}_8 \rangle$ does not handle V because of reduction.

$\bigcup_{\mathcal{A} \in auts(config(K)(q))} constitution(\mathcal{A})(map(config(K)(q))(\mathcal{A}))$.

We note $UA(K) = \bigcup_{q \in K} constitution(K)(q)$ the universal set of atomic components of K.

4.4 Compatibility, composition

▶ **Definition 42** (Union of configurations). Let $C_1 = (\mathbf{A}_1, \mathbf{S}_1)$ and $C_2 = (\mathbf{A}_2, \mathbf{S}_2)$ be configurations such that $\mathbf{A}_1 \cap \mathbf{A}_2 = \emptyset$. Then, the union of C_1 and C_2, denoted $C_1 \cup C_2$, is the configuration $C = (\mathbf{A}, \mathbf{S})$ where $\mathbf{A} = \mathbf{A}_1 \cup \mathbf{A}_2$ (lexicographically ordered) and $\mathbf{S}$ agrees with $\mathbf{S}_1$ on $\mathbf{A}_1$, and with $\mathbf{S}_2$ on $\mathbf{A}_2$. It is clear that configuration union is commutative and associative. Hence, we will freely use the n-ary notation $C_1 \cup ... \cup C_n$ (for any $n \geq 1$) whenever $\forall i, j \in [1:n], i \neq j, auts(C_i) \cap auts(C_j) = \emptyset$.

▶ **Definition 43** (PCA partially-compatible at a state). Let $\mathbf{X} = (X_1, ..., X_n)$ be a family of PCA. We note $psioa(\mathbf{X}) = (psioa(X_1), ..., psioa(X_n))$. The PCA $X_1, ..., X_n$ are partially-compatible at state $q_{\mathbf{X}} = (q_{X_1}, ..., q_{X_n}) \in states(X_1) \times ... \times states(X_n)$ iff:

1. $\forall i, j \in [1:n], i \neq j : auts(config(X_i)(q_{X_i})) \cap auts(config(X_j)(q_{X_j})) = \emptyset$.
2. $\{sig(X_1)(q_{X_1}), ..., sig(X_n)(q_{X_n})\}$ is a set of compatible signatures.
3. $\forall i, j \in [1:n], i \neq j : \forall a \in \widehat{sig}(X_i)(q_{X_i}) \cap \widehat{sig}(X_j)(q_{X_j}) : created(X_i)(q_{X_i})(a) \cap$

$created(X_j)(q_{X_j})(a) = \emptyset$.

4. $\forall i, j \in [1:n], i \neq j : constitution(X_i)(q_{X_i}) \cap constitution(X_j)(q_{X_j}) = \emptyset$

We can remark that if $\forall i, j \in [1:n], i \neq j : auts(config(X_i)(q_{X_i})) \cap auts(config(X_j)(q_{X_j})) = \emptyset$ and $\{sig(X_1)(q_{X_1}), ..., sig(X_n)(q_{X_n})\}$ is a set of compatible signatures, then $config(X_1)(q_{X_1}) \cup ... \cup config(X_n)(q_{X_n})$ is a reduced compatible configuration.

If $\mathbf{X}$ is partially-compatible at state $q_{\mathbf{X}}$, for every action $a \in \widehat{sig}(psioa(\mathbf{X}))(q_{\mathbf{X}})$, we note $\eta_{(\mathbf{X},q_{\mathbf{X}},a)} = \eta_{(psioa(\mathbf{X}),q_{\mathbf{X}},a)}$ and we extend this notation with $\eta_{(\mathbf{X},q_{\mathbf{X}},a)} = \delta_{q_{\mathbf{X}}}$ if $a \notin \widehat{sig}(psioa(\mathbf{X}))(q_{\mathbf{X}})$.

▶ **Definition 44** (pseudo execution). Let $\mathbf{X} = (X_1, ..., X_n)$ be a set of PCA. A *pseudo execution fragment* of $\mathbf{X}$ is a pseudo execution fragment of $psioa(\mathbf{A})$, s. t. for every non final state q^i, $\mathbf{X}$ is partially-compatible at state q^i (namely the conditions (1) and (3) need to be satisfied)

A *pseudo execution* α of $\mathbf{X}$ is a pseudo execution fragment of $\mathbf{X}$ with $fstate(\alpha) = (\bar{q}_{X_1}, ..., \bar{q}_{X_n})$.

▶ **Definition 45** (reachable state). Let $\mathbf{X} = (X_1, ..., X_n)$ be a set of PSIOA. A state q of $\mathbf{X}$ is *reachable* if it exists a pseudo execution α of $\mathbf{X}$ ending on state q.

▶ **Definition 46** (partially-compatible PCA). Let $\mathbf{X} = (X_1, ..., X_n)$ be a set of PCA. The automata $X_1, ..., X_n$ are *ℓ-partially-compatible* with $\ell \in \mathbb{N}$ if no pseudo-execution α of $\mathbf{X}$ with $|\alpha| \leq \ell$ ends on non-partially-compatible state q. The automata $X_1, ..., X_n$ are *partially-compatible* if $\mathbf{X}$ is partially-compatible at each reachable state q, i. e. $\mathbf{X}$ is *ℓ-partially-compatible* for every $\ell \in \mathbb{N}$.

▶ **Definition 47** (compatible PCA). Let $\mathbf{X} = (X_1, ..., X_n)$ be a set of PCA. The automata $X_1, ..., X_n$ are *compatible* if the automata $X_1, ..., X_n$ are partially-compatible for each state of $states(X_1) \times ... \times states(X_n)$.

▶ **Definition 48** (Composition of configuration automata). Let $X_1, ..., X_n$, be compatible (resp. partially-compatible) configuration automata. Then $X = X_1||...||X_n$ is the state machine consisting of the following components:

1. $psioa(X) = psioa(X_1)||...||psioa(X_n)$ (where the composition can be the one dedicated to only partially-compatible PCA).
2. A configuration mapping $config(X)$ given as follows. For each $x = (x_1, ..., x_n) \in states(X), config(X)(x) = config(X_1)(x_1) \cup ... \cup config(X_n)(x_n)$.
3. For each $x = (x_1, ..., x_n) \in states(X)$, a mapping $created(X)(x)$ with domain $\widehat{sig}(X)(x)$ and given as follows. For each $a \in \widehat{sig}(X)(x), created(X)(x)(a) = \bigcup_{a \in \widehat{sig}(X_i)(x_i), i \in [1:n]} created(X_i)(x_i)(a)$.
4. A hidden-action mapping $hidden\text{-}actions(X)$ with domain $states(X)$ and given as follows. For each $x = (x_1, ..., x_n) \in states(X)$, $hidden\text{-}actions(x) = \bigcup_{i \in [1:n]} hidden\text{-}actions(x_i)$

We define $states(X) = states(sioa(X)), start(X) = start(sioa(X)), sig(X) = sig(sioa(X)), steps(X) = steps(sioa(X))$, and likewise for all other (sub)components and attributes of $sioa(X)$.

▶ **Theorem 49** (PCA closeness under composition). *Let $X_1, ..., X_n$, be compatible or partially-compatible PCA. Then $X = X_1||...||X_n$ is a PCA.*

Proof. We need to show that X verifies all the constraints of definition 40.

- (Constraint) 1: The demonstration is basically the same as the one in [1], section 5.1, proposition 21, p 32-33. Let $\bar{q}_X$ and $(\mathbf{A}, \mathbf{S}) = config(X)(\bar{q}_X)$. By the composition of psioa, then $\bar{q}_X = (\bar{q}_{X_1}, ..., \bar{q}_{X_n})$. By definition, $config(X)(\bar{q}_X) = config(X_1)(\bar{q}_{X_1}) \cup ... \cup config(X_n)(\bar{q}_{X_n})$. Since for every $j \in [1:n]$, X_j is a configuration automaton, we apply constraint 1 to X_j to conclude $\mathbf{S}(\mathcal{A}_\ell) = \bar{q}_{\mathcal{A}_\ell}$ for every $\mathcal{A}_\ell \in auts(config(X_j)(\bar{q}_{X_j})$. Since $(auts(config(X_1)(\bar{q}_{X_1}), ..., auts(config(X_n)(\bar{q}_{X_n}))$ is a partition of $\mathbf{A}$ by definition of composition, $\mathbf{S}(\mathcal{A}_\ell) = \bar{q}_{\mathcal{A}_\ell}$ for every $\mathcal{A}_\ell \in \mathbf{A}$ which ensures X verifies constraint 1.

- (Constraint 2) Let (x, a, η) be an arbitrary element of $steps(X)$. We will establish $config(X)(x) \stackrel{a}{\Longrightarrow}_\varphi \eta'$ with $\varphi = created(X)(x)(a)$ and $\eta'(config(X)(\mathbf{y})) = \eta(\mathbf{y})$ for every state $\mathbf{y} \in states(X)$. For brevity, let $\mathcal{A}_i = sioa(X_i)$ for $i \in [1:n]$. Now $(x, a, \eta) \in steps(X)$. So $(x, a, \eta) \in steps(sioa(X))$ by definition. Also by Definition 48, $sioa(X) = sioa(X_1)||...||sioa(X_n) = \mathcal{A}_1||...||\mathcal{A}_n$. From definition of sioa composition, there exists a nonempty $\phi^a_e \subseteq [1:n]$ such that $\forall i \in \phi^a_e$, $a \in \widehat{sig}(\mathcal{A}_i)(\mathbf{x_i})$ and $\forall j \in \phi^a_n = ([1:n] \setminus \phi^a_e)$, $a \notin \widehat{sig}(\mathcal{A}_j)(\mathbf{x_j})$.
  So, $(x, a, \eta) \in steps(\mathcal{A}_1||...||\mathcal{A}_n$. Since $x \in states(\mathcal{A}_1||...||\mathcal{A}_n)$, we can write x, as $(\mathbf{x}_1, ..., \mathbf{x}_n)$ where $\mathbf{x}_i \in states(\mathcal{A}_i)$ for $i \in [1:n]$. In the same way, we can write $\eta = \eta_1 \otimes ... \otimes \eta_n$ where for each $i \in \phi^a_e, \eta_i = \eta_{\mathbf{x}_i,a}$ $(\mathbf{x} \stackrel{a}{\to} \eta_i)$ and $j \in \phi^a_n, \eta_j = \delta_{\mathbf{x}_j}$.
  We have $(\bigwedge_{i\in\phi^a_e} a \in \widehat{sig}(\mathcal{A}_i)(\mathbf{x_i}) \wedge (\mathbf{x}_i, a, \eta_i) \in steps(\mathcal{A}_i)) \wedge (\bigwedge_{j\in[1:n]\setminus\phi^a_e} a \notin \widehat{sig}(\mathcal{A}_j)(\mathbf{x}_j) \wedge \eta_j = \delta_{\mathbf{x}_j}$ (a)
  Each X_i, $i \in [1:n]$, is a configuration automaton. Hence, by (a) and constraint 2 applied to each X_i, with $i \in \phi$, we have: $\bigwedge_{i\in\phi^a_n} config(X_i)(x_i) \stackrel{a}{\Longrightarrow}_{\varphi_i} \eta'_i$ with $\varphi_i = created(X_i)(x_i)(a)$ and $\eta'_i(config(X)(\mathbf{y}_i)) = \eta_i(\mathbf{y}_i)$ for every state $\mathbf{y}_i \in states(X_i)$, and $\bigwedge_{j\in\phi^a_n} config(X_j)(x_j) \stackrel{a}{\Longrightarrow}_\emptyset \delta'_{\mathbf{x}_j}$.
  Since $X_1, ..., X_n$ are compatible, we have that $config(X_1)(\mathbf{x}_1) \cup ... \cup config(X_n)(\mathbf{x}_n)$ and $config(X_1)(\mathbf{y}_1) \cup ... \cup config(X_n)(\mathbf{y}_n)$ are both reduced compatible configurations for every $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_n)$ s. t. $\mathbf{y}_k \in supp(\eta_k)$ for each $k \in [1:n]$.
  By definition, $\varphi = created(X)(x)(a) = \bigcup_{i\in\phi^a_e} created(X_i)(x_i)(a)$.
  Thereafter, we obtain
  $(\bigcup_{k\in[1:n]} config(X_k)(\mathbf{x}_k)) \stackrel{a}{\Longrightarrow}_\phi \eta')$ where $\eta' = \eta'_1 \otimes ... \otimes \eta'_n$.
  For every $\mathbf{y} \in states(X), \eta'(config(X)(\mathbf{y})) = \eta(\mathbf{y})$
  Finally, we obtain $config(X)(x) \stackrel{a}{\Longrightarrow}_{created(X)(x)(a)} \eta'$ with $\eta'(config(X)(\mathbf{y})) = \eta(\mathbf{y})$ for every $\mathbf{y} \in states(X)$.
- (Constraint 3) Let $\mathbf{x}$ be an arbitrary state in $states(X)$ and η' an arbitrary probability measure on the configuration with a support corresponding to reduced compatible configuration such that $config(X)(x) \stackrel{a}{\Longrightarrow}_\varphi \eta'$ for some action a with $\varphi = created(X)(x)(a)$. We must show $\exists \eta_{\mathbf{x},a} \in P(Q_X, \mathcal{F}_{Q_X}) : (x, a, \eta_{\mathbf{x},a}) \in steps(X)$ $(\mathbf{x} \stackrel{a}{\to} \eta_{\mathbf{x},a})$ and for every state $\mathbf{y} \in states(X)$, $\eta'(config(X)(\mathbf{y})) = \eta_{\mathbf{x},a}(\mathbf{y})$.
  We can write $\mathbf{x}$ as $(\mathbf{x}_1, ..., \mathbf{x}_n)$ where $\mathbf{x}_i \in states(X_i)$ for $i \in [1:n]$. Since $X_1, ..., X_n$ are compatible, we have, by compatibility of configuration automata, that $auts(config(X_i)(\mathbf{x}_i)) \cap auts(config(X_j)(\mathbf{x}_j)) = \emptyset, \forall i, j \in [1:n], i \neq j$, (thus, all SIOA in these configurations are unique) and that $config(X_1)(x_1) \cup ... \cup config(X_n)(\mathbf{x}_n)$ is a reduced compatible configuration. Also, from configuration composition, $config(X)(\mathbf{x}) = \bigcup_{i\in[1:n]} config(X_i)(\mathbf{x}_i)$, that is $\bigcup_{i\in[1:n]} config(X_i)(\mathbf{x}_i) \stackrel{a}{\Longrightarrow}_\varphi \eta'$. (a)
  From definition of sioa composition, there exists a nonempty $\phi^a_e \subseteq [1:n]$ such that $\forall i \in \phi^a_e$, $a \in \widehat{sig}(\mathcal{A}_i)(\mathbf{x_i})$ and $\forall j \in \phi^a_{ne} = ([1:n] \setminus \phi^a_e)$, $a \notin \widehat{sig}(\mathcal{A}_j)(\mathbf{x_j})$.
  We have $config(X)(\mathbf{x}) \stackrel{a}{\Longrightarrow}_\varphi \eta'$. with $\eta' = \eta'_1 \otimes ... \otimes \eta'_n$ and for every $i \in \phi^a_e$ $supp(\eta'_i) \subseteq \{c' || \exists c'', (c' = reduced(c'')) \wedge (auts(c'') = auts(config(X_i)(\mathbf{x}_i)) \cup \varphi_i) \wedge (\forall \mathcal{A} \in \varphi_i, maps(c'')(\mathcal{A}) = \bar{\mathbf{x}_\mathcal{A}})\}$ with $\varphi_i = created(X_i)(\mathbf{x}_i)(a)$ and for every $j \in \phi^a_{ne} = ([1:n] \setminus \phi^a_e)$, $\eta'_j = \delta_{Config(X_j)(\mathbf{x}_j)}$
  We have for every $i \in \phi^a_e$ $config(X_i)(\mathbf{x}_i) \stackrel{a}{\Longrightarrow}_{\varphi_i} \eta'_i$, which means for every $i \in \phi^a_e$,

$(\mathbf{x}_i, a, \eta_i) \in steps(X_i)$ with for every $\mathbf{y}_i$ $\eta_i(\mathbf{y}_i) = \eta'_i(config(X_i)(\mathbf{y}_i))$.
For every $j \in \phi^a_{ne} = [1:n] \setminus \phi^a_e$, we note $\eta_j = \delta_{\mathbf{x}_j}$.
From this, $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$, $\eta = \eta_1 \otimes ... \otimes \eta_n$, and definition of configuration composition, we conclude $(\mathbf{x}, a, \eta) \in steps(X)$ and for every $\mathbf{y} \in states(Y)$, $\eta(\mathbf{y}) = \eta'(config(X)(\mathbf{y}))$

- (Constraint 4).

For every $i \in [1, n]$, we note $h_{X_i} = hidden\text{-}actions(X_i)(q_{X_i})$ and $h = \bigcup_{i \in [1,n]} h_{X_i}$. Since $\{X_i | i \in [1, n]\}$ are partially-compatible in state $q_X = (q_{X_1}, ..., q_{X_n})$, we have both $\{config(X_i)(q_{X_i}) | i \in [1, n]\}$ compatible and $\forall i, j \in [1, n], in(config(X_i)(q_{X_i})) \cap h_{X_j} = \emptyset$. By compatibility, $\forall i, j \in [1, n], out(config(X_i)(q_{X_i})) \cap out(config(X_j)(q_{X_j})) = int(config(X_i)(q_{X_i})) \cap \widehat{sig}(config(X_j)(q_{X_j})) = \emptyset$, which finally gives $\forall i, j \in [1, n], \widehat{sig}(config(X_i)(q_{X_i})) \cap h_{X_j} = \emptyset$.
Hence, we can apply commutativity to obtain $hide(sig(config(X_1)(q_{X_1})) \times .... \times config(X_n)(q_{X_n}), h_{X_1} \cup ... \cup h_{X_n}) = hide(sig(config(X_1)(q_{X_1})), h_{X_1}) \times ... \times hide(sig(config(X_n)(q_{X_n})), h_{X_n})$.
That is $sig(psioa(X))(q_X) = sig(psioa(X_1))(q_{X_1})) \times .... \times sig(psioa(X_n))(q_{X_n}))$ because of (1) is compatible with $sig(psioa(X))(q_X) = hide(sig(config(X)(x)), h)$ because of (2) and (4).
Furthermore $h \subset config(X)(q_X)$, since $h_{X_i} \subset config(X_i)(q_{X_i})$.
This terminates the proof.

◀

5 Projection

This section aims to formalise the idea of a PCA $X_\mathcal{A}$ considered without an internal PSIOA $\mathcal{A}$. This PCA will be noted $Y_\mathcal{A} = X_\mathcal{A} \setminus \{\mathcal{A}\}$. This is an important step in our reasoning since we will be able to formalise in which sense $X_\mathcal{A}$ and $psioa(X_\mathcal{A} \setminus \{\mathcal{A}\}) || \mathcal{A}$ are similar.

5.1 projection on configurations

At first we need some particular precautions to define properly the probabilistic spaces.

The next definition captures the idea of probabilistic measure deprived of a psioa $\mathcal{A}$.

▶ **Definition 50** (probabilistic measure projection). Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ be a (lexically ordered) family of PSIOA partiall-compatible at state $q = (q_1, ..., q_n) \in Q_{\mathcal{A}_1} \times ... \times Q_{\mathcal{A}_n}$. Let $\mathbf{A}^s = (\mathcal{A}_{s^1}, ..., \mathcal{A}_{s^n}) \subset \mathbf{A}$. We note :

- $q \setminus \{\mathcal{A}_k\} = (q_1, ..., q_{k-1}, q_{k+1}, ..., q_n)$ if $\mathcal{A}_k \in \mathbf{A}$ and $q \setminus \{\mathcal{A}_k\} = q$ otherwise.
- $q \setminus \mathbf{A}^s = (q \setminus \{\mathcal{A}_{s^n}\}) \setminus (\mathbf{A}^s \setminus \{\mathcal{A}_{s^n}\})$ (recursive extension of the previous item).
- $q \restriction \mathcal{A}_k = q_k$ if $\mathcal{A}_k \in \mathbf{A}$ only.
- $q \restriction \mathbf{A}^s = q \setminus (\mathbf{A} \setminus \mathbf{A}^s)$ (recursive extension of the previous item).

Let $q' = q \setminus \mathbf{A}^s$ and $q'' = q \restriction \mathbf{A}^s$ if $\mathbf{A}^s \subset \mathbf{A}$. Let $\mathbf{A}' = \mathbf{A} \setminus \mathbf{A}^s$ and $\mathbf{A}'' = \mathbf{A}^s \subset \mathbf{A}$. Let $a' \in \widehat{sig}(\mathbf{A}')(q')$ and $a'' \in \widehat{sig}(\mathbf{A}'')(q'')$. We note

- $\eta_{(\mathbf{A},q,a')} \setminus \mathbf{A}^s \triangleq \eta_{(\mathbf{A}',q',a')}$ and
- $\eta_{(\mathbf{A},q,a'')} \restriction \mathbf{A}^s \triangleq \eta_{(\mathbf{A}'',q'',a'')}$ if $\mathbf{A}^s \subset \mathbf{A}$.

Then we apply this notation to preserving distributions.

▶ **Definition 51** (preserving distribution projection). Let η_p be a preserving distribution. Let $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ its family support. Let H be its set of companion distributions of η_p (s.

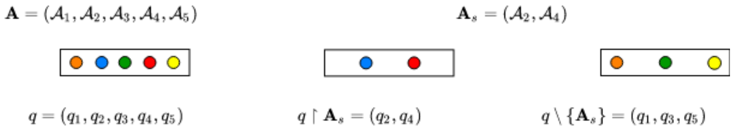


Figure 7 State projection

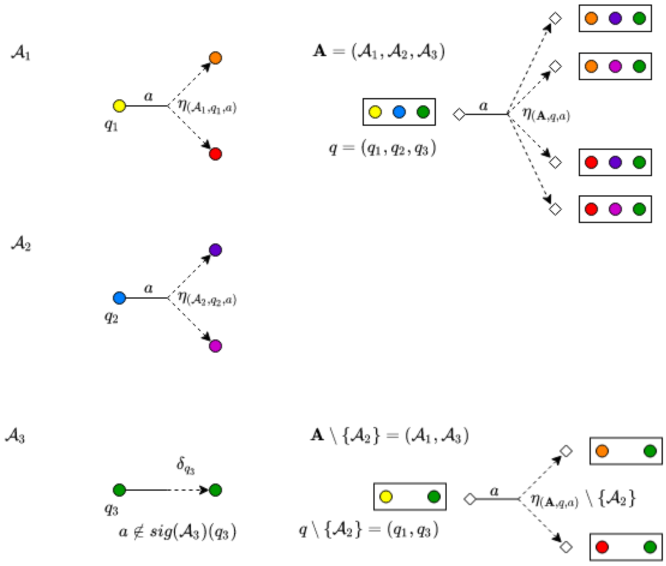


Figure 8 Family transition projection

t. for every $\eta \in H$, $\eta = \eta_1 \otimes ... \otimes \eta_n$ with $\eta_i \in Disc(Q_{\mathcal{A}_i})$).Then $\eta_p \setminus \mathbf{A}^s$ is the preserving distribution with $\mathbf{A} \setminus \mathbf{A}^s$ as familiy support and $H' = \{\eta \setminus \mathbf{A}^s | \eta \in H\}$ as companion distribution set. If $\mathbf{A}^s \subset \mathbf{A}$, then $\eta_p \upharpoonright \mathbf{A}^s$ is the preserving distribution with $\mathbf{A} \upharpoonright \mathbf{A}^s$ as familiy support and $H'' = \{\eta \upharpoonright \mathbf{A}^s | \eta \in H\}$ as companion distribution set.

▶ **Definition 52** (intrinsinc transition projection)**.** Let $\eta \in Disc(Q_{conf})$ generated by φ and $\eta_p \in Disc(Q_{conf})$. We note $\eta \setminus \mathbf{A}^s$ the probabilistic measure on configurations generated by $\varphi \setminus \mathbf{A}^s$ and $\eta_p \setminus \mathbf{A}^s$ and we note $\eta \upharpoonright \mathbf{A}^s$ the probabilistic measure on configurations generated by $\varphi \upharpoonright \mathbf{A}^s$ and $\eta_p \upharpoonright \mathbf{A}^s$.

Then we can easily determine some results when projection is applied.

▶ **Lemma 53** (family distribution projection)**.** *(see figure 11) Let* $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$*, let* $\eta = \eta_1 \otimes ... \otimes \eta_n$ *with* $\eta_i \in Disc(Q_{\mathcal{A}_i})$ *for every* $i \in [1, n]$ *. Let* $\eta' = \eta \setminus \{\mathcal{A}_k\}$*. Let* $Q'_{\mathcal{A}} = \{q \setminus \{\mathcal{A}_k\} | q \in Q_{\mathcal{A}}\}$*.*

For every $q' \in Q'_{\mathcal{A}}$*,* $\eta'(q') = \Sigma_{(q \in Q_{\mathcal{A}}, q \setminus \{\mathcal{A}_k\} = q')} \eta(q)$

Proof. This comes directly from the law of total probability. The Bayes law gives $\eta'(q') = \sum \eta(q'|q)\eta(q)$ with $\eta(q'|q) = \delta_{q'=q \setminus \{\mathcal{A}_k\}}$. Thus $\eta(q) = \sum_{q'=q \setminus \{\mathcal{A}_k\}} \eta(q)$. ◀

▶ **Lemma 54** (preserving distribution projection)**.** *(see figure 12) Let* η_p *be a preserving distribution with* $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ *as family support . Let* C_Y *be a configuration* $(\eta_p \setminus \{\mathcal{A}_k\})(C_Y) = \Sigma_{(C_X, C_X \setminus \{\mathcal{A}_k\} = C_Y)} \eta_p(C_X)$*.*

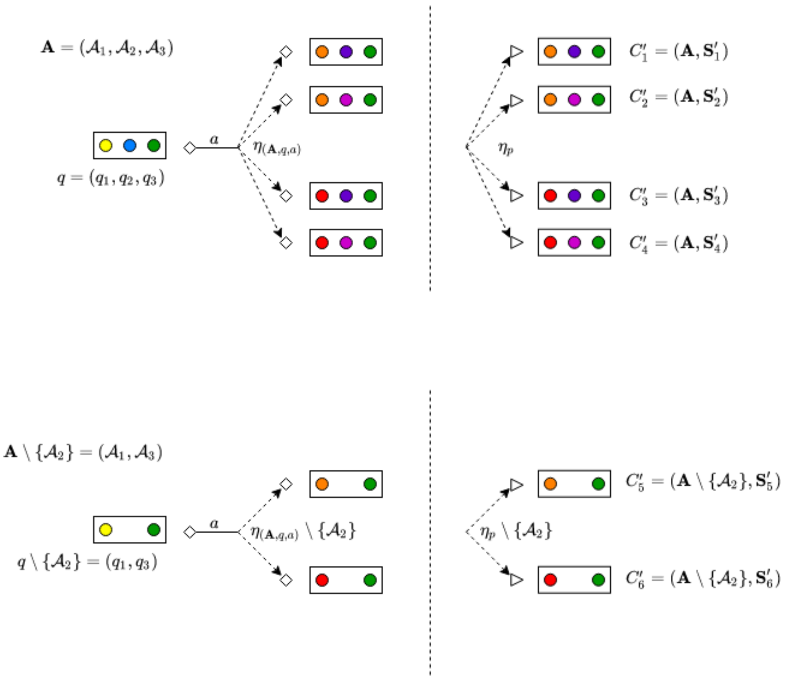


Figure 9 Preserving distribution projection

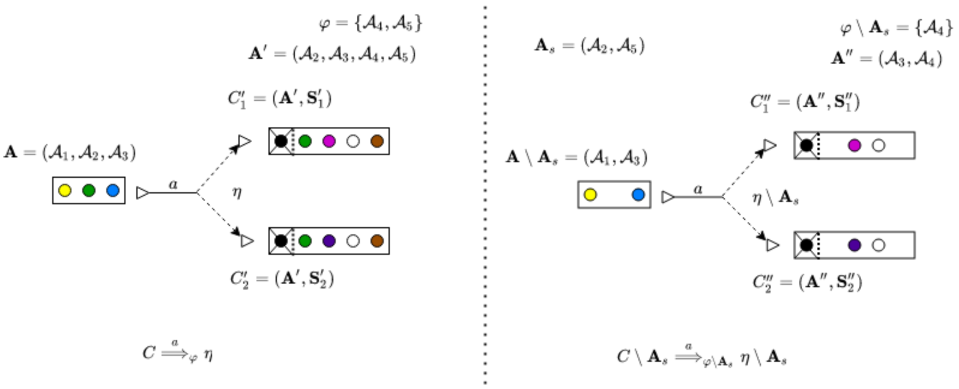


Figure 10 Intrinsinc transition projection

Proof. We can apply lemma 53 for every pair $(\eta, \eta \setminus \{\mathcal{A}_k\})$ s. t. η is a companion distribution of η_p (and $\eta \setminus \{\mathcal{A}_k\}$ is a companion distribution of $\eta_p \setminus \{\mathcal{A}_k\}$ by definition). Then we substitute in the sum of 53 every state q by the corresponding configuration. ◀

▶ **Lemma 55** (reduced distribution projection). *Let η_p be a preserving distribution with $\mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$ as family support . Let η_r be generated by φ and η_p. Let C_Y be a configuration.*

$$(\eta_p \setminus \{\mathcal{A}_k\})(C_Y) = \Sigma_{(C_X, C_X \setminus \{\mathcal{A}_k\} = C_Y)} \eta_p(C_X).$$

Let C_Y be a configuration $(\eta_r \setminus \{\mathcal{A}_k\})(C_Y) = \Sigma_{(C_X, C_X \setminus \{\mathcal{A}_k\} = C_Y)} \eta_r(C_X)$.

Proof. For a preserving transition, we get $(\eta_p \setminus \{\mathcal{A}_k\})(C_Y) = \Sigma_{(C_X, C_X \setminus \{\mathcal{A}_k\} = C_Y)} \eta_p(C_X)$ for

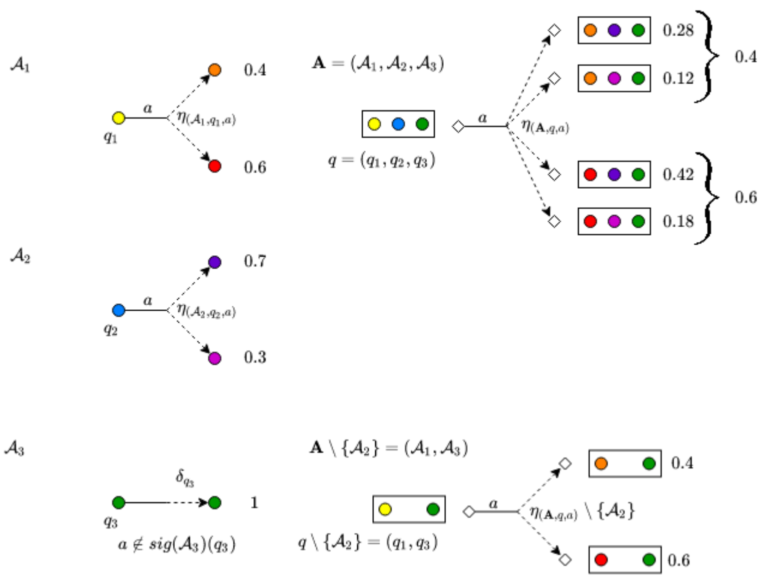


Figure 11 total probability law for family transition projection

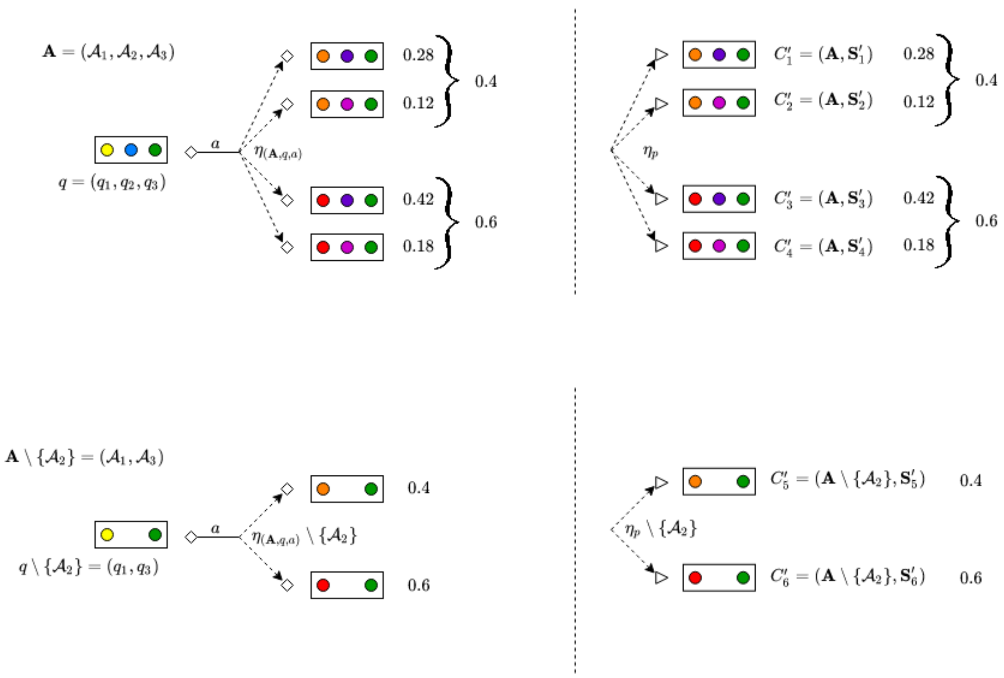


Figure 12 total probability law for preserving configuration distribution and its companion distribution

every configuration C_Y from lemma 54. By definition 38, it follows the same relation for the non-reduced transition which is matching the preserving transition. It follows the same relation for the reduced transition which is matching the non-reduced transition. ◀

▶ **Lemma 56** (projection on an intrinsinc transition). *Let C be a configuration, P an automaton $a \in \widehat{sig}(C \backslash P)$, $\varphi \subset Autids$ and $\eta \in Disc(Q_{conf})$, s. t. $C \stackrel{a}{\Longrightarrow}_{\varphi} \eta_r$. Then , $C \backslash \{P\} \stackrel{a}{\Longrightarrow}_{(\varphi \backslash \{P\})}$*

$(\eta_r \setminus \{P\})$.

Proof. We note $auts(C) = \mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n)$, $\mathbf{S} = auts(C)$ and $\mathcal{A} = \mathcal{A}_1||...||\mathcal{A}_n$. We note $q = (\mathbf{S}(\mathcal{A}_1), ..., \mathbf{S}(\mathcal{A}_n))$. Since a is enabled in $C \setminus \{P\}$, $(q \setminus \{P\}, a, \eta)$ is a transition of $\mathcal{A}$ (unique from q and a by transition determinism), while $(q, a, \eta \setminus \{P\})$ is a transition of $\mathcal{A}'$ the automaton issued from the composition of automata in $\mathbf{A} \setminus \{P\}$. This comes from the definition of composition 11. Now η_r is generated from φ and η_p where η is a companion distribution of η_p. In the same way, $\eta_r \setminus \{P\}$ is generated from $\varphi \setminus \{P\}$ and $\eta_p \setminus \{P\}$ where $\eta \setminus \{P\}$ is a companion distribution of $\eta_p \setminus \{P\}$.

Thus, $C \setminus \{P\} \stackrel{a}{\rightharpoonup} (\eta_p \setminus \{P\})$ and then $C \setminus \{P\} \stackrel{a}{\Longrightarrow}_{(\varphi \setminus \{P\})} (\eta_r \setminus \{P\})$.

◀

5.2 projection on PCA

Now we can define our PCA deprived of a PSIOA.

▶ **Definition 57** ($\mathcal{A}$-fair PCA). Let $\mathcal{A} \in Autids$. Let X be a PCA. We say that X is $\mathcal{A}$-fair if for every states q_X, q'_X, s. t. $config(X)(q_X) \setminus \mathcal{A} = config(X)(q'_X) \setminus \mathcal{A}$, then $created(X)(q_X) = created(X)(q'_X)$ and $hidden\text{-}actions(X)(q_X) = hidden\text{-}actions(X)(q'_X)$.

A $\mathcal{A}$-fair PCA is a PCA s. t. we can deduce its current properties from its current configuration deprived of $\mathcal{A}$. This allows the next definition to be well-defined.

▶ **Definition 58** ($X \setminus \{P\}$). (see figure 13) Let $P \in Autids$. Let X be a P-fair PCA. We note $X \setminus \{P\}$ the automaton Y, verifying:

- it exists a total map $\mu_s : states(X) \to states(Y)$ and $\mu_d : Disc(Q_X, \mathcal{F}_{Q_X}) \to Disc(Q_Y, \mathcal{F}_{Q_Y})$ s. t.
  - $\mu_s(\bar{q}_X) = \bar{q}_Y$
  - if $config(X)(\mathbf{x}) = (\mathsf{A}, \mathsf{S})$, $config(Y)(\mu_s(\mathbf{x})) = (\mathsf{A} \setminus \{P\}, \mathsf{S} \restriction (\mathsf{A} \setminus \{P\}))$
  - $sig(Y)(\mu_s(\mathbf{x})) = sig(X)(\mathbf{x}) \setminus P$.
  - $\forall \mathbf{x} \in states(X), \forall a \in sig(Y)(\mu_s(\mathbf{x}))$, $created(Y)(\mu_s(\mathbf{x}))(a) = created(X)(\mathbf{x})(a) \setminus \{P\}$
  - $\forall \mathbf{x} \in states(X), \forall a \in sig(Y)(\mu_s(\mathbf{x}))$ if $(\mathbf{x}, a, \eta) \in step(X)$, $(\mu_s(\mathbf{x}), a, \mu_d(\eta)) \in step(Y)$ where $\mu_d(\eta)(\mathbf{y}) = \Sigma_{\mathbf{x}, \mu_s(\mathbf{x}) = \mathbf{y}} \eta(\mathbf{x})$.
  - $\forall x \in states(X)$, if $\mathcal{A} \in auts(config(X)(q_X))$, then $hidden\text{-}actions(Y)(\mu_s(x)) = hidden\text{-}actions(X)(x) \setminus out(\mathcal{A})(maps(config(X)(q_X))(\mathcal{A})$, otherwise $hidden\text{-}actions(Y)(\mu_s(x)) = hidden\text{-}actions(X)(x)$.

In the remaining, if we consider a PCA X deprived of a PSIOA $\mathcal{A}$ we always implicitly assume that X is $\mathcal{A}$-fair.

Here we prove a serie of lemma to show that $Y = X \setminus \{P\}$ is indeed a PCA. by verifying all the constraints.

▶ **Lemma 59** (corresponding transition measure for projection). *Let P be a PSIOA. Let X be a P-fair PCA. Let $Y = X \setminus \{P\}$. Let (q_X, a, η_X) be a transition of X where $a \in act(config(X)(q_X) \setminus \{P\})$. Let η'_X s. t. $Config(X)(q_X) \stackrel{a}{\Longrightarrow}_{\varphi_X} \eta'_X$ with $\eta_X(q'_X) = \eta'_X(config(X)(q'_X))$ for every q'_X and $\varphi_X = created(X)(q_X)(a)$ (which exists by definition).*

Then $(q_Y = \mu_s(q_X), a, \eta_Y = \mu_d(\eta_X))$ is a transition of Y and $Config(Y)(q_Y) \stackrel{a}{\Longrightarrow}_{\varphi_Y} \eta'_Y$ with $\eta'_Y = \eta'_X \setminus \{P\}$, $\eta_Y(q'_Y) = \eta'_Y(config(Y)(q'_Y))$ for every q'_Y and $\varphi_Y = (\varphi_X \setminus \{P\}) = created(Y)(q_Y)(a)$.

Proof. At first, by definition of Y, $Config(Y)(q_Y = \mu_s(q_X)) = Config(X)(q_X) \setminus \{P\}$. Then, since $a \in act(config(X)(q_X) \setminus \{P\})$, we can apply lemma 56. Thus $Config(Y)(q_Y) \overset{a}{\Longrightarrow}_{\varphi_Y} \eta'_Y$ with $\eta'_Y = \eta'_X \setminus \{P\}$ and $\varphi_Y = (\varphi_X \setminus \{P\})$. By definition, $created(Y)(q_Y)(a) = created(X)(q_X)(a) \setminus \{P\}$, thus $\varphi_Y = created(Y)(q_Y)(a)$.

Let q_Y be a state of Y. By definition of $Y = X \setminus \{P\}$, $(\mu_d(\eta_X))(q_Y) = \Sigma_{q_X, \mu_s(q_X) = q_Y} \eta_X(q_X)$. By assumption, $\eta_X(q_X) = \eta'_X(config(X)(q_X))$, thus $(\mu_d(\eta_X))(q_Y) = \Sigma_{q_X, \mu_s(q_X) = q_Y} \eta'_X(config(X)(q_X))$. We substitue q_X with $config(X)(q_X)$ in the sum and obtain $(\mu_d(\eta_X))(q_Y) = \Sigma_{config(X)(q_X), config(X)(q_X) \setminus \{P\} = config}$ since $\mu_s(q_X) = q_Y$ if and only if $config(X)(q_X) \setminus \{P\} = config(Y)(q_Y)$ by definition of $Y = X \setminus \{P\}$. Therafter, we use the lemma 55 and get $(\mu_d(\eta_X))(q_Y) = \eta'_Y(config(Y)(q_Y))$ with $\eta'_Y = \eta'_X \setminus \{P\}$.

◀

▶ **Lemma 60** (extension of a preserving transition)**.** *Let C_Y be a configuration, P an automaton that is not contained in $\mathbf{A}_Y = auts(C_Y)$, $a \in \widehat{sig}(C_Y)$, s. t. $C_Y \overset{a}{\rightharpoonup} \eta_{Y,p}$ with $\mathbf{A}_Y$ as family support and η as companion distribution.*

Then for every $q_P \in states(P)$, for every configuration $C_X = (auts(C_Y) \cup \{P\}, maps(C_Y) \cup \{(P, q_p)\})$ we have $C_X \overset{a}{\rightharpoonup} \eta'_{X,p}$ with $\mathbf{A}_X = \mathbf{A}_Y \cup \{P\}$ as family support and η' as companion distribution where

$\eta' = \eta' \otimes \eta_{q_P,a}$ *if* $a \in \widehat{sig}(P)(q_p)$ *or* $\eta = \eta \otimes \delta_{q_P}$ *otherwise.*

Proof. Let $\mathbf{A}_Y = auts(C_Y)$ and $\mathcal{A}_Y$ the automaton issued from the composition of $mathbf{A}_Y$. Let $\mathbf{A}_X = auts(C_X) = auts(C_Y) \cup \{P\}$ and $\mathcal{A}_X$ the automaton issued from the composition of $mathbf{A}_X$.

Let (q, a, η) be transition of $\mathcal{A}_Y$, then by definition of composition, for every $q_P \in states(P)$ for the unique state q', s. t. both $q' \setminus \{P\} = q$ and $q' \upharpoonright P = q_P$. Then, by definition 11 of composition (q', a, η') is a transition of $\mathcal{A}_X$ with η' $\eta' = \eta \otimes \eta_{q_P,a}$ if $a \in \widehat{sig}(P)(q_p)$ or $\eta' = \eta \otimes \delta_{q_P}$ otherwise.

Then η' is a companion distribution of $\eta_{X,p}$, while η is a companion distribution of $\eta_{Y,p}$. ◀

▶ **Lemma 61** (extension of an intrinsinc transition)**.** *Let C_Y be a configuration, $\varphi_Y \subset Autids$, P an automaton that is not contained in $auts(C_Y) \cup \varphi_Y$, $a \in \widehat{sig}(C_Y)$, s. t. $C_Y \overset{a}{\Longrightarrow}_{\varphi_Y} \eta_Y$ where η_Y is generated by $\eta_{Y,p}$ and φ_Y where η is a companion distribution of $\eta_{Y,p}$.*

Then for every $q_P \in states(P)$, for every configuration $C_X = (auts(C_Y) \cup \{P\}, maps(C_Y) \cup \{(P, q_p)\})$, for every set φ_X, s. t. $\varphi_Y = \varphi_X \setminus \{P\}$, we have $C_X \overset{a}{\Longrightarrow}_{\varphi_X} \eta_X$ where η_X is generated by $\eta_{X,p}$ and φ_X with $\varphi_Y = \varphi_X \setminus \{P\}$ where η' is a companion distribution of $\eta_{X,p}$ with $\eta' = \eta \otimes \eta_{q_P,a}$ if $a \in \widehat{sig}(P)(q_p)$ or $\eta' = \eta \otimes \delta_{q_P}$ otherwise.

Proof. Immediate from last lemma and definition of intrinsic transition generated by a preserving transition and a set of automata ids. ◀

▶ **Lemma 62** (existence of intrinsinc transition)**.** *Let X be a PCA, $P \in Autids$ and $Y = X \setminus \{P\}$.*

$\exists y \in States(Y), \eta'_Y \in Disc(Q_{conf}, \mathcal{F}_{Q_{conf}}), a \in \widehat{sig}(Config(Y)(y)), \varphi_Y = created(Y)(y)(a)$ *s. t.*

$Config(Y)(y) \overset{a}{\Longrightarrow}_{\varphi_Y} \eta'_Y$ *implies*

It exists $\exists x \in States(X), \mu_s(x) = y, \eta'_X \in Disc(Q_{conf}, \mathcal{F}_{Q_{conf}}), \eta'_Y = (\eta'_X \setminus \{P\}), a \in \widehat{sig}(Config(X)(x) \setminus \{P\}), \varphi_X = created(X)(x)(a)$ *s. t.*

$Config(X)(x) \stackrel{a}{\Longrightarrow}_{\varphi_X} \eta'_X$.

Proof. By definition of Y, if $y \in states(Y)$, it exists $x \in states(X)$, $\mu_s(x) = y$, $config(X)(x) \setminus P = config(Y)(y)$ and $created(X)(x)(a) = created(Y)(y)(a) \setminus P$. If $P \in auts(config(X)(x))$ with $maps(config(X)(x))(P) = q_p$, we can apply the lemma 61.

We obtain $Config(X)(x) \stackrel{a}{\Longrightarrow}_{created(X)(x)(a)} \eta'_X$ and $\eta'_X = \eta'_Y \setminus P$. If $P \notin auts(config(X)(x))$, the conclusion is the same. ◀

Now we are able to demonstrate the theorem of the section that claims the PCA set is closed under projection.

▶ **Theorem 63** ($X \setminus \{P\}$ is a PCA). *Let* $P \in Autids$. *Let* X *be a* P*-fair PCA, then* $Y = X \setminus \{P\}$ *is a PCA.*

Proof.
- (Constraint 1) By definition, $config(Y)(\bar{q}_Y) = config(X)(\mu_s(\bar{q}_X))$. Since the constraint 1 is respected by X, it is a fortiori respected by Y.
- (Constraint 2) Let $(q_Y, a, \eta_Y) \in steps(Y)$. By definition of Y, we know it exists $(q_X, a, \eta_X) \in steps(X)$ with $\eta_Y = \mu_d(\eta_X)$ and $q_Y = \mu_s(q_X)$. Then, because of constraint 2 ensured by X, we obtain $config(X)(q_X) \stackrel{a}{\Longrightarrow}_{\varphi_X} \eta'_X$ with $\eta_X(q'_X) = \eta'_X(Config(X)(q'_X))$ for every $q'_X \in states(X)$, $\varphi_X = created(X)(q_X)(a)$.
  Finally, we can apply lemma 59 to obtain that $config(Y)(y) \stackrel{a}{\Longrightarrow}_{\varphi_Y} \eta'_Y$ with $\eta_Y(q'_Y) = \eta'_Y(Config(Y)(q'_Y))$ for every $q'_Y \in states(Y)$, $\varphi_Y = created(Y)(q_Y)(a)$.
- (Constraint 3) $\exists y \in States(Y), \eta'_Y \in Disc(Q_{conf}, \mathcal{F}_{Q_{conf}}), a \in \widehat{sig}(Config(Y)(y)), \varphi_Y = created(Y)(y)(a)$ s. t.
  $Config(Y)(y) \stackrel{a}{\Longrightarrow} \eta'_Y$
  Because of lemma 62, it implies it exists x, $\mu_s(x) = y$, s. t.
  $config(X)(x) \stackrel{a}{\Longrightarrow}_{\varphi_X} \eta'_X$ with $\eta'_Y = \eta'_X \setminus P$, $\varphi_X = created(X)(x)(a)$ and $\varphi_Y = \varphi_X \setminus P$.
  Since X respect the constraint 3 of PCIOA, we obtain that (x, a, η_X) exists with $\eta_X(x) = \eta'_X(config(X)(x))$.
  Then we get $(y = \mu_s(x), a, \eta_Y = \mu_d(\eta_X))$ by definition of Y.
  We can use the lemma 59 to deduce that $\eta_Y(y') = \eta'_Y(config(Y)(y'))$ for every $y' \in states(Y)$.
- (Constraint 4) By definition $sig(Y)(q_Y = \mu_s(q_X)) \triangleq hide(sig(config(Y)(q_Y), hidden\text{-}actions(Y)(q_Y))$ where $hidden\text{-}actions(Y)(q_Y) \triangleq hidden\text{-}actions(X)(q_X) \setminus out(\mathcal{A})(map(config(X)(q_X))(\mathcal{A}))$, if (*) $\mathcal{A} \in auts(config(X)(q_X))$, $hidden\text{-}actions(Y)(q_Y) \triangleq hidden\text{-}actions(X)(q_X)$ otherwise (**), Since X is supposed to be P-fair, even if it exists q'_X, s. t. $\mu_s(q'_X) = q_Y$, then $hidden\text{-}actions(X)(q_X) = hidden\text{-}actions(X)(q'_X)$, so $hidden\text{-}actions(Y)(q_Y))$ is well-defined.
  Furthermore, if (*), $hidden\text{-}actions(X)(q_X) \setminus out(\mathcal{A})(map(config(X)(q_X))(\mathcal{A})) \subseteq out(config(X)(q_X)) \setminus out(\mathcal{A})(map(config(X)(q_X))(\mathcal{A}))$ Because of compatibility of $config(X)(q_X)$, $out(\mathcal{A})(map(config(X)(q_X))(\mathcal{A})) \cap out(config(Y)(q_Y)) = \emptyset$, thus $out(config(X)(q_X)) \setminus out(\mathcal{A})(map(config(X)(q_X))(\mathcal{A})) = out(config(Y)(q_Y))$, which means $hidden\text{-}actions(Y)(q_Y)) \subseteq out(config(Y)(q_Y))$.
  otherwise (**) we have $hidden\text{-}actions(Y)(q_Y)) = hidden\text{-}actions(X)(q_X)) \subseteq out(config(X)(q_X))$ and $out(config(X)(q_X)) = out(config(Y)(q_Y))$
  Thus $hidden\text{-}actions(Y)(q_Y)) \subseteq out(config(Y)(q_Y))$

◀

6 Reconstruction

In last section, we have shown that $Y = X \setminus \mathcal{A}$ was a PCA. In this section we want to show that, (as long as no re-creation of $\mathcal{A}$ occurs), $psioa(X \setminus \{\mathcal{A}\})||\mathcal{A}$ and X are linked by an homomorphism. This concept is formalised in theorems 78 and 82. Hence it is always possible to transfer a reasoning on X into a reasoning on $psioa(X \setminus \{\mathcal{A}\})||\mathcal{A}$ if no re-creation of $\mathcal{A}$ occurs.

6.1 Simpleton wrapper

▶ **Definition 64** (Simpleton wrapper). (see figure 14) Let $\mathcal{A}$ be a PSIOA. We note $\tilde{\mathcal{A}}^{sw}$ the *simpleton wrapper* of $\mathcal{A}$ as the following PCA:

- It exists a bijection $ren_{sw} : \begin{cases} Q_{\mathcal{A}} & \rightarrow & Q_{\tilde{\mathcal{A}}^{sw}} \\ q_{\mathcal{A}} & \mapsto & \tilde{q}_{\tilde{\mathcal{A}}^{sw}} = ren_{sw}(q_{\mathcal{A}}) \end{cases}$ s. t. $psioa(\tilde{\mathcal{A}}^{sw}) = ren_{sw}(\mathcal{A})$, that is $psioa(\tilde{\mathcal{A}}^{sw})$ differs from $\mathcal{A}$ only syntactically.
- $\forall \tilde{q}_{\tilde{\mathcal{A}}^{sw}} \in states(\tilde{\mathcal{A}}^{sw}), config(\tilde{\mathcal{A}}^{sw})(\tilde{q}_{\tilde{\mathcal{A}}^{sw}}) = reduced(\{\mathcal{A}\}, \mathbf{S} : \mathcal{A} \mapsto q_{\mathcal{A}} = ren_{sw}^{-1}(q_{\mathcal{A}}))$
- $\forall \tilde{q}_{\tilde{\mathcal{A}}^{sw}} \in states(\tilde{\mathcal{A}}^{sw})$, $\forall a \in \widehat{sig}(\tilde{\mathcal{A}}^{sw})(\tilde{q}^{sw}_{\tilde{\mathcal{A}}})$, $hidden\text{-}actions(\tilde{\mathcal{A}}^{sw})(\tilde{q}_{\tilde{\mathcal{A}}^{sw}}) = \emptyset$ and $created(\tilde{\mathcal{A}}^{sw})(\tilde{q}_{\tilde{\mathcal{A}}^{sw}})(a) = \emptyset$.

We can remark that when $\tilde{\mathcal{A}}^{sw}$ enters in $\tilde{q}^{\phi}_{\tilde{\mathcal{A}}^{sw}} = ren_{sw}(q^{\phi}_{\mathcal{A}})$ where $\widehat{sig}(\tilde{\mathcal{A}}^{sw})(\tilde{q}^{\phi}_{\tilde{\mathcal{A}}^{sw}}) = \emptyset$, this matches the moment where $\mathcal{A}$ enters in $q^{\phi}_{\mathcal{A}}$ where $\widehat{sig}(\mathcal{A})(q^{\phi}_{\mathcal{A}}) = \emptyset$, s. t. the corresponding configuration is the empty one.

▶ **Lemma 65.** *Let $\mathcal{A}$ be a PSIOA. Let $\tilde{\mathcal{A}}^{sw}$ its simpleton wrapper with $psioa(\tilde{\mathcal{A}}^{sw}) = ren_{sw}(\mathcal{A})$. Let $\mu \in Disc(frags(\tilde{\mathcal{A}}^{sw}))$ $apply_{\tilde{\mathcal{A}}^{sw}}(ren_{sw}(\mu), \rho)(ren_{sw}(\alpha)) = apply_{\mathcal{A}}(\mu, \rho)(\alpha)$.*

Proof. By induction. The only key point is that (i) $\forall q \in states(\mathcal{A}), constitution(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q)) = constitution(\mathcal{A})(q)$ and (ii) for q^{ϕ} s. t. $sig(\mathcal{A})(q^{\phi}) = \emptyset$, $constitution(tilde\mathcal{A}^{sw})(ren_{sw}(q^{\phi})) = \emptyset$ which means that (*) T is enabled in q iff T is enabled in $ren_{sw}(q)$ and that (**) a is triggered by T in state q iff a is triggered by T in state $ren_{sw}(q)$.

By induction on $|\rho|$.

Basis: $apply_{\mathcal{A}}(\mu, \lambda)(\alpha) = \mu(\alpha)$, while $apply_{\tilde{\mathcal{A}}^{sw}}(ren_{sw}(\mu), \lambda)(ren_{sw}(\alpha))) = ren_{sw}(\mu)(ren_{sw}(\alpha)) = \mu(\alpha)$.

Let assume this is true for ρ_1. We consider $\alpha^{s+1} = \alpha^s \frown a^{s+1} q^{s+1}$ and $\rho_2 = \rho_1 T$.

$apply_{\mathcal{A}}(\mu, \rho_1 T)(\alpha^{s+1}) = apply_{\mathcal{A}}(apply_{\mathcal{A}}(\mu, \rho_1), T)(\alpha^{s+1}) = p_1(\alpha^{s+1}) + p_2(\alpha^{s+1})$

- $p_1(\alpha^{s+1}) = \begin{cases} apply_{\mathcal{A}}(\mu, \rho_1)(\alpha^s) \cdot \eta_{(\mathcal{A}, q^s, a^{s+1})}(q^{s+1}) & \text{if } \alpha^{s+1} = \alpha^s \frown a^{s+1} q^{s+1}, a^{s+1} \text{ triggered by } T \text{ enabled} \\ 0 & \text{otherwise} \end{cases}$
- $p_2(\alpha^{s+1}) = \begin{cases} apply_{\mathcal{A}}(\mu, \rho_1)(\alpha^{s+1}) & \text{if } T \text{ is not enabled after } \alpha^{s+1} \\ 0 & \text{otherwise} \end{cases}$

Parallely, we have

$apply_{\tilde{\mathcal{A}}^{sw}}(ren_{sw}(\mu), \rho_1 T)(ren_{sw}(\alpha^{s+1})) = apply_{\tilde{\mathcal{A}}^{sw}}(apply_{\tilde{\mathcal{A}}^{sw}}(ren_{sw}(\mu), \rho_1), T)(ren_{sw}(\alpha^{s+1})) = p'_1(ren_{sw}(\alpha^{s+1})) + p'_2(ren_{sw}(\alpha^{s+1}))$

- $p'_1(ren_{sw}(\alpha^{s+1})) = \begin{cases} apply_{\tilde{\mathcal{A}}^{sw}}(ren_{sw}(\mu), \rho_1)(ren_{sw}(\alpha^s)) \cdot \eta_{(\tilde{\mathcal{A}}^{sw}, ren_{sw}(q^s), a^{s+1})}(ren_{sw}(q^{s+1})) & \text{if } (**) \\ 0 & \text{otherwise} \end{cases}$
- $p'_2(ren_{sw}(\alpha^{s+1})) = \begin{cases} apply_{\tilde{\mathcal{A}}^{sw}}(ren_{sw}(\mu), \rho_1)(ren_{sw}(\alpha^{s+1})) & \text{if } T \text{ is not enabled after } ren_{sw}(\alpha^{s+1}) \\ 0 & \text{otherwise} \end{cases}$

with $(**)$: $ren_{sw}(\alpha^{s+1}) = ren_{sw}(\alpha^s)a^{s+1}ren_{sw}(q^{s+1}), a^{s+1}$ triggered by T.

We have : T enabled after $\alpha \iff T$ enabled after $ren_{sw}(\alpha)$. The leftward terms are equal by induction hypothesis, since $|\rho_1| = |\rho_2| - 1$. Since the probabilistic distributions are in bijection we can obtain the equality for rightward terms. The conditions are matched in the same manner because of signature bijection a. Thus we can conclude that $p'_1(ren_{sw}(\alpha^{s+1})) = p_1(\alpha^{s+1})$ and $p'_2(ren_{sw}(\alpha^{s+1})) = p_2(\alpha^{s+1})$, which leads to the result.

◀

6.2 Partial-compatibility

In this section, we show that $(X_{\mathcal{A}} \setminus \{\mathcal{A}\})$ and $\tilde{\mathcal{A}}^{sw}$ are partially-compatible and that $(X_{\mathcal{A}} \setminus \{\mathcal{A}\})||\tilde{\mathcal{A}}^{sw}$ mimics $X_{\mathcal{A}}$ as long as no creation of $\mathcal{A}$ occurs (see figure 15).

In this subsection we show that $psioa(X \setminus \{\mathcal{A}\})$ and $\mathcal{A}$ are partially-compatible if minor conditions are respected. We will use the notation $\mathbf{Z} = (psioa(X \setminus \{\mathcal{A}\}), \mathcal{A})$ and in case of partiall-compatibility of $\mathbf{Z}$, $\mathcal{Z} = psioa(X \setminus \{\mathcal{A}\})||\mathcal{A}$.

▶ **Definition 66** ($\mathcal{A}$-conservative PCA)**.** Let X be a PCA, $\mathcal{A} \in Autids$. We say that X is $\mathcal{A}$-conservative if it is $\mathcal{A}$-fair and for every state q_X, $C_x = config(X)(q_X)$ s. t. $\mathcal{A} \in aut(C_X)$ and $map(C_X)(\mathcal{A}) \triangleq q_{\mathcal{A}}$, $hidden\text{-}actions(X)(q_X) = hidden\text{-}actions(X)(q_X) \setminus \widehat{ext}(\mathcal{A})(q_{\mathcal{A}})$.

A $\mathcal{A}$-conservative PCA is a PCA that does not hide any output action that could be an external action of $\mathcal{A}$. This allows the compatibility between $X \setminus \mathcal{A}$ and $\mathcal{A}$.

This allows the compatibility between $X \setminus \mathcal{A}$ and $\tilde{\mathcal{A}}^{sw}$.

▶ **Definition 67** ($\mu_z^{\mathcal{A}}$ and $\mu_e^{\mathcal{A}}$ mapping)**.** Let $\mathcal{A} \in Autids$, X be a $\mathcal{A}$-fair PCA, $Y = X \setminus \mathcal{A}$. Let $\tilde{\mathcal{A}}^{sw}$ be the simpleton wrapper of $\mathcal{A}$, where $psioa(\tilde{\mathcal{A}}^{sw}) = ren_{sw}(\mathcal{A})$. Let $q^{\phi}_{\mathcal{A}} \in states(\mathcal{A})$ the (assumed) unique state s. t. $\widehat{sig}(\mathcal{A})(q^{\phi}_{\mathcal{A}}) = \emptyset$. We note $\mu_z^{\mathcal{A}} : states(X) \rightarrow states(Y) \times states(\tilde{\mathcal{A}}^{sw})$ s. t. $\forall x \in states(X)$, $\mu_z^{\mathcal{A}}(x) = (\mu_s^{\mathcal{A}}(x), ren_{sw}(q_{\mathcal{A}}))$ with $q_{\mathcal{A}} = map(config(X)(x))(\mathcal{A})$ if $\mathcal{A} \in (auts(config(X)(x)))$ and $q_{\mathcal{A}} = q^{\phi}_{\mathcal{A}}$ otherwise.

For every alternating sequence $\alpha = x^0, a^1, s^1, a^2...$ of states of and actions of X α_X, we note $\mu_e^{\mathcal{A}}(\alpha_X)$ the alternating sequence $\alpha = \mu_z^{\mathcal{A}}(x^0), a^1, \mu_z^{\mathcal{A}}(x^1), a^2, ...$

The symbol $\mathcal{A}$ is omitted when this is clear in the context.

▶ **Lemma 68** (preservation of signature compatibility of configurations)**.** *Let $\mathcal{A} \in Autids$. Let X be a $\mathcal{A}$-conservative PCA, $Y = X \setminus \mathcal{A}$. Let $q_X \in states(X)$, $C_X = config(X)(q_X)$, $\mathbf{A}_X = aut(C_X)$, $\mathbf{S}_X = map(C_X)$.*

If $\mathcal{A} \in \mathbf{A}_X$ and $q_{\mathcal{A}} = \mathbf{S}_X(\mathcal{A})$, then $sig(C_Y)$ and $sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_{\mathcal{A}}))$ are compatible and $sig(C_X) = sig(C_Y) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_{\mathcal{A}}))$.

If $\mathcal{A} \notin \mathbf{A}_X$, then $sig(C_Y)$ and $sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q^{\phi}_{\mathcal{A}}))$ are compatible and $sig(C_X) = sig(C_Y) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q^{\phi}_{\mathcal{A}}))$.

Proof. Let $\mathcal{A} \in Autids$ Let X and $Y \setminus \{\mathcal{A}\}$ be PCA. Let $q_X \in states(X)$. Let $C_X = config(X)(q_X)$, $\mathbf{A}_X = auts(C_X)$ and $\mathbf{S}_X = map(C_X)$. Let $q_Y \in states(Y), q_Y = \mu_s(q_X)$. Let $C_Y = config(Y)(q_Y)$, $\mathbf{A}_Y = auts(C_Y)$ and $\mathbf{S}_Y = map(C_Y)$. By definition of Y, $C_Y = C_X \setminus \{\mathcal{A}\}$.

Case 1: $\mathcal{A} \in \mathbf{A}_X$

Since X is a PCA, C_X is a compatible configuration, thus $((\mathbf{A}_Y, \mathbf{S}_Y) \cup (\mathcal{A}, q_\mathcal{A}))$ is a compatible configuration. Finally $sig(C_Y)$ and $sig(\mathcal{A})(q_\mathcal{A})$ are compatible with $sig(\mathcal{A})(q_\mathcal{A}) = sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q^{\phi}_{\mathcal{A}}))$.

By definition of intrinsinc attributes of a configuration, that are constructed with the attributes of the automaton issued from the composition of the family of automata of the configuration, we have $\mathbf{A}_X = \mathbf{A}_Y \cup \{\mathcal{A}\}$ and $sig(C_X) = sig(C_Y) \times sig(\mathcal{A})(q_\mathcal{A})$, that is $sig(C_X) = sig(C_Y) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_\mathcal{A}))$.

Case 2: $\mathcal{A} \notin \mathbf{A}_X$

Since X is a PCA, C_X is a compatible configuration, thus $C_Y = C_X$ is a compatible configuration. Finally $sig(C_Y)$ and $sig(\mathcal{A})(q^{\phi}_{\mathcal{A}}) = (\emptyset, \emptyset, \emptyset) = sig(\mathcal{A})(q_\mathcal{A}) = sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q^{\phi}_{\mathcal{A}}))$ are compatible.

By definition of intrinsinc attributes of a configuration, that are constructed with the attributes of the automaton issued from the composition of the family of automata of the configuration (here $\mathbf{A}_Y$ and $\mathbf{A}_X = \mathbf{A}_Y$), we have $sig(C_X) = sig(C_Y)$. Furthermore, $sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q^{\phi}_{\mathcal{A}})) = sig(\mathcal{A})(q^{\phi}_{\mathcal{A}}) = (\emptyset, \emptyset, \emptyset)$. Thus $sig(C_X) = sig(C_Y) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q^{\phi}_{\mathcal{A}}))$ ◀

▶ **Lemma 69** (preservation of signature). *Let $\mathcal{A} \in Autids$. Let X be a $\mathcal{A}$-conservative PCA, $\mathcal{A} \in Autids$, $Y = X \setminus \{\mathcal{A}\}$. For every $q_X \in states(X)$, we have $sig(X)(q_X) = sig(Y)(q_Y) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_\mathcal{A}))$ with $(q_Y, ren_{sw}(q_\mathcal{A})) = \mu^{\mathcal{A}}_z(q_X)$.*

Proof. The last lemma 68 tell us for every $q_X \in states(X)$, we have $sig(config(X)(q_X)) = sig(config(Y)(q_Y)) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_\mathcal{A}))$ with $(q_Y, ren_{sw}(q_\mathcal{A})) = \mu_z(q_X)$. Since X is $\mathcal{A}$-conservative, we have (*) $sig(X)(q_X) = hide(sig(config(X)(q_X)), \underline{acts})$ where $\underline{acts} \subseteq (out(X)(q_X) \setminus (ext(\mathcal{A})(q_\mathcal{A}))$. Hence $sig(Y)(q_Y) = hide(sig(config(Y)(q_Y)), \underline{acts})$. Since (**) $\underline{acts} \cap ext(\mathcal{A})(q_\mathcal{A}) = \emptyset$, $sig(Y)(q_Y)$ and $sig(\mathcal{A})(q_\mathcal{A})$ are also compatible. We have $sig(config(X)(q_X)) = sig(config(Y)(q_Y)) \times sig(\mathcal{A})(q_\mathcal{A}) = sig(config(Y)(q_Y)) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_\mathcal{A}))$ which gives because of (*) $hide(sig(config(X)(q_X)), \underline{acts}) = hide(sig(config(Y)(q_Y)), \underline{acts}) \times sig(\mathcal{A})(q_\mathcal{A})$, that is $sig(X)(q_X) = sig(Y)(q_Y) \times sig(\mathcal{A})(q_\mathcal{A}) = sig(Y)(q_Y) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_\mathcal{A}))$.

◀

▶ **Lemma 70** (preservation of partial-compatibility at any reachable state). *Let $\mathcal{A} \in Autids$, X be a $\mathcal{A}$-conservative PCA, $Y = X \setminus \{\mathcal{A}\}$, $\mathbf{Z} = (psioa(Y), \tilde{\mathcal{A}}^{sw})$ Let $z = (y, \tilde{q}_{\tilde{\mathcal{A}}^{sw}}) \in states(Y) \times states(\tilde{\mathcal{A}}^{sw})$ and $x \in states(X)$ s. t. $\mu_z(x) = z$. Then $\mathbf{Z}$ is partially compatible at state z (in the sense of definition 43).*

Proof. Since X is a $\mathcal{A}$-conservative PCA, the previous lemma 69 ensures that $sig(Y)(y)$ and $sig(\mathcal{A})(q_\mathcal{A}) = sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_\mathcal{A}))$ are compatible, thus by definition $\mathbf{Z}$ is partially compatible at state z. ◀

We show that reconstruction preserves probabilistic distribution of corresponding transition.

▶ **Lemma 71** (preservation of transition). *Let $\mathcal{A} \in Autids$, X be a $\mathcal{A}$-conservative PCA, $Y = X \setminus \{\mathcal{A}\}$, $\mathbf{Z} = (Y, \tilde{\mathcal{A}}^{sw})$. Let $q_Z = (q_Y, \tilde{q}_{\tilde{\mathcal{A}}^{sw}}) \in states(Y) \times states(\tilde{\mathcal{A}}^{sw})$ and $q_X \in states(X)$ s. t. $\mu_z(q_X) = q_Z$. Let $a \in sig(X)(x) = sig(Y)(y) \times sig(\tilde{\mathcal{A}}^{sw})(\tilde{q}_{\tilde{\mathcal{A}}^{sw}})$, verifying*

- *(No creation from $\mathcal{A}$) If both $\mathcal{A} \in map(config(X)(q_X))$ and $a \notin sig((config(X)(q_X) \setminus \mathcal{A}))$,then $created(X)(x)(a) = \emptyset$*

If we are in one of this case

1. $\mathcal{A} \in auts(config(X)(x))$
2. $\mathcal{A} \notin auts(config(X)(x))$ *and* $\mathcal{A} \notin created(X)(x)(a)$ *(X does not create $\mathcal{A}$ with probability 1)*

Then for every $q'_X \in states(X)$, $\eta_{(X,q_X,a)}(q'_X) = \eta_{(\mathbf{Z},q_z,a)}(\mu_z(q'_X))$.

Proof. By lemma 69, we have $sig(X)(q_X) = sig(Y)(q_Y) \times sig(\mathcal{A})(q_\mathcal{A}) = sig(Y)(y) \times sig(\tilde{\mathcal{A}}^{sw})(\tilde{q}_{\tilde{\mathcal{A}}^{sw}} = ren_{sw}(q_\mathcal{A}))$.

We note $\varphi_X = created(X)(q_X)(a)$, $\varphi_Y = created(X)(q_X)(a) \setminus \mathcal{A}$. We note $\mathbf{A}_X = auts(config(X)(q_X))$, $\mathbf{A}_Y = auts(config(Y)(q_Y))$, $\mathbf{S}_X = map(config(X)(q_X))$, $\mathbf{S}_Y = map(config(Y)(q_Y))$, $\mathcal{A}_X$ (resp. $\mathcal{A}_Y$) the composition of automata in $\mathbf{A}_X$ (resp. $\mathbf{A}_Y$).

If $a \notin sig(config(X)(q_X) \setminus \mathcal{A}) \wedge a \in sig(\mathcal{A})(q_\mathcal{A})$, then $\varphi_X = \varphi_Y = \emptyset$.

Since X (resp. Y) is a PCA and $(q_X, a, \eta_{(X,q_X,a)}) \in D_X$ (resp. if $a \in sig(Y)(q_Y)$, $(q_Y, a, \eta_{(Y,q_Y,a)}) \in D_X$) the constraint says that it exists $\eta_{(C_X,a)}$ (resp. $\eta_{(C_Y,a)}$) reduced configuration distribution s. t. $config(X)(q_X) \Longrightarrow_{\varphi_X} \eta_{(C_X,a)}$ (resp. $config(Y)(q_Y) \Longrightarrow_{\varphi_Y} \eta_{(C_Y,a)}$) where for every $q'_X \in states(X)$, $\eta_{(C_X,a)}(config(X)(q'_X)) = \eta_{(X,q_X,a)}(q'_X)$ (resp. $q'_Y \in states(Y)$, $\eta_{(C_Y,a)}(config(Y)(q'_Y)) = \eta_{(Y,q_Y,a)}(q'_Y)$) and $\eta_{(C_X,a)}$ (resp. $\eta_{(C_Y,a)}$) generated from φ_X (resp. φ_Y) and $\eta_{(C_X,a),p}$ (resp. $\eta_{(C_Y,a),p}$) with companion distribution $\eta_{(\mathbf{A}_X,q_X,a)} \in Disc(Q_{\mathbf{A}_X})$ (resp. $\eta_{(\mathbf{A}_Y,q_Y,a)} \in Disc(Q_{\mathbf{A}_Y})$).

If $a \in sig(\mathcal{A})(q_\mathcal{A})$, it exists $\eta_{(\mathcal{A},q_\mathcal{A},a)} \in Disc(Q_\mathcal{A})$, $(q_\mathcal{A}, a, \eta_{(\mathcal{A},q_\mathcal{A},a)}) \in D_\mathcal{A}$. By construction of $Y = X \setminus \{\mathcal{A}\}$, if $\mathcal{A} \in \mathbf{A}_X$, $\eta_{(\mathbf{A}_X,q_X,a)} = \eta_{(\mathbf{A}_Y,q_Y,a)} \otimes \eta_{(\mathcal{A},q_\mathcal{A},a)}$ and otherwise $\eta_{(\mathbf{A}_X,q_X,a)} = \eta_{(\mathbf{A}_Y,q_Y,a)}$. Finally, also by construction of $Y = X \setminus \{\mathcal{A}\}$ we know that for every $a \in sig(Y)(q_Y)$, for every $q'_X \in states(X)$, $\eta_{(X,q_X,a)}(q'_X) = \eta_{(Y,q_Y,a)}(\mu_s(q'_X))$.

1. $\mathcal{A} \in auts(config(X)(x))$. We know that $\eta_{\mathbf{A}_X,q_X,a} = \eta_{\mathbf{A}_Y,q_Y,a} \otimes \eta_{(\mathcal{A},q_\mathcal{A},a)}$. This means that for every configuration $C'_X = C'_Y \cup C'_\mathcal{A}$ with $C'_X = (\mathbf{A}_X, \mathbf{S}'_X)$, $C'_Y = (\mathbf{A}_Y \mathbf{S}'_Y)$, $C'_\mathcal{A} = (\mathcal{A}, \{(\mathcal{A}, q'_\mathcal{A})\})$, $\eta_{(C_X,a),p}(C'_X) = (\eta_{(C_Y,a),p} \otimes \eta_{(\mathcal{A},q_\mathcal{A},a)}))(C'_Y, C'_\mathcal{A})$. Since we assume no creation from $\mathcal{A}$, we also have for every configuration $C''_X = C''_Y \cup C''_\mathcal{A}$ with $C''_X = (\mathbf{A}''_X, \mathbf{S}''_X)$, $C''_Y = (\mathbf{A}''_Y \mathbf{S}''_Y)$, $C''_\mathcal{A} = (\mathcal{A}, q''_\mathcal{A})$, $\eta_{(C_X,a)}(C''_X) = (\eta_{(C_Y,a)} \otimes \eta_{(\mathcal{A},q_\mathcal{A},a)}))(C''_Y, C''_\mathcal{A})$. Hence for every states $q''_X, q''_Z = (q''_Y, q''_\mathcal{A}) = \mu_z(q''_X)$, $\eta_{(X,q_X,a)}(q''_X) = (\eta_{(Y,q_Y,a)} \otimes \eta_{(\mathcal{A},q_\mathcal{A},a)}))(q''_Y, q''_\mathcal{A}) = (\eta_{(Y,q_Y,a)} \otimes \eta_{(ren_{sw}(\mathcal{A}),ren_{sw}(q_\mathcal{A}),a)})))(q''_Y, ren_{sw}(q''_\mathcal{A})) = \eta_{(Z,q_Z,a)(\mu_z(q''_X))}$, which ends the proof for this case.
2. $\mathcal{A} \notin auts(config(X)(q_X))$ and $\mathcal{A} \notin created(X)(x)(a)$. In this case $\varphi_X = \varphi_Y$ because we assume no creation of $\mathcal{A}$ and we obtain $\eta_{(C_X,a)} = \eta_{(C_Y,a)}$. Furthermore, $q_\mathcal{A} = q^\phi_\mathcal{A}$ and thus $a \notin \widehat{sig}(\mathcal{A})(q_\mathcal{A})$, i. e. $\eta_{(\mathbf{Z},q_Z,a)}(\mu_z(q'_X)) = (\eta_{(Y,q_Y,a)} \otimes \delta_{ren_{sw}(q_\mathcal{A},a)})(q''_Y, ren_{sw}(q''_\mathcal{A})) = (\eta_{(Y,q_Y,a)} \otimes \delta_{q^\phi_\mathcal{A}})(\mu_s(q'_X), q^\phi_\mathcal{A}) = \eta_{(Y,q_Y,a)}(\mu_s(q'_X)) = \eta_{(X,q_X,a)}(q'_X)$ which ends the proof for this case.

◀

▶ **Definition 72** ($\mathcal{A}$-twin)**.** Let $\mathcal{A} \in Autids$. Let X, X' be PCA. We say that X' is a $\mathcal{A}$-twin of X if it differs from X at most only by its start states $\bar{q}_{X'}$ reachable by X s. t. $\mathcal{A} \in config(X')(\bar{q}_{X'})$ and $map(config(X')(\bar{q}_{X'}))(\mathcal{A}) = \bar{q}_\mathcal{A}$. If X' is a $\mathcal{A}$-twin of X and $Y = X \setminus \mathcal{A}$ and $Y' = X' \setminus \mathcal{A}$, we slightly abuse the notation and say that Y' is a $\mathcal{A}$-twin of Y'.

▶ **Lemma 73** (0-partial-compatibility after reconstruction)**.** *Let* $\mathcal{A} \in Autids$*. Let* X *be a PCA* $\mathcal{A}$*-conservative. Let* $Y = X \setminus \mathcal{A}$*. Let* Y' *be a* $\mathcal{A}$*-twin of* Y*.*

Then Y' and $\tilde{\mathcal{A}}^{sw}$ are 0-partially-compatible (In the sense of definition 46).

Proof. Since $q_X \in states(X)$ and X is a PCA, $C_X \triangleq config(X)(q_X)$ is a compatible configuration by definition, which implies $sig(config(Y)(q_{Y'}))$ and $sig(ren_{sw}(\mathcal{A}))(ren_{sw}(\bar{q}_{\mathcal{A}}))$ are compatible signatures and equally for $sig(config(Y')(\bar{q}_{Y'}))$ and $sig(ren_{sw}(\mathcal{A}))(ren_{sw}(\bar{q}_{\mathcal{A}}))$. Since X is $\mathcal{A}$-conservative, $sig(Y')(\bar{q}_{Y'})$ and $sig(\mathcal{A})(\bar{q}_{\mathcal{A}}) = sig(ren_{sw}(\mathcal{A}))(ren_{sw}(\bar{q}_{\mathcal{A}}))$ are compatible signatures. (a compatible output of $sig(config(X)(q_X))$ cannot become an internal action of $sig(Y')(\mu_s(q_X))$ non-compatible with $sig(\mathcal{A})(map(C_X)(\mathcal{A}))$). ◀

▶ **Lemma 74** (partial surjectivity 1)**.** *Let $\mathcal{A} \in Autids$. Let X be a PCA $\mathcal{A}$-conservative. Let $Y = X \setminus \mathcal{A}$. Let Y' be a $\mathcal{A}$-twin of Y. Let $\mathbf{Z} = (Y', \tilde{\mathcal{A}}^{sw})$.*

Let $\alpha = q^0, a^1, ..., a^k, q^k$ be a pseudo execution of $\mathbf{Z}$. Let assume $q^s_{\tilde{\mathcal{A}}^{sw}} \neq ren_{sw}(q^{\phi}_{\mathcal{A}})$ for every $s \in [0, k]$. Then it exists $\tilde{\alpha} \in frags(X)$, s. t. $\mu_e(\tilde{\alpha}) = \alpha$. If $Y' = Y$, it exists $\tilde{\alpha} \in execs(X)$, s. t. $\mu_e(\tilde{\alpha}) = \alpha$.

Proof. By induction on each prefix $\alpha^s = q^0, a^1, ..., a^s, q^s$ with $s \leq k$.

Basis: For $Y = Y'$, $\mu_z(\bar{q}_X) = (\bar{q}_Y, ren_{sw}(\bar{q}_{\mathcal{A}}))$ For $Y \neq Y'$, it exists q'_X s. t. $\mu_z(q'_X) = (\bar{q}_{Y'}, ren_{sw}(\bar{q}_{\mathcal{A}}))$ by definition of $\mathcal{A}$-twin. Hence $\mu_e(q'_X) = (\bar{q}_{Y'}, ren_{sw}(\bar{q}_{\mathcal{A}}))$

Induction: we assume this is true for s and we show it implies this true for $s+1$. We note $\tilde{\alpha}_s$ s. t. $\mu_e(\tilde{\alpha}^s) = \alpha^s$. We also note $\tilde{q}^s = lstate(\tilde{\alpha}^s)$ and we have by induction assumption $\mu_z(\tilde{q}^s) = q^s = (q^s_Y, q^s_{\mathcal{A}})$. Because of preservation of signature compatibility, $sig(X)(\tilde{q}^s)) = sig(Y)(q^s_Y)) \times sig(ren_{sw}(\mathcal{A}))(q^s_{ren_{sw}(\mathcal{A})}))$. Hence $a^{k+1} \in sig(X)(\tilde{q}^s)$. Finally we can use preservation of transition since no creation of $\mathcal{A}$ can occur to conclude. ◀

▶ **Theorem 75** (Partial-compatibility after resconstruction)**.** *Let $\mathcal{A} \in Autids$. Let X be a PCA $\mathcal{A}$-conservative. Let $Y = X \setminus \mathcal{A}$. Let Y' be a $\mathcal{A}$-twin of Y. Let $\mathbf{Z} = (Y', \tilde{\mathcal{A}}^{sw})$. Then Y' and $\tilde{\mathcal{A}}^{sw}$ are partially-compatible.*

Proof. Let $q_{\mathbf{Z}} = (q_{Y'}, q_{\tilde{\mathcal{A}}^{sw}})$ be a reachable state of $\mathbf{Z}$. Case 1) $q_{\tilde{\mathcal{A}}^{sw}} = q^{\phi}_{\tilde{\mathcal{A}}^{sw}}$. The compatibility is immediate since $sig(\tilde{\mathcal{A}}^{sw})(q^{\phi}_{\tilde{\mathcal{A}}^{sw}}) = \emptyset$. Case 2) $q_{\tilde{\mathcal{A}}^{sw}} \neq q^{\phi}_{\tilde{\mathcal{A}}^{sw}}$. Since $q_{\tilde{\mathcal{A}}^{sw}}$ is reachable, it exists a pseudo execution α of $\mathbf{Z}$ with $lstate(\alpha) = q_{\tilde{\mathcal{A}}^{sw}}$. Since $\mathcal{A}$ cannot be re-created after destruction by neither Y or $\tilde{\mathcal{A}}^{sw}$ we can use the previous lemma to show it exists $\tilde{\alpha} \in frags(X)$, s. t. $\mu_e(\tilde{\alpha}) = \alpha$. Thus, $lstate(\alpha) = \mu_z(lstate(\tilde{\alpha}))$ which means $\mathbf{Z}$ is partially-compatible at $lstate(\alpha)$. Hence $\mathbf{Z}$ is partially-compatible at every reachable state, which means Y' and $\tilde{\mathcal{A}}^{sw}$ are partially-compatible. We can legitimately note $\mathcal{Z}' = Y'||\tilde{\mathcal{A}}^{sw}$. ◀

Since $\mathbf{Z}' = (Y', \tilde{\mathcal{A}}^{sw})$ is partially-compatible, we can legitimately note $\mathcal{Z}' = Y'||\tilde{\mathcal{A}}^{sw}$, which will be the standard notation in the remaining.

6.3 Probabilisitc distribution preservation without creation

▶ **Lemma 76** (partial surjectivity 2)**.** *Let $\mathcal{A} \in Autids$. Let X be a PCA $\mathcal{A}$-conservative. Let $Y = X \setminus \mathcal{A}$. Let Y' be a $\mathcal{A}$-twin of Y. Let $\mathcal{Z} = Y'||\tilde{\mathcal{A}}^{sw}$.*

Let $\alpha = q^0, a^1, ..., a^k, q^k$ be a an execution of $\mathcal{Z}$. Let assume (a) $q^s_{\tilde{\mathcal{A}}^{sw}} \neq ren_{sw}(q^{\phi}_{\mathcal{A}})$ for every $s \in [0, k^*]$ (b) $q^s_{\tilde{\mathcal{A}}^{sw}} = q^{\phi}_{\tilde{\mathcal{A}}^{sw}}$ for every $s \in [k^*+1, k]$ (c) for every $s \in [k^*+1, k-1]$, for every $\tilde{q}^s$, s. t. $\mu_z(\tilde{q}^s) = q^s$, $\mathcal{A} \notin created(X)(\tilde{q}^s)(a^{s+1})$. Then it exists $\tilde{\alpha} \in frags(X)$, s. t. $\mu_e(\tilde{\alpha}) = \alpha$. If $Y' = Y$, it exists $\tilde{\alpha} \in execs(X)$, s. t. $\mu_e(\tilde{\alpha}) = \alpha$.

Proof. We already know this is true up to k^* because of lemma 74. We perform the same induction than the one of the previous lemma on partial surjectivity: We note $\tilde{\alpha}_s$ s. t. $\mu_e(\tilde{\alpha}^s) = \alpha^s$. We also note $\tilde{q}^s = lstate(\tilde{\alpha}^s)$ and we have by induction assumption $\mu_z(\tilde{q}^s) = q^s = (q^s_Y, q^s_{\mathcal{A}})$. Because of preservation of signature compatibility, $sig(X)(\tilde{q}^s)) = sig(Y)(q^s_Y)) \times sig(ren_{sw}(\mathcal{A}))(ren_{sw}(q^s_{\mathcal{A}}))$. Hence $a^{k+1} \in sig(X)(\tilde{q}^s)$. Now we use the assumption (c), that says that $\mathcal{A} \notin created(X)(\tilde{q}^s)(a^{s+1})$ to be able to apply preservation of transition since no creation of $\mathcal{A}$ can occurs. ◀

▶ **Lemma 77.** *Let $\mathcal{A} \in Autids$. Let X be a PCA $\mathcal{A}$-conservative. Let $Y = X \setminus \mathcal{A}$. Let Y' be a $\mathcal{A}$-twin of Y. Let $\mathbf{Z}' = (Y', \tilde{\mathcal{A}}^{sw})$.*

1. *Y' and $\tilde{\mathcal{A}}^{sw}$ are partially-compatible, thus we can legitimately note $\mathcal{Z}' = Y'||\tilde{\mathcal{A}}^{sw}$.*
2. *Furthermore, for every execution fragment $\alpha \in frags(X)$, with $\mu_z(fstate(\alpha)) \in states(\mathcal{Z}')$ verifying*
   - *No creation of $\mathcal{A}$: If $\mathcal{A} \notin auts(config(X)(q^s_X))$ then $\mathcal{A} \notin created(X)(q^s_X)(a^{s+1})$.*
   - *No creation from $\mathcal{A}$: $\forall s$, verifying $a^{s+1} \notin sig(config(X)(q^s_X) \setminus \mathcal{A}) \wedge a^{s+1} \in sig(\tilde{\mathcal{A}}^{sw})(q^s_{\tilde{\mathcal{A}}^{sw}})$, with $\mu_z(q^s_X) = q_{\mathcal{Z}} = (q^s_Y, q^s_{\tilde{\mathcal{A}}^{sw}})$, $created(X)(q^s_X)(a) = \emptyset$.*

   then $\mu_e(\alpha) \in frags(\mathcal{Z})$.

Proof. By induction on the size s of a prefix α^s of α. Basis: The result is immediate by assumption for $\alpha^s = q^0_X$, since $\mu_z(q^0_X)$ is assumed to be a state of $\mathcal{Z}$. Induction: We assume this is true for α^s and we want to show this is also true for $\alpha^{s+1} = \alpha^s \frown a^{s+1} q^{s+1}$. We have signature preservation for q^s and $\mu_z(q^s)$, thus $a^{s+1} \in sig(\mathcal{Z})$. Moreover, we have transition preservation, thanks to the assumptions, thus $\mu_z(q^{s+1}) \in supp(\eta_{\mathcal{Z},\mu_z(q^s),a)})$ which means that $\mu_e(\alpha^{s+1})$ is an execution of α^{s+1}, this ends the induction and the proof. ◀

▶ **Theorem 78** (Preserving probabilistic distribution without creation)**.** *Let $\mathcal{A} \in Autids$. Let X be a $\mathcal{A}$-conservative PCA. Let $Y = X \setminus \mathcal{A}$. Let Y' be a $\mathcal{A}$-twin of Y. Let $\mathcal{Z}' = Y'||\tilde{\mathcal{A}}^{sw}$. Let $\mathcal{E}$ be an environment of X. Let ρ be a schedule.*

For every execution fragment $\alpha = q^0 a^1 q^1 ... q^k \in frags(X||\mathcal{E})$ with $\mu_z(q^0) \in states(\mathcal{Z})$, verifying:

- *No creation of $\mathcal{A}$: For every $s \in [0, k-1]$, if $\mathcal{A} \notin auts(config(X)(q^s_X))$ then $\mathcal{A} \notin created(X)(q^s_X)(a^{s+1})$.*
- *No creation from $\mathcal{A}$: $\forall s \in [0, k-1]$, verifying $a^{s+1} \notin sig(config(X)(q^s_X) \setminus \mathcal{A}) \wedge a^{s+1} \in sig(\tilde{\mathcal{A}}^{sw})(q^s_{\tilde{\mathcal{A}}^{sw}}$, with $\mu_z(q^s_X) = q_{\mathcal{Z}'} = (q^s_{Y'}, q^s_{\tilde{\mathcal{A}}^{sw}})$, $created(X)(q^s_X)(a) = \emptyset$.*

then for every $q_X \in states(X)$ s. t. $\mu_z(q_X) \in states(\mathcal{Z}')$, $apply_{X||\mathcal{E}}(\delta_{(q_X, q_{\mathcal{E}})}, \rho)(\alpha) = apply_{(\mathcal{Z}'||\mathcal{E})}(\delta_{(\mu_z(q_X), q_{\mathcal{E}})}, \rho)(\mu_e(\alpha))$.

Proof. We recall that for every $s \in [0, k-1]$, if $(q^s_{\mathcal{Z}'}, q^s_{\mathcal{E}}) = (\mu_z(q^s_X), q^s_{\mathcal{E}})$, $\eta_{(X, q^s_X, a^{s+1})}(q^{s+1}_X) = \eta_{(\mathcal{Z}', q^s_{\mathcal{Z}'}, a^{s+1})}(\mu_z(q^{s+1}_X))$, since $q^s_{\mathcal{Z}'} = \mu_z(q^s_X)$. Hence $\eta_{(X, q^s_X, a^{s+1})}(q^{s+1}_X) \otimes \eta_{(\mathcal{E}, q^s_{\mathcal{E}}, a^{s+1})}(q^{s+1}_{\mathcal{E}}) = \eta_{(\mathcal{Z}', q^s_{\mathcal{Z}'}, a^{s+1})}(\mu_z(q^{s+1}_X)) \otimes \eta_{(\mathcal{E}, q^s_{\mathcal{E}}, a^{s+1})}(q^{s+1}_{\mathcal{E}})$, which gives $\eta_{(X||\mathcal{E}, (q^s_X, q^s_{\mathcal{E}}), a^{s+1})}((q^{s+1}_X, q^{s+1}_{\mathcal{E}})) = \eta_{(\mathcal{Z}'||\mathcal{E}, (q^s_{\mathcal{Z}'}, q^s_{\mathcal{E}}), a^{s+1})}((\mu_z(q^{s+1}_X), q^{s+1}_{\mathcal{E}}))$ and finally $\eta_{(X||\mathcal{E}, q^s, a^{s+1})}(q^{s+1})) = \eta_{(\mathcal{Z}'||\mathcal{E}, q^s, a^{s+1})}(\mu_z(q^{s+1}))$.

By induction on $|\rho|$.

Basis: $apply_{X||\mathcal{E}}(\delta_{(q_X, q_{\mathcal{E}})}, \lambda) = \delta_{(q_X, q_{\mathcal{E}})}$, while $apply_{\mathcal{Z}'||\mathcal{E}}(\delta_{\mu_z(q_X), q_{\mathcal{E}}}, \lambda) = \delta_{(\mu_z(q_X), q_{\mathcal{E}})}$ and $\mu_e((q_X, q_{\mathcal{E}})) = (\mu_z(q_X), q_{\mathcal{E}})$.

Let assume this is true for ρ_1. We consider $\alpha^{s+1} = \alpha^s \frown a^{s+1} q^{s+1}$ and $\rho_2 = \rho_1 T$.

$apply_{X||\mathcal{E}}(\delta_{(q_X,q_\mathcal{E})}, \rho_1 T)(\alpha^{s+1}) = apply_{X||\mathcal{E}}(apply_{X||\mathcal{E}}(\delta_{(q_X,q_\mathcal{E})}, \rho_1), T)(\alpha^{s+1}) = p_1(\alpha^{s+1})+ p_2(\alpha^{s+1})$

- $p_1(\alpha^{s+1}) = \begin{cases} apply_{X||\mathcal{E}}(\delta_{(q_X,q_\mathcal{E})}, \rho_1)(\alpha^s) \cdot \eta^X(q^{s+1}) & \text{if } \alpha^{s+1} = \alpha^s \frown a^{s+1} q^{s+1}, a^{s+1} \text{ triggered by } T \text{ enabled} \\ 0 & \text{otherwise} \end{cases}$
- $p_2(\alpha^{s+1}) = \begin{cases} apply_{X||\mathcal{E}}(\delta_{(q_X,q_\mathcal{E})}, \rho_1)(\alpha^{s+1}) & \text{if } T \text{ is not enabled after } \alpha^{s+1} \\ 0 & \text{otherwise} \end{cases}$

with $\eta^X = \eta_{(X||\mathcal{E},q^s,a^{s+1})}$

Parallely, we have

$apply_{\mathcal{Z}'||\mathcal{E}}(\delta_{(\mu_z(q_X),q_\mathcal{E})}, \rho_1 T)(\mu_e(\alpha^{s+1})) = apply_{\mathcal{Z}'||\mathcal{E})}(apply_{\mathcal{Z}'||\mathcal{E}}(\delta_{(\mu_z(q_X),q_\mathcal{E})}, \rho_1), T)(\mu_e(\alpha^{s+1})) = p'_1(\mu_e(\alpha^{s+1})) + p'_2(\mu_e(\alpha^{s+1}))$

- $p'_1(\mu_e(\alpha^{s+1})) = \begin{cases} apply_{\mathcal{Z}'||\mathcal{E}}(\delta_{(\mu_z(q_X),q_\mathcal{E})}, \rho_1)(\mu_e(\alpha^s)) \cdot \eta^{\mathcal{Z}'}(\mu_z(q^{s+1})) & \text{if } (**) \\ 0 & \text{otherwise} \end{cases}$
- $p'_2(\mu_e(\alpha^{s+1})) = \begin{cases} apply_{\mathcal{Z}'||\mathcal{E}}(\delta_{(\mu_z(q_X),q_\mathcal{E})}, \rho_1)(\mu_e(\alpha^{s+1})) & \text{if } T \text{ is not enabled after } \mu_e(\alpha^{s+1}) \\ 0 & \text{otherwise} \end{cases}$

with $\eta^{\mathcal{Z}'} = \eta_{(\mathcal{Z}'||\mathcal{E},\mu_z(q^s),a^{s+1})}$ and $(**) : \mu_e(\alpha^{s+1}) = \mu_e(\alpha^s) a^{s+1} \mu_z(q^{s+1}), a^{s+1}$ triggered by T

We have : T enabled after $\alpha \iff T$ enabled after $\mu_e(\alpha)$. The leftward terms are equal by induction hypothesis, since $|\rho_1| = |\rho_2| - 1$. Using transition preservation we can obtain the equality for rightward terms. The conditions are matched in the same manner because of sigature homomorphism and we assume no creation from or of $\mathcal{A}$. Thus we can conclude that $p'_1(\mu_e(\alpha^{s+1})) = p_1(\alpha^{s+1})$ and $p'_2(\mu_e(\alpha^{s+1})) = p_2(\alpha^{s+1})$, which leads to $apply_{(X||\mathcal{E})}(\delta_{(q_X,q_\mathcal{E})}, \rho_1 T)(\alpha^{s+1}) = apply_{\mathcal{Z}'||\mathcal{E}}(\delta_{(\mu_z(q_X),q_\mathcal{E})}, \rho_1 T)(\mu_e(\alpha^{s+1}))$, which terminates the proof. ◀

6.4 Partial homomorphism

▶ **Definition 79** (configuration-equivalents states). Let X be a PCA. Let $q, q' \in states(X)$. We say that q and q' are *configuration-equivalents* iff $config(X)(q) = config(X)(q')$. The PCA X is said *configuration-equivalence-free* if for every configuration-equivalents pair (q, q'), $q = q'$.

▶ **Lemma 80** (injectivity of μ_z (modulo configuration-equivalence)). *Let $\mathcal{A} \in Autids$. Let X be a $\mathcal{A}$-conservative configuration-equivalence-free PCA, $Y = X \setminus \mathcal{A}$, .$Y'$ a $\mathcal{A}$-twin of Y. Then μ_z is an injection.*

Proof. Let $(q_Y, \tilde{q}_{\tilde{\mathcal{A}}^{sw}})$ be a states of $Y'||\tilde{\mathcal{A}}^{sw}$. Let q_X and q'_X s. t. $\mu_z(q_X) = \mu_z(q'_X) = (q_Y, \tilde{q}_{\tilde{\mathcal{A}}^{sw}})$. We will show that $q_X = q'_X$, by showing they are configuration-equivalent. At fist $config(X)(q_X) \setminus \mathcal{A} = config(X \setminus \mathcal{A})(q_Y) = config(X)(q'_X) \setminus \mathcal{A}$. Then $config(X)(q_X) = config(X)(q_X) \setminus \mathcal{A} = config(X)(q'_X)$ if $\mathcal{A} \notin aut(config(X)(q_X))$. So we treat the case where $\mathcal{A} \in aut(config(X)(q_X))$ and $aut(config(X)(q_X)(\mathcal{A}) = q_\mathcal{A}$. In this case $config(X)(q_X) = (config(X)(q_X) \setminus \mathcal{A}) \cup \{(\mathcal{A}, q_\mathcal{A})\} = config(X)(q'_X)$. Thus q_X, q'_X are configuration-equivalent, so if X is configuration-equivalence-free, then $q_X = q'_X$. Hence, μ_z is an injective function. ◀

▶ **Lemma 81** (injectivity of μ_e (modulo configuration-equivalence)). *Let $\mathcal{A} \in Autids$. Let X be a $\mathcal{A}$-conservative configuration-equivalence-free PCA, $Y = X \setminus \mathcal{A}$, .$Y'$ a $\mathcal{A}$-twin of Y. Then μ_e is an injection.*

Proof. Let $\alpha = q^0 a^1 ... q^s a^{s+1} q^{s+1} ...$. We have $\mu_e(\alpha) = \mu_z(q^0), a^1, ... \mu_z(q^s) a^{s+1} \mu_z(q^{s+1}) ...$ with μ_z an injection and identity function on actions an injection too. Thus μ_e is an injection. ◀

▶ **Theorem 82** (partial bijectivity). *Let $\mathcal{A} \in Autids$. Let X be a $\mathcal{A}$-conservative, configuration-equivalence-free PCA. Let $Y = X \setminus \mathcal{A}$. Let Y' be a $\mathcal{A}$-twin of Y. Let $\mathcal{Z}' = psioa(Y')||\mathcal{A}$.*

Let $\alpha = q^0, a^1, ..., a^k, q^k$ be an execution fragment of $\mathcal{Z}'$ where (a) $q^s_{\mathcal{A}} \neq q^{\phi}_{\mathcal{A}}$ for every $s \in [0, k^*]$ (b) $q^s_{\mathcal{A}} = q^{\phi}_{\mathcal{A}}$ for every $s \in [k^*+1, k]$ (c) for every $s \in [k^*+1, k-1]$, for every $\tilde{q}^s$, s. t. $\mu_z(\tilde{q}^s) = q^s$, $\mathcal{A} \notin created(X)(\tilde{q}^s)(a^{s+1})$. Then it exists a unique $\tilde{\alpha} \in frags(X)$, s. t. $\mu_e(\tilde{\alpha}) = \alpha$. If $Y' = Y$, it exists a unique $\tilde{\alpha} \in execs(X)$, s. t. $\mu_e(\tilde{\alpha}) = \alpha$.

Proof. We use partial surjectivity 2 for existence and partial injectivity for uniqueness. ◀

6.5 Composition and projection are commutative

▶ **Definition 83** ($\simeq$ relation between PCA states). Let $U = ((Q_U, \mathcal{F}_{Q_U}), \bar{q}_U, sig(U), D_U)$, $V = ((Q_V, \mathcal{F}_{Q_V}), \bar{q}_V, sig(V), D_V)$ be two PCA. Let $(q_U, q_V) \in Q_U \times Q_V$ s. t.

- $config(U)(q_U) = config(V)(q_V)$
- $hidden\text{-}actions(U)(q_U) = hidden\text{-}actions(V)(q_V)$
- $(\ sig(U)(q_U) = sig(V)(q_V)\ )$
- $\forall a \in sig(U)(q_U) \cup sig(V)(q_V)$, $created(U)(q_U)(a) = created(V)(q_V)(a)$

then we say that $q_U \simeq q_V$

The third point is implied by the two first points.

▶ **Lemma 84.** *Let $U = ((Q_U, \mathcal{F}_{Q_U}), \bar{q}_U, sig(U), D_U)$, $V = ((Q_V, \mathcal{F}_{Q_V}), \bar{q}_V, sig(V), D_V)$ be two PCA. Let $((q_U, q_V), (q'_U, q'_V)) \in (Q_U \times Q_V)^2$ s. t.*

- $config(U)(q_U) = config(V)(q_V)$
- $\forall a \in \widehat{sig}(U)(q_U) = \widehat{sig}(V)(q_V), created(U)(q_U)(a) = created(V)(q_V)(a)$
- $config(U)(q'_U) = config(V)(q'_V)$

then $\forall a \in sig(U)(q_U) = sig(V)(q_V)$, $\eta_{(U,q_U,a)}(q'_U) = \eta_{(V,q_V,a)}(q'_V)$.

Proof. We know that $config(U)(q_U) = config(V)(q_V) \triangleq C$ and $config(U)(q'_U) = config(V)(q'_V) \triangleq C'$.

Thus if it exists a reduced configuration distribution η' an action a and $\varphi \subset Autids$ s. t. $C \stackrel{a}{\Longrightarrow}_{\varphi} \eta'$, then both $(q_U, a, \eta_{(U,q_U,a)}) \in D_U$ with $\eta_{(U,q_U,a)}(q'_U) = \eta'(C')$ and $created(U)(q_U)(a) = \varphi$ and $(q_V a, \eta_{(V q_V, a)}) \in D_V$ with $\eta_{(V,q_V,a)}(q'_V) = \eta'(C')$, $created(V)(q_V)(a) = \varphi$ that is

$\eta_{(U,q_U,a)}(q'_U) = \eta_{(V,q_V,a)}(q'_v)$ and $created(U)(q_U)(a) = created(V)(q_V)(a)$.

Also if it exists $(q_U, a, \eta_{(U,q_U,a)}) \in D_U$, then it exists a reduced configuration distribution η', s. t. $C \stackrel{a}{\Longrightarrow}_{\varphi} \eta'$ with $\varphi = created(U)(q_U)(a) = created(V)(q_V)(a)$ and $\eta_{(U,q_U,a)}(q'_U) = \eta'(C')$. Thus it exists $(q_V a, \eta_{(V q_V, a)}) \in D_V$ with $\eta_{(V,q_V,a)}(q'_V) = \eta'(C') = \eta_{(U,q_U,a)}(q'_U)$.

Hence we obtain for every $((q_U, q_V), (q'_U, q'_V)) \in (Q_U \times Q_V)^2$, s. t.

- $config(U)(q_U) = config(V)(q_V)$
- $\forall a \in \widehat{sig}(U)(q_U) = \widehat{sig}(V)(q_V), created(U)(q_U)(a) = created(V)(q_V)(a)$
- $config(U)(q'_U) = config(V)(q'_V)$

then $\forall a \in sig(U)(q_U) = sig(V)(q_V)$, $\eta_{(U,q_U,a)}(q'_U) = \eta_{(V,q_V,a)}(q'_V)$. ◀

▶ **Definition 85** ($\simeq$ relation between PCA)**.** Let $U = ((Q_U, \mathcal{F}_{Q_U}), \bar{q}_U, sig(U), D_U)$, $V = ((Q_V, \mathcal{F}_{Q_V}), \bar{q}_V, sig(V), D_V)$ be two PCA where it exists an isomorphism $iso_{Q_{UV}} : Q_U \to Q_V$ ($iso_{Q_{VU}} = (iso_{Q_{UV}})^{-1} : Q_V \to Q_U$) s. t.

- $\bar{q}_V = iso_{Q_{UV}}(\bar{q}_U)$
- for every $(q_U, q_V) \in Q_U \times Q_V$, s. t. $q_V = iso_{Q_{UV}}(q_U)$, $q_U \simeq q_V$
- for every $((q_U, q_V), (q'_U, q'_V) \in (Q_U \times Q_V)^2$, s. t. $q_V = iso_{Q_{UV}}(q_U)$ and $q'_V = iso_{Q_{UV}}(q'_U)$, $\forall a \in sig(U)(q_U) \cup sig(V)(q_V)$, $\eta_{(U,q_U,a)}(q'_U) = \eta_{(V,q_V,a)}(q'_V)$.

then we say that $U \simeq V$

▶ **Lemma 86.** *Let $\mathcal{A} \in Autids$. Let X be a $\mathcal{A}$-conservative PCA. Let $\mathcal{E}$ be a PCA compatible with X.*

1. *$\mathcal{E}$ is compatible with Y'.*
2. *Let $q_{\mathcal{E}} \in states(\mathcal{E})$, $C_{\mathcal{E}} = config(\mathcal{E})(q_{\mathcal{E}})$. Let $q_X \in states(X)$, $C_X = config(X)(q_X)$. If it exists $q'_X \in states(X)$, s. t. $\mathcal{A} \in auts(config(X)(q'_X))$, then $(C_X \cup C_{\mathcal{E}}) \setminus \mathcal{A} = (C_X \setminus \mathcal{A}) \cup C_{\mathcal{E}}$.*
3. *Let $U = (X||\mathcal{E}) \setminus \mathcal{A}$ and $V = (X \setminus \mathcal{A})||\mathcal{E}$. Let $q_X \in states(X)$ and $q_{\mathcal{E}} \in states(\mathcal{E})$. Let $q_U = \mu_s^{\mathcal{A}}((q_X, q_{\mathcal{E}}))$ and $q_V = (\mu_s^{\mathcal{A}}(q_X), q_{\mathcal{E}})$. If it exists $q'_X \in states(X)$, s. t. $\mathcal{A} \in auts(config(X)(q'_X))$, then*
   - $q_U \simeq q_V$
   - $\bar{q}_U = \mu_s^{\mathcal{A}}((\bar{q}_X, \bar{q}_{\mathcal{E}}))$ *and* $\bar{q}_V = (\mu_s^{\mathcal{A}}(\bar{q}_X), \bar{q}_{\mathcal{E}})$

Proof. **1.** $\mathcal{E}$ is partially compatible with X for every state $(q_{\mathcal{E}}, q_X) \in states(\mathcal{E}) \times states(X)$, thus this is a fortiori true for every state $(q_{\mathcal{E}}, q_Y) \in states(\mathcal{E}) \times states(Y)$, since the configurations are the same excepting $\mathcal{A}$ is absent in $config(Y)(q_Y = \mu_s^{\mathcal{A}}(q_X))$. Thus $\mathcal{E}$ is partially compatible with Y' for every state $(q_{\mathcal{E}}, q_Y) \in states(\mathcal{E}) \times states(Y)$, which means $\mathcal{E}$ is compatible with Y'.

2. We note $\mathbf{A}_{\mathcal{E}} = auts(C_{\mathcal{E}})$, $\mathbf{S}_{\mathcal{E}} = map(C_{\mathcal{E}})$ and $\mathbf{A}_X = auts(C_X)$ and $\mathbf{S}_X = map(C_X)$. Since $\mathcal{E}$ is partially compatible with X for every state $(q_{\mathcal{E}}, q_X) \in states(\mathcal{E}) \times states(X)$, If it exists $q'_X \in states(X)$, s. t. $\mathcal{A} \in auts(config(X)(q'_X))$, then $\mathcal{A} \notin \mathbf{A}_{\mathcal{E}}$. Hence $(\mathbf{A}_X \cup \mathbf{A}_{\mathcal{E}}) \setminus \mathcal{A} = (\mathbf{A}_X \setminus \mathcal{A}) \cup \mathbf{A}_{\mathcal{E}}$, thus we obtain $(C_X \cup C_{\mathcal{E}}) \setminus \mathcal{A} = (C_X \setminus \mathcal{A}) \cup C_{\mathcal{E}}$.

3. Let $U = (X||\mathcal{E}) \setminus \mathcal{A}$ and $V = (X \setminus \mathcal{A})||\mathcal{E}$. Since $\mathcal{E}$ is partially compatible with X for every state $(q_{\mathcal{E}}, q_X) \in states(\mathcal{E}) \times states(X)$, If it exists $q'_X \in states(X)$, s. t. $\mathcal{A} \in auts(config(X)(q'_X))$, then $\mathcal{A} \notin \mathbf{A}_{\mathcal{E}}$.

- $config(U)(q_U) = (config(X)(q_X) \cup config(\mathcal{E})(q_{\mathcal{E}})) \setminus \mathcal{A} = (config(X)(q_X) \setminus \mathcal{A}) \cup config(\mathcal{E})(q_{\mathcal{E}})) = config(V)(q_V)$
- We note $q_{\mathcal{A}} = map(config(X)(q_X))(\mathcal{A})$ if $\mathcal{A} \in auts(config(X)(q_X))$, $q_{\mathcal{A}} = q_{\mathcal{A}}^{\phi}$ otherwise. We note $h_X = hidden\text{-}actions(X)(q_X)$ and $h_{\mathcal{E}} = hidden\text{-}actions(\mathcal{E})(q_{\mathcal{E}})$ and $h = (h_X \cup h_{\mathcal{E}}) \setminus \widehat{ext}(\mathcal{A})(q_{\mathcal{A}}))$ and $h' = (h_X \setminus \widehat{ext}(\mathcal{A})(q_{\mathcal{A}})) \cup h_{\mathcal{E}}$. Since X and $\mathcal{E}$ are partially-compatible in state $(q_X, q_{\mathcal{E}})$, we have both $config(X)(q_X)$ and $config(\mathcal{E})(q_{\mathcal{E}})$ compatible and $in(config(X)(q_X)) \cap h_{\mathcal{E}} = in(config(\mathcal{E})(q_{\mathcal{E}})) \cap h_X = \emptyset$. By compatibility, $out(config(X)(q_X)) \cap out(config(\mathcal{E})(q_{\mathcal{E}})) = \widehat{int}(config(X)(q_X)) \cap \widehat{sig}(config(\mathcal{E})(q_{\mathcal{E}})) = \widehat{sig}(config(X)(q_X)) \cap \widehat{int}(config(\mathcal{E})(q_{\mathcal{E}}))\emptyset$, which gives $\widehat{loc}(config(X)(q_X)) \cap h_{\mathcal{E}} = \widehat{loc}(config(\mathcal{E})(q_{\mathcal{E}})) \cap h_X = \emptyset$ and finally $\widehat{sig}(config(X)(q_X)) \cap h_{\mathcal{E}} = \widehat{sig}(config(\mathcal{E})(q_{\mathcal{E}})) \cap h_X = \emptyset$. This lead us to $h = h'$.
- We have $sig(U)(q_U) = hide(sig(config(U)(q_U), h)$ and $sig(V)(q_V) = hide(sig(config(V)(q_V), h')$ Since $config(U)(q_U) = config(V)(q_V)$ and $h = h'$, $sig(U)(q_U) = sig(V)(q_V)$.

- Since $\mathcal{E}$ is compatible with X, if it exists q'_X, s. t. $\mathcal{A} \in auts(config(X)(q'_X))$, $\mathcal{E}$ never creates $\mathcal{A}$. for every $a \in sig(q_U)$, $created(U)(q_U)(a) = (created(X)(q_X)(a) \cup created(\mathcal{E})(q_\mathcal{E})(a))\backslash\mathcal{A} = (created(X)(q_X)(a)\backslash\mathcal{A}) \cup created(\mathcal{E})(q_\mathcal{E})(a) = created(V)(q_V)(a)$
- By definition of projection and composition, we have $\bar{q}_U = \mu_s^\mathcal{A}((\bar{q}_X, \bar{q}_\mathcal{E}))$ and $\bar{q}_V = (\mu_s^\mathcal{A}(\bar{q}_X), \bar{q}_\mathcal{E})$.

◀

▶ **Theorem 87** (Projection and composition are commutative). *Let $\mathcal{A} \in Autids$. Let X be a PCA. where it exists $q'_X \in states(X)$, s. t. $\mathcal{A} \in auts(config(X)(q'_X))$. Let $\mathcal{E}$ be an environment for X. $(X||\mathcal{E}) \setminus \mathcal{A} \simeq (X \setminus \mathcal{A})||\mathcal{E}$.*

Proof. Let $U = (X||\mathcal{E})\backslash\mathcal{A} = ((Q_U, \mathcal{F}_{Q_U}), \bar{q}_U, sig(U), D_U)$ and $V = (X\backslash\mathcal{A})||\mathcal{E} = ((Q_V, \mathcal{F}_{Q_V}), \bar{q}_V, sig(V), D_V)$.

We have to show that there is an isomorphism iso between $U = (X||\mathcal{E})\backslash\mathcal{A} = ((Q_U, \mathcal{F}_{Q_U}), \bar{q}_U, sig(U), D_U)$ and $V = (X \backslash \mathcal{A})||\mathcal{E} = ((Q_V, \mathcal{F}_{Q_V}), \bar{q}_V, sig(V), D_V)$, s. t. it exists a bijection $iso_{Q_{UV}}$ between $(Q_U, \mathcal{F}_{Q_U})$ and $(Q_V, \mathcal{F}_{Q_V})$, where

- $\bar{q}_V = iso_{Q_{UV}}(\bar{q}_U)$
- for every $(q_U, q_V) \in Q_U \times Q_V$, s. t. $q_V = iso_{Q_{UV}}(q_U)$, $q_U \simeq q_V$
- for every $((q_U, q_V), (q'_U, q'_V) \in (Q_U \times Q_V)^2$, s. t. $q_V = iso_{Q_{UV}}(q_U)$ and $q'_V = iso_{Q_{UV}}(q'_U)$, $\forall a \in sig(U)(q_U) \cup sig(V)(q_V)$, $\eta_{(U,q_U,a)}(q'_U) = \eta_{(V,q_V,a)}(q'_V)$.

Let $q_X, q'_X \in states(X)$ and $q_\mathcal{E}, q'_\mathcal{E} \in states(\mathcal{E})$. Let $q_U = \mu_s^\mathcal{A}((q_X, q_\mathcal{E}))$, $q'_U = \mu_s^\mathcal{A}((q'_X, q'_\mathcal{E}))$, $q_V = (\mu_s^\mathcal{A}(q_X), q_\mathcal{E})$ and $q'_V = (\mu_s^\mathcal{A}(q'_X), q'_\mathcal{E})$.

At first we need to show there is a bijection between Q_U and Q_V. We note $iso_{Q_{UV}} : \mu_s((q_X, q_\mathcal{E})) \mapsto (\mu_s(q_X), q_\mathcal{E})$ and $iso_{Q_{VU}} : (\mu_s(q_X), q_\mathcal{E}) \mapsto \mu_s((q_X, q_\mathcal{E}))$ Thus mutual surjection is obvious, we need to show these are also injection. If $iso_{Q_{VU}}(q_V) = iso_{Q_{VU}}(q'_V)$, this implies $q_U = q'_U$, which implies $q_X \setminus \mathcal{A} = q'_X \setminus \mathcal{A}$ and so $q_V = q'_V$. For the same reasons If $iso_{Q_{UV}}(q_U) = iso_{Q_{UV}}(q'_U)$, this implies $q_V = q'_V$, which implies $q_X \setminus \mathcal{A} = q'_X \setminus \mathcal{A}$ and so $q_U = q'_U$.

Second, the choice of $iso_{Q_{UV}}$ and$iso_{Q_{VU}}$ gives the same criteria of the last lemma.

Third, we already know that for every $((q_U, q_V), (q'_U, q'_V) \in (Q_U \times Q_V)^2$, s. t. $q_V = iso_{Q_{UV}}(q_U)$ and $config(V)(q'_V) = config(U)(q'_U)$, $\forall a \in sig(U)(q_U) = sig(V)(q_V)$, $\eta_{(U,q_U,a)}(q'_U) = \eta_{(V,q_V,a)}(q'_V)$.

It rest to show that if $config(V)(q'_V) = config(U)(q'_U)$ and $q'_U \in supp(\eta_{(U,q_U,a)})$, then $q'_V = iso_{Q_{UV}}(q'_U)$. Because of constraint 3 of PCA, if $q''_U \in supp(\eta_{(U,q_U,a)})$ and $config(U)(q''_U) = config(U)(q''_U)$, then $q''_U = q'_U$ and in the same manner, if $q''_V \in supp(\eta_{(UVq_V,a)})$ and $config(V)(q''_V) = config(V)(q''_V)$, then $q''_V = q'_V$. Moreover $config(V)(iso_{Q_{VU}}(q'_U)) = config(U)(q'_U)$, so we necessarily have $q'_V = iso_{Q_{UV}}(q'_U)$, which means $q'_U \simeq q'_V$. Finally, we obtain for every $((q_U, q_V), (q'_U, q'_V) \in (Q_U \times Q_V)^2$, s. t. $q_V = iso_{Q_{UV}}(q_U)$ and $config(V)(q'_V) = config(U)(q'_U)$, $\forall a \in sig(U)(q_U) = sig(V)(q_V)$, $\eta_{(U,q_U,a)}(q'_U) = \eta_{(V,q_V,a)}(q'_V)$.

◀

There is an isomorphism between $(X||\mathcal{E}) \setminus \mathcal{A}$ and $(X \setminus \mathcal{A})||\mathcal{E}$ and the syntactic name of each state is arbitrary, which justify the choice of the sign $\simeq$.

7 Travel from one probabilistic space to another

In last section we have shown that the probability distribution of $X||\mathcal{E}$ was preserved by $\tilde{\mathcal{A}}^{sw}||(X \setminus \{\mathcal{A}\}||\mathcal{E})$, as long as $\mathcal{A}$ was not re-created by X.

In this section we take an interest in PCA $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ that differ only on the fact that $\mathcal{B}$ supplants $\mathcal{A}$ in $X_{\mathcal{B}}$. We define some equivalence classes on set of executions. These equivalence classes will allow us to transfer some reasoning on a situation on an execution α of $\mathcal{A}||psioa(X_{\mathcal{A}} \setminus \mathcal{A}||\mathcal{E})$ into an execution $\tilde{\alpha}$ of $X_{\mathcal{A}}||\mathcal{E}$.

7.1 Correspondence between two PCA

We formalise the idea that two configurations are the same excepting the fact that the process $\mathcal{B}$ supplants $\mathcal{A}$ but with the same external signature. The next definition comes from [1].

▶ **Definition 88** ($\lhd_{\mathcal{AB}}$-corresponding configurations)**.** (see figure 16) Let $\Phi \subseteq Autids$, and $\mathcal{A}, \mathcal{B}$ be PSIOA identifiers. Then we define $\Phi[\mathcal{B}/\mathcal{A}] = (\Phi \setminus \mathcal{A}) \cup \{\mathcal{B}\}$ if $\mathcal{A} \in \Phi$, and $\Phi[\mathcal{B}/\mathcal{A}] = \Phi$ if $\mathcal{A} \notin \Phi$. Let C, D be configurations. We define $C \lhd_{\mathcal{AB}} D$ iff (1) $auts(D) = auts(C)[\mathcal{B}/\mathcal{A}]$, (2) for every $\mathcal{A}' \notin auts(C) \setminus \{\mathcal{A}\} : map(D)(\mathcal{A}') = map(C)(\mathcal{A}')$, and (3) $ext(\mathcal{A})(s) = ext(\mathcal{B})(t)$ where $s = map(C)(\mathcal{A}), t = map(D)(\mathcal{B})$. That is, in $\lhd_{\mathcal{AB}}$-corresponding configurations, the SIOA other than $\mathcal{A}, \mathcal{B}$ must be the same, and must be in the same state. $\mathcal{A}$ and $\mathcal{B}$ must have the same external signature. In the sequel, when we write $\Psi = \Phi[\mathcal{B}/\mathcal{A}]$, we always assume that $\mathcal{B} \notin \Phi$ and $\mathcal{A} \notin \Psi$.

▶ Proposition 1. Let C, D be configurations such that $C \lhd_{AB} D$. Then $ext(C) = ext(D)$.

Proof. The proof is in [1], section 6, p. 38. ◀

▶ Remark. It is possible to have to configurations C, D s. t. $C \lhd_{\mathcal{AA}} D$. That would mean that C and D only differ on the state of $\mathcal{A}$ (s or t) that has even the same external signature in both cases $ext(\mathcal{A})(s) = ext(\mathcal{A})(t)$, while we would have $int(\mathcal{A})(s) \neq int(\mathcal{A})(t)$.

▶ **Lemma 89** (Same configuration)**.** *Let $\mathcal{A}, \mathcal{B} \in Autids$. Let $X_{\mathcal{A}}$, $X_{\mathcal{B}}$ be $\mathcal{A}$-fair and $\mathcal{B}$-fair PCA respectively, where $X_{\mathcal{A}}$ never contains $\mathcal{B}$ and $X_{\mathcal{B}}$ never contains $\mathcal{A}$. Let $Y_{\mathcal{A}} = X_{\mathcal{A}} \setminus \{\mathcal{A}\}$, $Y_{\mathcal{B}} = X_{\mathcal{B}} \setminus \{\mathcal{B}\}$. Let x_a, x_b s. t. $config(X_{\mathcal{A}})(x_a) \lhd_{AB} config(X_{\mathcal{B}})(x_b)$. Let $y_a = \mu_s(x_a)$, $y_b = \mu_s(x_b)$*

Then $config(Y_A)(y_a) = config(Y_B)(y_b)$.

Proof. By projection, we have $config(Y_{\mathcal{A}})(y_a) \lhd_{AB} config(Y_{\mathcal{B}})(y_b)$ with each configuration that does not contain $\mathcal{A}$ nor $\mathcal{B}$, thus for $config(Y_{\mathcal{A}})(y_a)$ and $config(Y_{\mathcal{B}})(y_b)$ contain the same set of automata ids (rule (1) of $\lhd_{AB}$) and map each automaton of this set to the same state (rule (2) of $\lhd_{AB}$). ◀

Now, we formalise the fact that two PCA create some PSIOA in the same manner, excepting for $\mathcal{B}$ that supplants $\mathcal{A}$.

▶ **Definition 90** (Creation corresponding configuration automata)**.** Let X, Y be configuration automata and $\mathcal{A}, \mathcal{B}$ be SIOA. We say that X, Y are creation-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$ iff

1. X never creates $\mathcal{B}$ and Y never creates $\mathcal{A}$.

2. Let $\beta \in traces^*(X) \cap traces^*(Y)$ a finite trace of both X and Y, and let $\alpha \in execs^*(X), \pi \in execs^*(Y)$ a finite execution of both X and Y be such that $trace_{\mathcal{A}}(\alpha) = trace_{\mathcal{A}}(\pi) = \beta$. Let $x = last(\alpha), y = last(\pi)$, i.e., x, y are the last states along α, π, respectively. Then $\forall a \in \widehat{sig}(X)(x) \cap \widehat{sig}(Y)(y) : created(Y)(y)(a) = created(X)(x)(a)[\mathcal{B}/\mathcal{A}]$.

▶ **Lemma 91** (Same creation). *Let $\mathcal{A}, \mathcal{B} \in Autids$. Let $X_{\mathcal{A}}$, $X_{\mathcal{B}}$ be $\mathcal{A}$-fair and $\mathcal{B}$-fair PCA respectively, where $X_{\mathcal{A}}$ never contains $\mathcal{B}$ and $X_{\mathcal{B}}$ never contains $\mathcal{A}$.*

Let $Y_{\mathcal{A}} = X_{\mathcal{A}} \setminus \mathcal{A}$, $Y_{\mathcal{B}} = X_{\mathcal{B}} \setminus \mathcal{B}$

Let $(x_a, x_b) \in states(X_{\mathcal{A}}) \times states(X_{\mathcal{B}})$ and $act \in sig(X_{\mathcal{A}})(x_a) \cap sig(X_{\mathcal{B}})(x_b)$ s. t. $created(X_{\mathcal{B}})(x_b)(act) = created(X_{\mathcal{A}})(x_a)(act)[\mathcal{B}/\mathcal{A}]$.

Let $y_a = \mu_s(x_a)$, $y_b = \mu_s(x_b)$

Then $created(Y_{\mathcal{B}})(x_b)(act) = created(Y_{\mathcal{A}})(x_a)(act)$

Proof. By definition of PCA projection, we have $created(Y_{\mathcal{B}})(x_b)(act) = (created(X_{\mathcal{B}})(x_b)(act)) \setminus \mathcal{B} = (created(X_{\mathcal{A}})(x_a)(act)[\mathcal{B}/\mathcal{A}]) \setminus \mathcal{B} = created(X_{\mathcal{A}})(x_a)(act) \setminus \mathcal{A} = created(Y_{\mathcal{A}})(x_a)(act)$. ◀

▶ **Definition 92** (Hiding corresponding configuration automata). Let X, Y be configuration automata and $\mathcal{A}, \mathcal{B}$ be PSIOA. We say that X, Y are hiding-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$ iff

1. X never creates $\mathcal{B}$ and Y never creates $\mathcal{A}$.
2. Let $\beta \in traces^*(X) \cap traces^*(Y)$, and let $\alpha \in execs^*(X), \pi \in execs^*(Y)$ be such that $trace_{\mathcal{A}}(\alpha) = trace_{\mathcal{A}}(\pi) = \beta$. Let $x = last(\alpha), y = last(\pi)$, i.e., x, y are the last states along α, π, respectively. Then $hidden\text{-}actions(Y)(y) = hidden\text{-}actions(X)(x)$.

▶ **Lemma 93** (Same hidden-actions). *Let $\mathcal{A}, \mathcal{B} \in Autids$. Let $X_{\mathcal{A}}$, $X_{\mathcal{B}}$ be $\mathcal{A}$-fair and $\mathcal{B}$-fair PCA respectively, where $X_{\mathcal{A}}$ never contains $\mathcal{B}$ and $X_{\mathcal{B}}$ never contains $\mathcal{A}$.*

Let $Y_{\mathcal{A}} = X_{\mathcal{A}} \setminus \mathcal{A}$, $Y_{\mathcal{B}} = X_{\mathcal{B}} \setminus \mathcal{B}$

Let x_a, x_b s. t. $hidden\text{-}actions(X_{\mathcal{B}})(x_b)(act) = hidden\text{-}actions(X_{\mathcal{A}})(x_a)$ and if $\mathcal{A} \in auts(config(X_{\mathcal{A}})(x_a)))$, then $ext(\mathcal{A})(map(\mathcal{A})(x_a)) = ext(\mathcal{B})(map(\mathcal{A})(x_b))$.

Let $y_a = \mu_s^{\mathcal{A}}(x_a)$, $y_b = \mu_s^{\mathcal{B}}(x_b)$

Then $hidden\text{-}actions(Y_{\mathcal{B}})(x_b) = hidden\text{-}actions(Y_{\mathcal{A}})(x_a)$

Proof. By definition of PCA projection, we have $hidden\text{-}actions(Y_{\mathcal{B}})(x_b)(act) = (hidden\text{-}actions(X_{\mathcal{B}})(x_b)(act)) \setminus out(\mathcal{B})(map(config(X_{\mathcal{B}})(x_b))) = (hidden\text{-}actions(X_{\mathcal{A}})(x_a)) \setminus out(\mathcal{B})(map(config(X_{\mathcal{B}})(x_b))) = hidden\text{-}actions(X_{\mathcal{A}})(x_a) \setminus out(\mathcal{A})(map(config(X_{\mathcal{A}})(x_a))) = hidden\text{-}actions(Y_{\mathcal{A}})(x_a)$. ◀

▶ **Definition 94.** Let Q_U, Q_V be sets of states and $Acts$ be a set of actions. Let α (resp. α') be an alternating sequence of states of Q_U (resp. Q_V) and actions of $Acts$ so that $\alpha = q^0, a^1, q^1 ... a^n, q^n$, $\alpha' = q'^0, a'^1, q'^1 ... a'^n, 'q^n$ and for every $i \in [0, n], q^i \simeq q'^i$ and for every $i \in [1, n], a^i = a'^i$, then we say that $\alpha \simeq \alpha'$.

▶ **Definition 95** (η^u *bij* η^v). Let U and V be PCA. Let $Q_U = states(U), Q_V = states(V)$ be sets of states and $Acts$ be a set of actions. Let $(\eta^u, \eta^v) \in Disc(Q_U) \times Disc(Q_V)$. We note η^u *bij* η^v if $supp(\eta^u)$ and $supp(\eta^v)$ are in bijection where for every $q'_u \in supp(\eta^u)$ it exists a unique $q'_v \in supp(\eta^v)$ s. t. $config(U)(q'_u) = config(V)(q'_v)$ and for every $(q'_u, q'_v) \in supp(\eta^u) \times supp(\eta^v)$ s. t. $config(U)(q'_u) = config(V)(q'_v)$, we have $\eta^u(q'_u) = \eta^v(q'_v)$.

▶ **Lemma 96.** *Let $\mathcal{A}, \mathcal{B} \in Autids$. Let $X_\mathcal{A}$, $X_\mathcal{B}$ be $\mathcal{A}$-fair and $\mathcal{B}$-fair PCA respectively, where $X_\mathcal{A}$ never contains $\mathcal{B}$ and $X_\mathcal{B}$ never contains $\mathcal{A}$.Let $Y_\mathcal{A} = X_\mathcal{A} \setminus \mathcal{A}$, $Y_\mathcal{B} = X_\mathcal{B} \setminus \mathcal{B}$.*

Let $(q_{Y_\mathcal{A}}, q_{Y_\mathcal{B}}) \in Q_{Y_\mathcal{A}} \times Q_{Y_\mathcal{B}}$ and an action a s. t.

- $config(Y_\mathcal{A})(q_{Y_\mathcal{A}}) = config(Y_\mathcal{B})(q_{Y_\mathcal{B}})$
- $act \in sig(config(Y_\mathcal{A})(q_{Y_\mathcal{A}}) = sig(config(Y_\mathcal{B})(q_{Y_\mathcal{B}})$
- $created(Y_\mathcal{A})(act)(q_{Y_\mathcal{A}}) = created(Y_\mathcal{B})(q_{Y_\mathcal{B}})(act)$

, then $\eta_{(Y_\mathcal{A}, q_{Y_\mathcal{A}}, act)}$ bij $\eta_{(Y_\mathcal{B}, q_{Y_\mathcal{B}}, act)}$

Proof. We note $C_a \triangleq config(Y_\mathcal{A})(q_{Y_\mathcal{A}})$ and $C_b \triangleq config(Y_\mathcal{B})(q_{Y_\mathcal{B}})$. Since $q_{Y_\mathcal{A}} \simeq q_{Y_\mathcal{B}}$, $C \triangleq C_a = C_b$, and hence $sig \triangleq sig(C_a) = sig(C_b)$ and for every. Since $\varphi \triangleq created(Y_\mathcal{A})(q_{Y_\mathcal{A}})(act) = created(Y_\mathcal{B})(q_{Y_\mathcal{B}})(act)$. Thus there is a unique η_p s. t. $C \overset{a}{\rightharpoonup} \eta_p$ and a unique η_r generated by φ and η_p s. t. $C \overset{a}{\Longrightarrow}_\varphi \eta_p$. Because of constraint 3, it exists $(q_{Y_\mathcal{A}}, act, \eta^a) \in D_{Y_\mathcal{A}}$ and $(q_{Y_\mathcal{A}}, act, \eta^b) \in D_{Y_\mathcal{B}}$ s. t. for every for every $C' \in supp(\eta_r)$, it exists a unique state $q'_{Y_\mathcal{A}} \in supp(\eta^a)$ (resp. $q'_{Y_\mathcal{B}} \in supp(\eta^b)$) of $Y_\mathcal{A}$ (resp. $Y_\mathcal{B}$) s. t. $config(Y_\mathcal{A})(q'_{Y_\mathcal{A}}) = C'$ (resp. $config(Y_\mathcal{A})(q'_{Y_\mathcal{A}}) = C'$) and $\eta^a(q'_{Y_\mathcal{A}}) = \eta_r(C')$ (resp. $\eta^b(q'_{Y_\lfloor}) = \eta_r(C')$. Thus $supp(\eta^a)$ and $supp(\eta^b)$ are in bijection where for every $q'_{Y_\mathcal{A}} \in supp(\eta^a)$ it exists a unique $q'_{Y_\mathcal{B}} \in supp(\eta^b)$ s. t. $config(Y_\mathcal{A})(q'_{Y_\mathcal{A}}) = config(Y_\mathcal{B})(q'_{Y_\mathcal{B}})$ and for every $(q'_{Y_\mathcal{A}}, q'_{Y_\mathcal{B}}) \in supp(\eta^a) \times supp(\eta^b)$ s. t. $config(Y_\mathcal{A})(q'_{Y_\mathcal{A}}) = config(Y_\mathcal{B})(q'_{Y_\mathcal{B}})$, we have $\eta^a(q'_{Y_\mathcal{A}}) = \eta^b(q'_{Y_\mathcal{B}})$. Thus η^a *bij* η^b ◀

▶ **Definition 97** ($\eta^u \simeq \eta^v$)**.** Let U and V be PCA. Let $Q_U = states(U), Q_V = states(V)$ be sets of states and $Acts$ be a set of actions. Let $(\eta^u, \eta^v) \in Disc(Q_U) \times Disc(Q_V)$. We note $\eta^u \simeq \eta^v$ if $supp(\eta^u)$ and $supp(\eta^v)$ are in bijection where for every $q'_u \in supp(\eta^u)$ it exists a unique $q'_v \in supp(\eta^v)$ s. t. $q'_u \simeq q'_v$ and for every $(q'_u, q'_v) \in supp(\eta^u) \times supp(\eta^v)$ s. t. $q'_u \simeq q'_v$, we have $\eta^u(q'_u) = \eta^v(q'_v)$.

▶ **Definition 98** (corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$)**.** Let $\mathcal{A}, \mathcal{B} \in Autids$, $X_\mathcal{A}$ and $X_\mathcal{B}$ be PCA we say that $X_\mathcal{A}$ and $X_\mathcal{B}$ are corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$, if they verify:

- $config(X_\mathcal{A})(\bar{q}_{X_\mathcal{A}}) \lhd_{AB} config(X_\mathcal{B})(\bar{q}_{X_\mathcal{B}})$.
- $X_\mathcal{A}, X_\mathcal{B}$ are creation-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$
- $X_\mathcal{A}, X_\mathcal{B}$ are hiding-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$
- $X_\mathcal{A}$ (resp. $X_\mathcal{B}$) is a $\mathcal{A}$-conservative (resp. $\mathcal{B}$-conservative) PCA.
- (No creation from $\mathcal{A}$ and $\mathcal{B}$)
  - $\forall q_{X_\mathcal{A}}, \in states(X_\mathcal{A}), \forall act$ verifying $act \notin sig(config(X_\mathcal{A})(q_{X_\mathcal{A}})\setminus\{\mathcal{A}\}) \wedge act \in sig(config(X_\mathcal{A})(q_{X_\mathcal{A}}))$, $created(X_\mathcal{A})(q_{X_\mathcal{A}})(act) = \emptyset$ and similarly
  - $\forall q_{X_\mathcal{B}}, \in states(X_\mathcal{B}), \forall act'$ verifying $act' \notin sig(config(X_\mathcal{B})(q_{X_\mathcal{B}})\setminus\{\mathcal{B}\}) \wedge act' \in sig(config(X_\mathcal{B})(q_{X_\mathcal{B}}))$, $created(X_\mathcal{B})(q_{X_\mathcal{B}})(act') = \emptyset$

▶ **Lemma 99.** *Let $\mathcal{A}, \mathcal{B} \in Autids$. Let $X_\mathcal{A}$, $X_\mathcal{B}$ be corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$. Let $Y_\mathcal{A} = X_\mathcal{A} \setminus \mathcal{A}$, $Y_\mathcal{B} = X_\mathcal{B} \setminus \mathcal{B}$.*

Let $(\alpha^a, \alpha^b) \in execs(Y_\mathcal{A}) \times execs(Y_\mathcal{B})$, s. t. $\alpha^a \simeq \alpha^b$, where $lstate(\alpha^a) = q_{Y_\mathcal{A}}$ and $lstate(\alpha^b) = q_{Y_\mathcal{B}}$ and $act \in sig(config(Y_\mathcal{A})(q_{Y_\mathcal{A}}) = sig(config(Y_\mathcal{B})(q_{Y_\mathcal{B}})$.

then $\eta_{(Y_\mathcal{A}, q_{Y_\mathcal{A}}, act)} \simeq \eta_{(Y_\mathcal{B}, q_{Y_\mathcal{B}}, act)})$

Proof. We already have $\eta_{(Y_\mathcal{A}, q_{Y_\mathcal{A}}, act)}$ *bij* $\simeq \eta_{(Y_\mathcal{B}, q_{Y_\mathcal{B}}, act)})$, by the previous lemma. Let $(q'_{Y_\mathcal{A}}, q'_{Y_\mathcal{B}}) \in supp(\eta_{(Y_\mathcal{A}, q_{Y_\mathcal{A}}, act)}) \times supp(\eta_{(Y_\mathcal{B}, q_{Y_\mathcal{B}}, act)}))$, s. t. $config(Y_\mathcal{A})(q_{Y_\mathcal{A}}) = config(Y_\mathcal{B})(q_{Y_\mathcal{B}})$.

- $hidden\text{-}actions(Y_\mathcal{B})(q'_{Y_\mathcal{B}}) = hidden\text{-}actions(Y_\mathcal{A})(q'_{Y_\mathcal{A}})$, because of hiding-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$.

- $created(Y_\mathcal{B})(q'_{Y_\mathcal{B}}) = created(Y_\mathcal{A})(q'_{Y_\mathcal{A}})$, because of creation-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$.

This ends the proof.

◀

▶ **Lemma 100.** *Let $\mathcal{A}, \mathcal{B} \in Autids$. Let $X_\mathcal{A}$, $X_\mathcal{B}$ be PCA corresponding w. r. t. $\mathcal{A}, \mathcal{B}$. Let $Y_\mathcal{A} = X_\mathcal{A} \setminus \mathcal{A}$, $Y_\mathcal{B} = X_\mathcal{B} \setminus \mathcal{B}$. Then $\tilde{\mathcal{B}}^{sw}$ and $Y_\mathcal{A}$ are partially compatible. (Symetrically, $\tilde{\mathcal{A}}^{sw}$ and $Y_\mathcal{B}$ are partially compatible.)*

Proof. By induction. Basis At first $\tilde{\mathcal{B}}^{sw}$ and $Y_\mathcal{B}$ are 0-partially-compatible. Moreover, we have $config(Y_\mathcal{B})(\bar{q}_{Y_\mathcal{B}}) = config(Y_\mathcal{A})(\bar{q}_{Y_\mathcal{A}})$, thus $\tilde{\mathcal{B}}^{sw}$ and $Y_\mathcal{A}$ are 0-partially-compatible. Induction: Now we want to show that every pseudo-execution of $(\tilde{\mathcal{B}}^{sw}, Y_\mathcal{A})$ ends on a partially-compatible state. Let $\alpha^a = q^{a,0}act^0, ..., act^\ell q^{a,\ell}$ be a pseudo-execution of $(\tilde{\mathcal{B}}^{sw}, Y_\mathcal{A})$. We will show by induction that P^ℓ : it exists a unique execution $\alpha^b = q^{b,0}act^0, ..., act^\ell q^{b,\ell}$ of $Y_\mathcal{B}||\tilde{\mathcal{B}}^{sw}$, s. t.

- $\alpha^b \simeq \alpha^a$ and
- $\forall s \in [1, \ell]$, $\eta_{((Y'_\mathcal{A}, \tilde{\mathcal{B}}^{sw}), q^{(a,s-1)}, act^s)} \simeq \eta_{((Y'_\mathcal{B}, \tilde{\mathcal{B}}^{sw}), q^{(b,s-1)}, act^s)}$.

We assume $P^{\ell-1}$ to be true and we show it imples P^ℓ We have $\eta_{((Y'_\mathcal{A}, \tilde{\mathcal{B}}^{sw}), q^{(a,s\ell-1)}, act^s)} \simeq \eta_{((Y'_\mathcal{B}, \tilde{\mathcal{B}}^{sw}), q^{(b,\ell-1)}, act^s)}$ from the last lemma. Because of this, if $q^{b,\ell} \in supp(\eta_{((Y'_\mathcal{B}, \tilde{\mathcal{B}}^{sw}), q^{(b,\ell-1)}, act^s)})$, then it exists $q^{a,\ell} \in supp(\eta_{((Y'_\mathcal{A}, \tilde{\mathcal{B}}^{sw}), q^{(a,\ell-1)}, act^s)})$ s. t. $q^{(a,\ell)} \simeq q^{(b,\ell)}$, that shows P^ℓ. Hence P^ℓ is true for every $\ell \in \mathbb{N}$. Furthermore, $q^{b,\ell}$ is a state of $Y_\mathcal{B}||\tilde{\mathcal{B}}^{sw}$. Thus $(\tilde{\mathcal{B}}^{sw}, Y_\mathcal{A})$ are partially-compatible at state $q^{(a,\ell)}$. We conclude that that every pseudo-execution of $(\tilde{\mathcal{B}}^{sw}, Y_\mathcal{A})$ ends on a partially-compatible state, which ends the proof.

◀

▶ **Definition 101.** Let $\mathcal{A}, \mathcal{B} \in Autids$. Let $X_\mathcal{A}$, $X_\mathcal{B}$ be PCA corresponding w. r. t. $\mathcal{A}, \mathcal{B}$. Let $Y_\mathcal{A} = X_\mathcal{A} \setminus \mathcal{A}$, $Y_\mathcal{B} = X_\mathcal{B} \setminus \mathcal{B}$. Let $Y'_\mathcal{A}$ be a $\mathcal{A}$-twin of $Y_\mathcal{A}$ and $Y'_\mathcal{B}$ be a $\mathcal{B}$-twin of $Y_\mathcal{B}$. We say that $Y'_\mathcal{A}$ and $Y'_\mathcal{B}$ are $\mathcal{AB}$-co-twin of $Y_\mathcal{A}$ and $Y_\mathcal{B}$ if it exists $\alpha^a \in execs(Y_\mathcal{A})$ and $\alpha^b \in execs(Y_\mathcal{B})$, s. t. (1) $lstate(\alpha^a) = \bar{q}_{Y'_\mathcal{A}}$ (2) $lstate(\alpha^b) = \bar{q}_{Y'_\mathcal{B}}$ and (3) $\alpha^a \simeq \alpha^b$.

▶ **Lemma 102.** *Let $\mathcal{A}, \mathcal{B} \in Autids$. Let $X_\mathcal{A}$, $X_\mathcal{B}$ be PCA corresponding w. r. t. $\mathcal{A}, \mathcal{B}$. Let $Y_\mathcal{A} = X_\mathcal{A} \setminus \mathcal{A}$, $Y_\mathcal{B} = X_\mathcal{B} \setminus \mathcal{B}$. Let $Y'_\mathcal{A}$ and $Y'_\mathcal{B}$ be $\mathcal{AB}$-co-twin of $Y_\mathcal{A}$ and $Y_\mathcal{B}$.*

Then $\tilde{\mathcal{B}}^{sw}$ and $Y'_\mathcal{A}$ are partially compatible. (Symetrically, $\tilde{\mathcal{A}}^{sw}$ and $Y'_\mathcal{B}$ are partially compatible.)

Proof. Immediate from previous lemma, since $\bar{q}_{Y'_\mathcal{A}}$ is reachable by $Y_\mathcal{A}$. ◀

▶ **Theorem 103** ($(\tilde{\mathcal{B}}^{sw}||Y'_\mathcal{A}) \simeq (\tilde{\mathcal{B}}^{sw}||Y'_\mathcal{B})$)**.** *Let $\mathcal{A}, \mathcal{B} \in Autids$. Let $X_\mathcal{A}$, $X_\mathcal{B}$ be PCA corresponding w. r. t. $\mathcal{A}, \mathcal{B}$. Let $Y_\mathcal{A} = X_\mathcal{A} \setminus \mathcal{A}$, $Y_\mathcal{B} = X_\mathcal{B} \setminus \mathcal{B}$. Let $Y'_\mathcal{A}$ and $Y'_\mathcal{B}$ be $\mathcal{AB}$-co-twin of $Y_\mathcal{A}$ and $Y_\mathcal{B}$*

$\tilde{\mathcal{B}}^{sw}$ and $Y'_\mathcal{A}$ are partially compatible. (Symetrically, $\tilde{\mathcal{A}}^{sw}$ and $Y'_\mathcal{B}$ are partially compatible.)

and for every $(\alpha^a, \alpha^b) \in frags(\tilde{\mathcal{B}}^{sw}||Y'_\mathcal{A}) \times frags(\tilde{\mathcal{B}}^{sw}||Y'_\mathcal{B})$ s. t. $\alpha^a \simeq \alpha^b$, for every $(\mu^a, \mu^b) \in Disc(frags(\tilde{\mathcal{B}}^{sw}||Y'_\mathcal{A})) \times Disc(frags(\tilde{\mathcal{B}}^{sw}||Y'_\mathcal{B}))$ s. t. $\mu^a \simeq \mu^b$ and for every sequence of tasks ρ, $apply_{(\tilde{\mathcal{B}}^{sw}||Y'_\mathcal{B})}(\mu^a, \rho)(\alpha^b) = apply_{(\tilde{\mathcal{B}}^{sw}||Y'_\mathcal{A})}(\mu^b, \rho)(\alpha^a)$.

Proof. We reuse the property P^ℓ that we proved to be true for every $\ell \in \mathbb{N}$.

P^ℓ: For every $\alpha^a = q^{a,0}act^0, ..., act^\ell q^{a,\ell}$ being an execution of $\tilde{\mathcal{B}}^{sw}||Y_\mathcal{A})$. it exists a unique execution $\alpha^b = q^{b,0}act^0, ..., act^\ell q^{b,\ell}$ of $Y'_\mathcal{B}||\tilde{\mathcal{B}}^{sw}$ s. t.

- $\alpha^b \simeq \alpha^a$ and
- $\forall s \in [1, \ell]$, $\eta_{((Y'_{\mathcal{A}}, \tilde{\mathcal{B}}^{sw}), q^{(a,s-1)}, act^s)} \simeq \eta_{((Y'_{\mathcal{B}}, \tilde{\mathcal{B}}^{sw}), q^{(b,s-1)}, act^s)}$.

Furthermore, the equality of probability of corresponding states gives the equality of corresponding executions for the same schedule.

We show it by induction on the size of ρ exactly as we did in the theorem of preservation of probabilistic distribution without creation.

Basis: $apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{B}})}(\mu^b, \lambda)(\alpha^b) = \mu^b(\alpha^b)$, while $apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{A}})}(\mu^a, \lambda)(\alpha^a) = \mu^a(\alpha^a) = \mu^b(\alpha^b)$.

Let assume this is true for ρ_1. We consider $\alpha^{a,s+1} = \alpha^{a,s} \frown a^{s+1} q^{a,s+1}$, $\alpha^{b,s+1} = \alpha^{b,s} \frown a^{s+1} q^{b,s+1}$ and $\rho_2 = \rho_1 T$.

$apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{B}})}(\mu^b, \rho_1 T)(\alpha^{b,s+1}) = apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{B}})}(apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{B}})}(\mu^b, \rho_1), T)(\alpha^{b,s+1}) = p_1(\alpha^{b,s+1}) + p_2(\alpha^{b,s+1})$

- $p_1(\alpha^{b,s+1}) = \begin{cases} apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{B}})}(\mu^b, \rho_1)(\alpha^{b,s}) \cdot \eta^b(q^{b,s+1}) & \text{if } \alpha^{b,s+1} = \alpha^{b,s} \frown a^{s+1} q^{b,s+1}, a^{s+1} \text{ triggered by } T \\ 0 & \text{otherwise} \end{cases}$
- $p_2(\alpha^{b,s+1}) = \begin{cases} apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{B}})}(\mu^b, \rho_1)(\alpha^{b,s+1}) & \text{if } T \text{ is not enabled after } \alpha^{b,s+1} \\ 0 & \text{otherwise} \end{cases}$

with $\eta^b = \eta_{((\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{B}}), q^{b,s}, a^{s+1})}$

Parallely, we have

$apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{A}})}(\mu^a, \rho_1 T)(\alpha^{a,s+1}) = apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{A}})}(apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{A}})}(\mu^a, \rho_1), T)(\alpha^{a,s+1}) = p'_1(\alpha^{a,s+1}) + p'_2(\alpha^{a,s+1})$

- $p'_1(\alpha^{a,s+1}) = \begin{cases} apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{A}})}(\mu^a, \rho_1)(\alpha^{a,s}) \cdot \eta^a(q^{a,s+1}) & \text{if } \alpha^{a,s+1} = \alpha^{a,s} \frown a^{s+1} q^{a,s+1}, a^{s+1} \text{ triggered by } T \\ 0 & \text{otherwise} \end{cases}$
- $p'_2(\alpha^{a,s+1}) = \begin{cases} apply_{(\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{A}})}(\mu^a, \rho_1)(\alpha^{a,s+1}) & \text{if } T \text{ is not enabled after } \alpha^{a,s+1} \\ 0 & \text{otherwise} \end{cases}$

with $\eta^a = \eta_{((\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{A}}), q^{a,s}, a^{s+1})}$

We have : T enabled after $\alpha^a \iff T$ enabled after α^b, since $constitution((\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{A}}))(lstate(\alpha^a)) = constitution((\tilde{\mathcal{B}}^{sw}||Y'_{\mathcal{B}}))(lstate(\alpha^b))$ The leftward terms are equal by induction hypothesis, since $|\rho_1| = |\rho_2| - 1$. Since the probabilistic distributions are in bijection we can obtain the equality for rightward terms. The conditions are matched in the same manner because of signature equality. Thus we can conclude that $p'_1(\alpha^{a,s+1}) = p_1(\alpha^{b,s+1})$ and $p'_2(\alpha^{a,s+1})) = p_2(\alpha^{b,s+1})$, which leads to the result.

◀

7.2 Handle destruction

▶ **Definition 104** (Ending on creation). Let $K_{\mathcal{A}}$ be a PCA. We say that $\alpha \in frags(K_{\mathcal{A}})$ *ends on $\mathcal{A}$ creation* iff $\alpha = (\alpha' a q)$ and $\mathcal{A} \in map(config(K_{\mathcal{A}})(q))$ and $\mathcal{A} \notin map(config(K_{\mathcal{A}})(lstate(\alpha')))$.

▶ **Definition 105** (Ending on destruction). Let $K_{\mathcal{A}}$ be a PCA. We say that $\alpha \in frags(K_{\mathcal{A}})$ *ends on $\mathcal{A}$ destruction* iff $\alpha = (\alpha' a q)$ and $\mathcal{A} \notin map(config(K_{\mathcal{A}})(q))$ and $\mathcal{A} \in map(config(K_{\mathcal{A}})(lstate(\alpha')))$.

▶ **Definition 106** (No creation). Let $K_{\mathcal{A}}$ be a PCA. We say that $\alpha \in frags(K_{\mathcal{A}})$ *does not create $\mathcal{A}$* if no prefix α' of α ends on $\mathcal{A}$ creation.

▶ **Definition 107** (No destruction)**.** Let $K_{\mathcal{A}}$ be a PCA. We say that $(\alpha) \in frags(K_{\mathcal{A}})$ *does not destroy* $\mathcal{A}$ if no prefix α' of α ends on $\mathcal{A}$ destruction.

▶ **Definition 108** (Permanence)**.** Let $\mathcal{A}$ be a PSIOA. Let $K_{\mathcal{A}}$ be a PCA. Let $\alpha \in frags(K_{\mathcal{A}})$. We say that $\mathcal{A}$ is *permanently present* in α if $\mathcal{A} \in map(config(K_{\mathcal{A}})(fstate(\alpha)))$ and α does not destroy $\mathcal{A}$. We say that $\mathcal{A}$ is *permanently absent* in α if $\mathcal{A} \notin map(config(K_{\mathcal{A}})(fstate(\alpha)))$ and α does not create $\mathcal{A}$. We say that α is $\mathcal{A}$*-permanent* if $\mathcal{A}$ is either permanently present or permanently absent in α.

Let $\mathcal{B}$ be another PSIOA partially-compatible with $\mathcal{A}$ and $\alpha \in frags(\mathcal{A}||\mathcal{B})$. We say that $\mathcal{A}$ is *permanently on* in α if $\forall j \in [0, |\alpha|], \widehat{sig}(\mathcal{A})(q^j_{\mathcal{A}}) \neq \emptyset$ and *permanently off* in α if $\forall j \in [0, |\alpha|], \widehat{sig}(\mathcal{A})(q^j_{\mathcal{A}}) = \emptyset$.

▶ **Definition 109** (Segment)**.** Let $\mathcal{A}$ be a PSIOA. Let $K_{\mathcal{A}}$ be a PCA. Let $\alpha \in frags(K_{\mathcal{A}})$. We say that α' is a $\mathcal{A}$*-filled-segment* if $\alpha' = \alpha^\frown aq$, $\mathcal{A}$ is permanently present in α but not in α'. and $map(config(K_{\mathcal{A}})(fstate(\alpha)))(\mathcal{A}) = \bar{q}_{\mathcal{A}}$. We say that α' is a $\mathcal{A}$*-unfilled-segment* if $\alpha' = \alpha^\frown aq$, $\mathcal{A}$ is permanently absent in α but not in α'. We say α' is a $\mathcal{A}$*-segment* if it is either $\mathcal{A}$-filled-segment or a $\mathcal{A}$-unfilled-segment.

Let $\mathcal{B}$ be another PSIOA partially-compatible with $\mathcal{A}$ and $\alpha' \in frags(\mathcal{A}||\mathcal{B})$. We say that $\mathcal{A}$ is turned off in $\alpha' = \alpha^\frown aq$, if $\mathcal{A}$ is *permanently on* in α but not in α'. We say that α' is a $\mathcal{A}$-segment if it is turned off in α' and $fstate(\alpha') \upharpoonright \mathcal{A} = \bar{q}_{\mathcal{A}}$.

▶ **Definition 110.** Let $\mathcal{A}$ be a PSIOA. Let $\tilde{\mathcal{A}}^{sw}$ its simpleton wrapper. Let $\mathcal{E}$ be an environment of $\tilde{\mathcal{A}}^{sw}$. Let $\tilde{\alpha} = \tilde{q}^0 a^1 \tilde{q}^1 ...$ be an execution of $\tilde{\mathcal{A}}^{sw}||\mathcal{E}$ with $PSIOA(\tilde{\mathcal{A}}^{sw}) = ren_{sw}(\mathcal{A})$ where each state $\tilde{q}^j = (\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}, \tilde{q}^j_{\mathcal{E}})$. We note $\gamma^{\mathcal{A}}_e(\alpha) = q^0 a^1 q^1 ...$ the execution of $\mathcal{A}||PSIOA(\mathcal{E})$ s. t. for every j, $q^j = (q^j_{\mathcal{A}}, q^j_{\mathcal{E}'}) = (ren^{-1}_{sw}(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}), \tilde{q}^j_{\mathcal{E}})$.

▶ **Lemma 111.** *Let $\mathcal{A}$ be a PSIOA. Let $\tilde{\mathcal{A}}^{sw}$ its simpleton wrapper. Let $\mathcal{E}$ be an environment of $\tilde{\mathcal{A}}^{sw}$. Let $\tilde{\alpha}$ be an execution of $\tilde{\mathcal{A}}^{sw}||\mathcal{E}$ with $PSIOA(\tilde{\mathcal{A}}^{sw}) = ren_{sw}(\mathcal{A})$, let $\alpha = \gamma^{\mathcal{A}}_e(\alpha)$ the corresponding execution of $\mathcal{A}||PSIOA(\mathcal{E})$.*

Then

1. *$\mathcal{A}$ is permanently on in $\alpha \Longleftrightarrow \mathcal{A}$ is permanently present in $\tilde{\alpha}$.*
2. *$\mathcal{A}$ is permanently off in $\alpha \Longleftrightarrow \mathcal{A}$ is permanently absent in $\tilde{\alpha}$.*
3. *α is a $\mathcal{A}$-segment $\Longleftrightarrow \tilde{\alpha}$ is a $\mathcal{A}$-filled-segment.*
4. *$\alpha = \alpha^1 {}^\frown \alpha^2$ where α^1 is a $\mathcal{A}$-segment and $\mathcal{A}$ is permanently off in $\alpha^2 \Longleftrightarrow \tilde{\alpha} = \tilde{\alpha}^1 {}^\frown \tilde{\alpha}^2$ where $\tilde{\alpha}^1$ is a $\mathcal{A}$-filled-segment and $\mathcal{A}$ is permanently absent in $\tilde{\alpha}^2$ and $\gamma^{\mathcal{A}}_e(\tilde{\alpha}^i) = \alpha^i$ for $i \in \{1, 2\}$.*

Proof. **1.**
- $\mathcal{A}$ is permanently present in $\tilde{\alpha} \Longrightarrow$ for every $j \in [0, n]$, $\mathcal{A} \in aut(\tilde{\mathcal{A}}^{sw})(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}))$. Since each state of $\tilde{\mathcal{A}}^{sw}$ is mapped to a reduced configuration, for every $j \in [0, n]$ $map(config(\tilde{\mathcal{A}}^{sw})(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}))(\mathcal{A}) \neq q^{\phi}_{\mathcal{A}}$. Thus, for every $j \in [0, n]$, if $(q^j_{\mathcal{A}}, q^j_{\mathcal{E}}) = \gamma^{\mathcal{A}}_e(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}, \tilde{q}^j_{\mathcal{E}})$, then $q^j_{\mathcal{A}} \neq q^{\phi}_{\mathcal{A}}$, which means $\mathcal{A}$ is permanently on in α. We obtained $\mathcal{A}$ is permanently present in $\tilde{\alpha} \Longrightarrow \mathcal{A}$ permanently on in α.
- $\mathcal{A}$ is not permanently present in $\tilde{\alpha} \Longrightarrow$ it exists $j \in [0, n]$, $\mathcal{A} \notin aut(config(\tilde{\mathcal{A}}^{sw})(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}))$. If $(q^j_{\mathcal{A}}, q^j_{\mathcal{E}}) = \gamma^{\mathcal{A}}_e(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}, \tilde{q}^j_{\mathcal{E}})$, with $\mathcal{A} \notin aut(config(\tilde{\mathcal{A}}^{sw})(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}))$, then $q^j_{\mathcal{A}} = q^{\phi}_{\mathcal{A}}$, which means $\mathcal{A}$ is not permanently on in α. By contraposition, $\mathcal{A}$ is permanently on in α $\Longrightarrow \mathcal{A}$ is permanently present in $\tilde{\alpha}$.

We obtained $\mathcal{A}$ is permanently on in $\alpha \Longleftrightarrow \mathcal{A}$ is permanently present in $\tilde{\alpha}$.

2.
- $\mathcal{A}$ is permanently absent in $\tilde{\alpha} \Longrightarrow$ for every $j \in [0, n]$, $\mathcal{A} \notin aut(config(X)(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}))$. Thus for every $j \in [0, n]$ where $(q^j_{\mathcal{A}}, q^j_{\mathcal{E}}) = \mu_z(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}, \tilde{q}^j_{\mathcal{E}})$, $q^j_{\mathcal{A}} = q^{\phi}_{\mathcal{A}}$, which means $\mathcal{A}$ is

permanently off in α. We obtained $\mathcal{A}$ is permanently absent in $\tilde{\alpha} \implies \mathcal{A}$ permanently off in α.

- $\mathcal{A}$ is not permanently absent in $\tilde{\alpha} \implies$ it exists $j \in [0,n]$, $\mathcal{A} \in aut(config(\tilde{\mathcal{A}}^{sw})(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}))$. Since each state of $\tilde{\mathcal{A}}^{sw}$ is mapped to a reduced configuration, $map(config(\tilde{\mathcal{A}}^{sw})(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}))(\mathcal{A}) \neq q^{\phi}_{\mathcal{A}}$. Thus if $(q^j_{\mathcal{A}}, q^j_{\mathcal{E}}) = \gamma^{\mathcal{A}}_e(\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}, \tilde{q}^j_{\mathcal{E}})$, then $q^j_{\mathcal{A}} \neq q^{\phi}_{\mathcal{A}}$, which means $\mathcal{A}$ is not permanently off in α. By contraposition, $\mathcal{A}$ is permanently off in $\alpha \implies \mathcal{A}$ is permanently absent in $\tilde{\alpha}$.

We obtained $\mathcal{A}$ is permanently off in $\alpha \iff \mathcal{A}$ is permanently absent in $\tilde{\alpha}$.

3. $|\tilde{\alpha}| = |\alpha|$.
   - (Case 1) $\tilde{\alpha} = \tilde{\alpha}' {}^\frown a\tilde{q}^n \iff \alpha = \alpha' {}^\frown a q^n$. $\mathcal{A}$ is permanently present in $\tilde{\alpha}' \iff \mathcal{A}$ is permanently on in $\tilde{\alpha}'$ and $\mathcal{A}$ is not permanently present in $\tilde{\alpha} \iff \mathcal{A}$ is not permanently on in $\tilde{\alpha}$. Thus α is a $\mathcal{A}$-segment $\iff$ is a $\mathcal{A}$-filled-segment.
   - (Case 2) $\tilde{\alpha} = \tilde{q}^0 \iff \alpha = q^0$. In this case α is not a $\mathcal{A}$-segment and $\tilde{\alpha}$ is not a $\mathcal{A}$-filled-segment.

   We obtained α is a $\mathcal{A}$-segment $\iff \tilde{\alpha}$ is a $\mathcal{A}$-filled-segment.
4. By conjonction of (2) and (3)

◀

▶ **Lemma 112.** *Let $\mathcal{A}$ be a PSIOA. Let X be a $\mathcal{A}$-conservative PCA. Let X' be a $\mathcal{A}$-twin of X. Let $Y' = X' \setminus \{\mathcal{A}\}$. Let $(\tilde{\alpha}, \alpha) \in frags(X') \times frags(\tilde{\mathcal{A}}^{sw}||Y')$, s. t. no creation of $\mathcal{A}$ occurs in $\tilde{\alpha}$ and $\mu^{\mathcal{A}}_e(\tilde{\alpha}) = \alpha$. Then*

1. *$\mathcal{A}$ is permanently present in $\alpha \iff \mathcal{A}$ is permanently present in $\tilde{\alpha}$.*
2. *$\mathcal{A}$ is permanently absent in $\alpha \iff \mathcal{A}$ is permanently absent in $\tilde{\alpha}$.*
3. *α is a $\mathcal{A}$-filled-segment $\iff \tilde{\alpha}$ is a $\mathcal{A}$-filled-segment.*
4. *$\alpha = \alpha^1 {}^\frown \alpha^2$ where α^1 is a $\mathcal{A}$-filled-segment and $\mathcal{A}$ is permanently present in $\alpha^2 \iff \tilde{\alpha} = \tilde{\alpha}^1 {}^\frown \tilde{\alpha}^2$ where $\tilde{\alpha}^1$ is a $\mathcal{A}$-filled-segment and $\mathcal{A}$ is permanently absent in $\tilde{\alpha}^2$ and $\mu^{\mathcal{A}}_e(\tilde{\alpha}^i) = \alpha^i$ for $i \in \{1,2\}$.*

Proof. For each state $(q^j_{\tilde{\mathcal{A}}^{sw}}, q^j_{Y'}) = \mu_z(q^j_{X'})$, $config(\tilde{\mathcal{A}}^{sw}||Y')((q^j_{\tilde{\mathcal{A}}^{sw}}, q^j_{Y'})) = config(X')(q^j_{X'})$, which gives the result immediatly. ◀

▶ **Lemma 113.** *Let $\mathcal{A}$ be a PSIOA. Let X be a $\mathcal{A}$-conservative PCA. Let $Y = X \setminus \{\mathcal{A}\}$. Let Y' be a $\mathcal{A}$-twin of PCA. Let $(\tilde{\alpha}, \alpha) \in frags(X) \times frags(\mathcal{A}||psioa(Y'))$, s. t. $\gamma^{\mathcal{A}}_e(\mu^{\mathcal{A}}_e(\tilde{\alpha})) = \alpha$. Then*

1. *$\mathcal{A}$ is permanently on in $\alpha \iff \mathcal{A}$ is permanently present in $\tilde{\alpha}$.*
2. *$\mathcal{A}$ is permanently off in $\alpha \iff \mathcal{A}$ is permanently absent in $\tilde{\alpha}$.*
3. *α is a $\mathcal{A}$-segment $\iff \tilde{\alpha}$ is a $\mathcal{A}$-filled-segment.*
4. *$\alpha = \alpha^1 {}^\frown \alpha^2$ where α^1 is a $\mathcal{A}$-segment and $\mathcal{A}$ is permanently off in $\alpha^2 \iff \tilde{\alpha} = \tilde{\alpha}^1 {}^\frown \tilde{\alpha}^2$ where $\tilde{\alpha}^1$ is a $\mathcal{A}$-filled-segment and $\mathcal{A}$ is permanently absent in $\tilde{\alpha}^2$ and $\mu^{\mathcal{A}}_e(\tilde{\alpha}^i) = \alpha^i$ for $i \in \{1,2\}$.*

Proof. By conjonction of the two last lemma. ◀

▶ **Definition 114** (Projection of configuration automaton into a contained SIOA)**.** Let $\mathcal{A}$ be a PSIOA. Let $\alpha = x_0 a_1 x_1 ... x_i a_{i+1} x_{i+1} ...$ be an execution of a configuration automaton X. Then $\alpha \upharpoonright\upharpoonright \mathcal{A}$ is a sequence of executions of $\mathcal{A}$, and results from the following steps:

1. insert a "delimiter" \$ after an action a_i whose execution causes $\mathcal{A}$ to set its signature to empty,

2. remove each $x_i a_{i+1}$ such that $\mathcal{A} \notin auts(X)(x_i)$,
3. remove each $x_i a_{i+1}$ such that $a_{i+1} \notin \widehat{sig}(\mathcal{A})(map(config(X)(x_i))(\mathcal{A}))$,
4. if α is finite, $x = last(\alpha)$, and $\mathcal{A} \notin auts(X)(x)$, then remove x,
5. replace each x_i by $map(config(X)(x_i))(\mathcal{A})$. $\alpha \upharpoonright\upharpoonright \mathcal{A}$ is, in general, a sequence of several (possibly an infinite number of) executions of $\mathcal{A}$, all of which are terminating except the last. That is, $\alpha \upharpoonright\upharpoonright \mathcal{A} = \alpha_1\$...\$\alpha_k$ where $(\forall j, 1 \leq j < k : \alpha_j \in texecs(\mathcal{A})) \wedge \alpha_k \in execs(\mathcal{A})$.

▶ **Definition 115** (Prefix relation among sequences of executions)**.** Let $\alpha^1\$...\$\alpha^k$ and $\delta^1\$...\$\delta^\ell$ be sequences of executions of some SIOA. Define $\alpha^1\$...\$\alpha^k \leq \delta^1\$...\$\delta^\ell$ iff $k \leq \ell \wedge (\forall j, 1 \leq j < k : \alpha^j = \delta^j) \wedge \alpha^k \leq \delta^k$. If $\alpha^1\$...\$\alpha^k \leq \delta^1\$...\$\delta^\ell$ and $\alpha^1\$...\$\alpha^k \neq \delta^1\$...\$\delta^\ell$ then we write $\alpha^1\$...\$\alpha^k < \delta^1\$...\$\delta^\ell$.

▶ **Definition 116** (Trace of a sequence of executions $strace_A(\alpha_1\$...\$\alpha_k)$)**.** Let $\alpha_1\$...\$\alpha_k$ be a sequence of executions of some SIOA A. Then $strace_A(\alpha_1\$...\$\alpha_k)$ is $trace_A(\alpha_1)\$...\$trace_A(\alpha_k)$, i.e., a sequence of traces of A, corresponding to the sequence of executions $\alpha_1\$...\$\alpha_k$.

▶ **Definition 117** ($\mathcal{A}$-partition of an execution)**.** Let $\mathcal{A}$ be a PSIOA. Let $K_\mathcal{A}$ be a PCA. Let α be an execution of $K_\mathcal{A}$. A $\mathcal{A}$-partition of α is a sequence $(\alpha^1, \alpha^2, ..., \alpha^n)$ of execution fragments s. t. $\alpha = \alpha^1 \frown \alpha^2 ... \frown \alpha^n$ and

- $\forall i \in [1 : n] \setminus \{1, n\}$ α^i is a $\mathcal{A}$ segment.
- Either α^n is a $\mathcal{A}$-segment or is $\mathcal{A}$-permanent.
- Either α^1 is a $\mathcal{A}$-segment or is $\mathcal{A}$-permanent and $n = 1$.

▶ **Lemma 118.** *Let $\mathcal{A}$ be a PSIOA. Let $K_\mathcal{A}$ be a PCA. Let α be a finite execution of $K_\mathcal{A}$. It exists a unique $\mathcal{A}$-partition of α.*

Proof. By induction on the number k of states in α. Basis: $\alpha = q^0$. (α^1) with $\alpha^1 = q^0$ is the unique partition of α with $n = 1$. If $\mathcal{A}$ is present in q^0, and $\mathcal{A}$ is permanently present, otherwise $\mathcal{A}$ is absent in α^1, and $\mathcal{A}$ is permanently absent α^1. Induction: We assume the predicate is true for k states in α and we want to show this is also true for $\alpha' = \alpha \frown a^{k+1} q^{k+1}$. We have $(\alpha^1, ..., \alpha^n)$ the unique $\mathcal{A}$-partition of α. By definition, α^n is either a $\mathcal{A}$-segment or a $\mathcal{A}$-permanent. We deal with 8 cases:

- $\mathcal{A}$ is present in q^{k+1}.
  - α^n is a $\mathcal{A}$-segment.
    * α^n is a $\mathcal{A}$-filled-segment. $(\alpha^1, ..., \alpha^n, (q^k a^{k+1} q^{k+1}))$ is a $\mathcal{A}$-partition of α', with $(q^k a^{k+1} q^{k+1})$ a $\mathcal{A}$-unfilled-segment. Unicity: $(\alpha^1, ..., \alpha^n \frown a^{k+1} q^{k+1})$ is not a partition since $\alpha^n \frown a^{k+1} q^{k+1}$ is neither a $\mathcal{A}$-segment nor $\mathcal{A}$-permanent
    * α^n is a $\mathcal{A}$-unfilled-segment. $(\alpha^1, ..., \alpha^n, (q^k a^{k+1} q^{k+1}))$ is a $\mathcal{A}$-partition of α', with $(q^k a^{k+1} q^{k+1})$ a $\mathcal{A}$-permanent execution fragment where $\mathcal{A}$ is permanently present. Unicity: $(\alpha^1, ..., \alpha^n \frown a^{k+1} q^{k+1})$ is not a partition since $\alpha^n \frown a^{k+1} q^{k+1}$ is neither a $\mathcal{A}$-segment nor $\mathcal{A}$-permanent.
  - α^n is $\mathcal{A}$-permanent
    * $\mathcal{A}$ is permanently absent in α^n . $(\alpha^1, ..., \alpha^n \frown a^{k+1} q^{k+1})$ is a $\mathcal{A}$-partition of α', with $\alpha^n \frown a^{k+1} q^{k+1}$ a $\mathcal{A}$-unfilled-segment. Unicity: $(\alpha^1, ..., \alpha^n, (q^k a^{k+1} q^{k+1}))$ is not a partition since α^n is not a segment.
    * $\mathcal{A}$ is permanently present in α^n. $(\alpha^1, ..., \alpha^n \frown a^{k+1} q^{k+1})$ is a $\mathcal{A}$-partition of α', with $\mathcal{A}$ permanently present in $\alpha^n \frown a^{k+1} q^{k+1}$. Unicity: $(\alpha^1, ..., \alpha^n, (q^k a^{k+1} q^{k+1}))$ is not a partition since α^n is not a segment.
- $\mathcal{A}$ is absent in q^{k+1}
  - α^n is a $\mathcal{A}$-segment

- * α^n is a $\mathcal{A}$-filled-segment. $(\alpha^1, ..., \alpha^n, (q^k a^{k+1} q^{k+1}))$ is a $\mathcal{A}$-partition of α', with $\mathcal{A}$ permanently absent in $(q^k a^{k+1} q^{k+1})$. Unicity: $(\alpha^1, ..., \alpha^n \frown a^{k+1} q^{k+1})$ is not a partition since $\alpha^n \frown a^{k+1} q^{k+1}$ is neither a $\mathcal{A}$-segment nor $\mathcal{A}$-permanent.
  * α^n is a $\mathcal{A}$-unfilled-segment. $(\alpha^1, ..., \alpha^n, (q^k a^{k+1} q^{k+1}))$ is a $\mathcal{A}$-partition of α', where $(q^k a^{k+1} q^{k+1})$ is a $\mathcal{A}$-filled-segment. Unicity: $(\alpha^1, ..., \alpha^n \frown a^{k+1} q^{k+1})$ is not a partition since $\alpha^n \frown a^{k+1} q^{k+1}$ is neither a $\mathcal{A}$-segment nor $\mathcal{A}$-permanent.
- α^n is $\mathcal{A}$-permanent
  * $\mathcal{A}$ is permanently absent in α^n . $(\alpha^1, ..., \alpha^n \frown a^{k+1} q^{k+1})$ is a $\mathcal{A}$-partition of α', with $\mathcal{A}$ permanently absent in $\alpha^n \frown a^{k+1} q^{k+1}$. Unicity: $(\alpha^1, ..., \alpha^n, (q^k a^{k+1} q^{k+1}))$ is not a partition since α^n is not a segment.
  * $\mathcal{A}$ is permanently present in α^n . $(\alpha^1, ..., \alpha^n \frown a^{k+1} q^{k+1})$ is a $\mathcal{A}$-partition of α', where $\alpha^n \frown a^{k+1} q^{k+1}$ is a $\mathcal{A}$-filled-segment. Unicity: $(\alpha^1, ..., \alpha^n, (q^k a^{k+1} q^{k+1}))$ is not a partition since α^n is not a segment.

We covered all the possibilities and at each time, it exists a unique $\mathcal{A}$-partition, By induction this is true for every finite execution.

◀

▶ **Lemma 119.** *Let $\mathcal{A}$ be a PSIOA. Let $K_{\mathcal{A}}$ be a PCA. Let α be an execution of $K_{\mathcal{A}}$. Let (α^1) be the $\mathcal{A}$-partition of α.*

- *if α is an unfilled segment that is ends on $\mathcal{A}$ creation, then*
  - *$\mathcal{A}$ is absent at $fstate(\alpha)$ and $\alpha \upharpoonleft\upharpoonleft \mathcal{A} = map(config(K_{\mathcal{A}})(lstate(\alpha))(\mathcal{A})$.*
- *otherwise, either*
  - *$\mathcal{A}$ is present at $fstate(\alpha)$ and $\alpha \upharpoonleft\upharpoonleft \mathcal{A} = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})$ or*
  - *$\mathcal{A}$ is absent at $fstate(\alpha)$ and $\alpha \upharpoonleft\upharpoonleft \mathcal{A}$ is the empty sequence.*

Proof.
- if α is an unfilled segment that is ends on $\mathcal{A}$ creation.
  - $\mathcal{A}$ is absent at $fstate(\alpha)$: We apply the rule (2) until $lstate(\alpha)$ excluded and we apply the rule (5) for $lstate(\alpha)$.
- otherwise, either
  - $\mathcal{A}$ is present at $fstate(\alpha)$: $\alpha \upharpoonleft\upharpoonleft \mathcal{A} = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})$ (this a totology since $\alpha = \alpha^1$)
  - $\mathcal{A}$ is absent at $fstate(\alpha)$: We apply the rule (2) until $lstate(\alpha)$ excluded and we apply the rule (4) for $lstate(\alpha)$.

◀

▶ **Lemma 120.** *Let $\mathcal{A}$ be a PSIOA. Let $K_{\mathcal{A}}$ be a PCA. Let α be an execution of $K_{\mathcal{A}}$. Let (α^1, α^2) be the $\mathcal{A}$-partition of α., where α^1 ends on $\mathcal{A}$-creation, then*

$\mathcal{A}$ is absent at $fstate(\alpha)$ and $\alpha \upharpoonleft\upharpoonleft \mathcal{A} = (\alpha^2 \upharpoonleft\upharpoonleft \mathcal{A})$.

Proof. Since $n = 2$, α^1 is a segment, so this is a $\mathcal{A}$-unfilled-segment, we apply the rule (2) until $lstate(\alpha^1)$ excluded and we apply the projection to the rest of execution fragment that is to α^2. ◀

▶ **Lemma 121.** *Let $\mathcal{A}$ be a PSIOA. Let $K_{\mathcal{A}}$ be a PCA. Let α be an execution of $K_{\mathcal{A}}$. Let $(\alpha^1, \alpha^2, ..., \alpha^n)$ be the $\mathcal{A}$-partition of α.*

- *if α ends on an unfilled segment that is ends on $\mathcal{A}$ creation, then either*
  - *$\mathcal{A}$ is present at $fstate(\alpha)$ and*
    $\alpha \upharpoonleft\upharpoonleft \mathcal{A} = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^3 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2\lceil n/2 \rceil - 1} \upharpoonleft\upharpoonleft \mathcal{A}) \frown map(config(K_{\mathcal{A}})(lstate(\alpha))(\mathcal{A}))$ or

- $\mathcal{A}$ *is absent at* $fstate(\alpha)$ *and* $\alpha \upharpoonleft\upharpoonleft \mathcal{A} = (\alpha^2 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^4 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2(\lfloor n/2 \rfloor)} \upharpoonleft\upharpoonleft \mathcal{A})^\frown map(config(K_{\mathcal{A}})(lstate(\alpha))(\mathcal{A})$

- *otherwise either*
  - $\mathcal{A}$ *is present at* $fstate(\alpha)$ *and* $\alpha \upharpoonleft\upharpoonleft \mathcal{A} = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^3 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2\lceil n/2 \rceil -1} \upharpoonleft\upharpoonleft \mathcal{A})$ *or*
  - $\mathcal{A}$ *is absent at* $fstate(\alpha)$ *and* $\alpha \upharpoonleft\upharpoonleft \mathcal{A} = (\alpha^2 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^4 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2(\lfloor n/2 \rfloor)} \upharpoonleft\upharpoonleft \mathcal{A})$

Proof. By induction on the size n of the $\mathcal{A}-partition$. We already proved the basis. in the last two lemma. We assume this is true for integer n and we show this is true for $n+1$: Let $\alpha' = \alpha^1 {}^\frown \alpha^2 {}^\frown ... \alpha^n {}^\frown \alpha^{n+1} = \alpha^\frown \alpha^{n+1}$ Case 1 If α' ends on an unfilled segment that is ends on $\mathcal{A}$ creation, then α^n is a filled-segment. Case 1a If $\mathcal{A}$ is present in $fstate(\alpha)$, then $\alpha' \upharpoonleft\upharpoonleft \mathcal{A} = \alpha \upharpoonleft\upharpoonleft \mathcal{A}^\frown map(config(K_{\mathcal{A}})(lstate(\alpha))(\mathcal{A})$ $\alpha \upharpoonleft\upharpoonleft \mathcal{A} = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^3 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2\lceil n/2 \rceil -1} \upharpoonleft\upharpoonleft \mathcal{A})^\frown map(config(K_{\mathcal{A}})(lstate(\alpha))(\mathcal{A})$ and we refind the waited value. Case 1b If $\mathcal{A}$ is absent in $fstate(\alpha)$, then $(\alpha^2 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^4 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2(\lfloor n/2 \rfloor)} \upharpoonleft\upharpoonleft \mathcal{A})^\frown map(config(K_{\mathcal{A}})(lstate(\alpha))(\mathcal{A})$ and we refind the waited value.

Case 2 α' does not end on $\mathcal{A}$ creation.

Case 2a $\mathcal{A}$ is present in $fstate(\alpha)$

Case 2ai n even ($2\lceil n/2 \rceil - 1 = n-1$ and $2\lceil (n+1)/2 \rceil - 1 = n+1$) We have α^n unfilled-segment and $\mathcal{A}$ present in α^{n+1}, thus $\alpha' \upharpoonleft\upharpoonleft \mathcal{A} = \alpha \upharpoonleft\upharpoonleft \mathcal{A}^\frown(\alpha^{n+1} \upharpoonleft\upharpoonleft \mathcal{A}) = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^3 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2\lceil n/2 \rceil -1} \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^{2\lceil n+1/2 \rceil -1} \upharpoonleft\upharpoonleft \mathcal{A})$ and we find the waited value.

Case 2aii n odd ($2\lceil n/2 \rceil - 1 = n$ and $2\lceil n+1/2 \rceil - 1 = n$) We have α^n filled-segment and $\mathcal{A}$ absent in α^{n+1}, thus $\alpha' \upharpoonleft\upharpoonleft \mathcal{A} = \alpha \upharpoonleft\upharpoonleft \mathcal{A} = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^3 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2\lceil n/2 \rceil -1} \upharpoonleft\upharpoonleft \mathcal{A}) = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^3 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2\lceil n+1/2 \rceil -1} \upharpoonleft\upharpoonleft \mathcal{A})$ and we find the waited value.

Case 2b $\mathcal{A}$ is absent in $fstate(\alpha)$

Case 2bi n even ($2\lfloor n/2 \rfloor = n$ and $2\lfloor n+1/2 \rfloor = n$) We have α^n filled-segment and $\mathcal{A}$ absent in α^{n+1}, thus $\alpha' \upharpoonleft\upharpoonleft \mathcal{A} = \alpha \upharpoonleft\upharpoonleft \mathcal{A} = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^3 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2\lfloor n/2 \rfloor} \upharpoonleft\upharpoonleft \mathcal{A}) = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^3 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2\lfloor n+1/2 \rfloor} \upharpoonleft\upharpoonleft \mathcal{A})$ and we find the waited value.

Case 2bii n odd ($2\lfloor n/2 \rfloor = n-1$ and $2\lfloor n+1/2 \rfloor = n+1$) We have α^n unfilled-segment and $\mathcal{A}$ present in α^{n+1}, thus $\alpha' \upharpoonleft\upharpoonleft \mathcal{A} = \alpha \upharpoonleft\upharpoonleft \mathcal{A}^\frown(\alpha^{n+1} \upharpoonleft\upharpoonleft \mathcal{A}) = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^3 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2\lfloor n/2 \rfloor} \upharpoonleft\upharpoonleft \mathcal{A})^\frown(\alpha^{n+1} \upharpoonleft\upharpoonleft \mathcal{A}) = (\alpha^1 \upharpoonleft\upharpoonleft \mathcal{A})(\alpha^3 \upharpoonleft\upharpoonleft \mathcal{A})...(\alpha^{2\lfloor n+1/2 \rfloor} \upharpoonleft\upharpoonleft \mathcal{A})$ and we find the waited value.

All the cases have been covered.

◀

7.3 $\bar{S}_{\mathcal{AB}}$, $S_{\mathcal{AB}}$ relation

Here we define a relation between executions α and π that captures the fact that they are the same excepting for internal aspects of $\mathcal{A}$ and $\mathcal{B}$. To define this relation, we needed to take particular cares with destruction and creation of $\mathcal{A}$ and $\mathcal{B}$.

▶ **Definition 122** (Execution correspondence relation, $S_{\mathcal{ABE}}$)). Let $\mathcal{A}, \mathcal{B}$ be PSIOA, let $\mathcal{E}$ be an environment for both $\mathcal{A}$ and $\mathcal{B}$. Let α, π be executions of automata $\mathcal{A}||\mathcal{E}$ and $\mathcal{B}||\mathcal{E}$ respectively.

Then we say that α is in relation $S_{(AB\mathcal{E})}$ with π, denoted $\alpha S_{(AB\mathcal{E})} \pi$ if

1. $\mathcal{A}$ is permanently off in $\alpha \Longleftrightarrow \mathcal{B}$ is permanently off in π. $\mathcal{A}$ is permanently on in $\alpha \Longleftrightarrow \mathcal{B}$ is permanently on in π.

2. (*) $\mathcal{A}$ is turned off in $\alpha \Longleftrightarrow \mathcal{B}$ is turned off in π. If (*), we can note $\alpha = \alpha_1^\frown \alpha_2$ and $\alpha_1 = \alpha_1'^\frown a q_1$, where $\widehat{sig}(\mathcal{A})(lstate(\alpha_1) \upharpoonright \mathcal{A}) = \emptyset$, $\widehat{sig}(\mathcal{A})(lstate(\alpha_1') \upharpoonright \mathcal{A}) \neq \emptyset$ and we can note $\pi = \pi_1^\frown \pi_2$ similarly.
3. $\pi \upharpoonright \mathcal{E} = \alpha \upharpoonright \mathcal{E}$. If (*), $\pi_i \upharpoonright \mathcal{E} = \alpha_i \upharpoonright \mathcal{E}$ for $i \in \{1, 2\}$.
4. $trace_{\mathcal{B}||\mathcal{E}}(\pi) = trace_{\mathcal{A}||\mathcal{E}}(\alpha)$. If (*) $trace_{\mathcal{B}||\mathcal{E}}(\pi_i) = trace_{\mathcal{A}||\mathcal{E}}(\alpha_i)$ for $i \in \{1, 2\}$.
5. $ext(\mathcal{A})(fstate(\alpha) \upharpoonright \mathcal{A}) = ext(\mathcal{B})(fstate(\pi) \upharpoonright \mathcal{B})$; $ext(\mathcal{A})(lstate(\alpha) \upharpoonright \mathcal{A}) = ext(\mathcal{B})(lstate(\pi) \upharpoonright \mathcal{B})$.

$S_{\mathcal{A}\mathcal{B}\mathcal{E}}$ is sometimes written $S_{\mathcal{A}\mathcal{B}}$ hen the environment is clear in the context.

The definition captures the fact that α and π only differs in the internal state and internal actions of $\mathcal{A}$ and $\mathcal{B}$. The conditions (1) and (2) say that $\mathcal{A}$ and $\mathcal{B}$ are destroyed in the same tempo in α and π. The condition (3) says α and π are the same executions from the common environment's point of view, condition (4) says the trace are equal, that is the actions can only differs in in the internal actions of $\mathcal{A}$ and $\mathcal{B}$.

▶ Remark. It is possible to have $(\alpha, \alpha') \in execs(\mathcal{A}||\mathcal{E})^2$ and $\alpha S_{\mathcal{A}\mathcal{A}\mathcal{E}} \alpha'$, that is α' and α only differs on internal state and internals action of $\mathcal{A}$. We note $S_{\mathcal{A}\mathcal{E}}$ to simplify $S_{\mathcal{A}\mathcal{A}\mathcal{E}}$ or even $S_{\mathcal{A}}$ when the environment is clear in the context .

▶ **Lemma 123.** *For every PSIOA $\mathcal{A}$, for every environment $\mathcal{E}$ of $\mathcal{A}$, $S_{\mathcal{A}}$ is an equivalence relation on $frags(\mathcal{A}||\mathcal{E})$.*

Proof. The conjonction of equivalence relations is an equivalence relation. (1), (2) are equivalence relation since the predicates are linked by the the equivalence relation $\Longleftrightarrow$. (3) (4) and (5) are equivalence relation since the predicates are linked by the the equivalence relation $=$. ◀

▶ **Lemma 124.** *Let $\mathcal{A}, \mathcal{B}$ be PSIOA, let $\mathcal{E}$ be an environment for both $\mathcal{A}$ and $\mathcal{B}$ Let $(\alpha, \alpha') \in frags(\mathcal{A}||\mathcal{E}), (\pi, \pi') \in frags(\mathcal{B}||\mathcal{E})$, s. t. $\alpha S_{\mathcal{A}} \alpha'$, $\pi S_{\mathcal{B}} \pi'$ and $\alpha' S_{\mathcal{A}\mathcal{B}} \pi'$*

Then $\alpha S_{\mathcal{A}\mathcal{B}} \pi$.

Proof. Each relation is true for α' and π'. By equivalence, each relation stay true for α and π. By conjonction of all the relations, the relation stays true for $S_{\mathcal{A}\mathcal{B}}$. ◀

▶ **Definition 125** (Execution correspondence relation, $\bar{S}_{\mathcal{A}\mathcal{B}}$)**.** Let $\mathcal{A}, \mathcal{B}$ be PSIOA. Let $K_{\mathcal{A}}, K_{\mathcal{B}}$ be PCA. Let α, π be execution fragments of configuration automata $K_{\mathcal{A}}, K_{\mathcal{B}}$ respectively. Then we say that α is in relation $\bar{S}_{AB}$ with π, denoted $\alpha \bar{S}_{AB} \pi$ iff

1. The partitions $(\alpha_1, ..., \alpha_n)$ and $(\pi_1, ..., \pi_n)$ of α and π respectively have the same size n.
2. $\forall i \in [1:n]$, (*) $\mathcal{A} \in auts(config(K_{\mathcal{A}})(fstate(\alpha_i))) \Longleftrightarrow \mathcal{B} \in auts(config(K_{\mathcal{B}})(fstate(\pi_i)))$ and (**) $\mathcal{B} \in auts(config(K_{\mathcal{B}})(lstate(\alpha_i))) \Longleftrightarrow \mathcal{B} \in auts(config(K_{\mathcal{B}})(lstate(\pi_i)))$
3. $\forall i \in [1:n]$, for every automaton $aut \neq \{\mathcal{A}, \mathcal{B}\}$ $\pi_i \upharpoonright\upharpoonright aut = \alpha_i \upharpoonright\upharpoonright aut$.
4. $\forall i \in [1:n]$ $trace_{K_{\mathcal{B}}}(\pi_i) = trace_{K_{\mathcal{A}}}(\alpha_i)$
5. $\forall i \in [1:n]$, if (*) $ext(\mathcal{A})(map(config(K_{\mathcal{A}})(fstate(\alpha_i)))(\mathcal{A})) = ext(\mathcal{B})(map(config(K_{\mathcal{B}})(fstate(\pi_i)))(\mathcal{B}))$; if (**) $ext(\mathcal{A})(map(config(K_{\mathcal{A}})(lstate(\alpha_i)))(\mathcal{A})) = ext(\mathcal{B})(map(config(K_{\mathcal{B}})(lstate(\pi_i)))(\mathcal{B}))$.

▶ Remark. It is possible to have $(\alpha, \alpha') \in execs(K_{\mathcal{A}})^2$ and $\alpha \bar{S}_{\mathcal{A}\mathcal{A}} \alpha'$, that is α' and α only differs on internal state and internals action of $K_{\mathcal{A}}$. We note $\bar{S}_{\mathcal{A}}$ to simplify $\bar{S}_{\mathcal{A}\mathcal{A}}$.

▶ **Lemma 126.** *Let $\mathcal{A} \in Autids$, $K_{\mathcal{A}}$ be a PCA. $\bar{S}_{\mathcal{A}}$ is an equivalence relation on $frags(K_{\mathcal{A}})$.*

Proof. The conjonction of equivalence relations is an equivalence relation. (2) is an equivalence relation since the predicates are linked by the the equivalence relation $\Longleftrightarrow$. (1), (3), (4) and (5) are equivalence relation since the predicates are linked by the the equivalence relation $=$. ◀

▶ **Lemma 127.** *Let $\mathcal{A} \in Autids$, $K_{\mathcal{A}}$ be a PCA. Let $(\alpha, \alpha') \in frags(K_{\mathcal{A}}), (\pi, \pi') \in frags(K_{\mathcal{B}})$, s. t. $\alpha \bar{S}_{\mathcal{A}} \alpha'$, $\pi \bar{S}_{\mathcal{B}} \pi'$ and $\alpha' \bar{S}_{\mathcal{AB}} \pi'$*

Then $\alpha \bar{S}_{\mathcal{AB}} \pi$.

Proof. Each relation is true for α' and π'. By equivalence, each relation stay true for α and π. By conjonction of all the relations, the relation stays true for $\bar{S}_{\mathcal{AB}}$. ◀

▶ Proposition 2. Let α, π be executions of configuration automata $K_{\mathcal{A}}, K_{\mathcal{B}}$ respectively. If $\alpha \bar{S}_{AB} \pi$, then $trace_{K_{\mathcal{A}}}(\alpha) = trace_{K_{\mathcal{B}}}(\pi)$

Proof. By clause 1 and 5 of the definition $\bar{S}_{AB}$. ◀

Equivalence class:

▶ **Definition 128** (equivalence class)**.** Let $\mathcal{A}$ be a PSIOA. Let $\mathcal{E}$ be an environment of $\mathcal{A}$. Let α be an execution fragment of $\mathcal{A}||\mathcal{E}$. We note $\underline{\alpha}_{\mathcal{AE}} = \{\alpha' | \alpha' S_{\mathcal{A}} \alpha\}$ Let $K_{\mathcal{A}}$ be a PCA. Let $\tilde{\alpha}$ be an execution fragment of $K_{\mathcal{A}}$. We note $\underline{\tilde{\alpha}}_{\mathcal{A}} = \{\tilde{\alpha}' | \tilde{\alpha}' \bar{S}_{\mathcal{A}} \tilde{\alpha}\}$.

When this is clear in the context, we note $\underline{\alpha}_{\mathcal{A}}$ or even $\underline{\alpha}$ for $\underline{\alpha}_{\mathcal{AE}}$ and $\underline{\tilde{\alpha}}$ for $\underline{\tilde{\alpha}}_{\mathcal{A}}$.

▶ **Lemma 129.** *Let $\mathcal{A}$ be a PSIOA. Let $K_{\mathcal{A}}$ be a PCA. Let α be an execution of $K_{\mathcal{A}}$. Let $(\alpha^1, \alpha^2, ..., \alpha^n)$ be the $\mathcal{A}$-partition of α.*

$\underline{\alpha} = \{\tilde{\alpha}^1 \frown \tilde{\alpha}^2 \frown ... \tilde{\alpha}^n | \tilde{\alpha}^i \bar{S}_{\mathcal{A}} \alpha^i \forall i \in [1:n]\}$

Proof. By induction on the size n of the partition. The basis is a tautology. Induction we assume this is true for integer n. Let $\alpha' = \alpha \frown \alpha^{n+1}$ and $(\alpha^1, ..., \alpha^n)$ the $\mathcal{A}$-partition of α and $(\alpha^1, ..., \alpha^n, \alpha^{n+1})$ the $\mathcal{A}$-partition of α'. We show $\underline{\alpha'} = \{\tilde{\alpha}^1 \frown \tilde{\alpha}^2 \frown ... \tilde{\alpha}^n \frown \tilde{\alpha}^{n+1} | \tilde{\alpha}^i \bar{S}_{\mathcal{A}} \alpha^i \forall i \in [1:n+1]\}$ by double inclusion.

Let $\tilde{\alpha}' \in \underline{\alpha'}$ with $(\tilde{\alpha}^1, \tilde{\alpha}^2, ..., \tilde{\alpha}^n, \tilde{\alpha}^{n+1})$ as $\mathcal{A}$-partition. We have $\tilde{\alpha}' = \tilde{\alpha}'_a \frown \tilde{\alpha}'_b$ with $\tilde{\alpha}'_a \in \underline{\alpha}$. By construction, the conditions (2), (3), (4), (5), (6) of definition of $\bar{S}_{\mathcal{AB}}$ are met for $\tilde{\alpha}^{n+1}$ and α^{n+1}. The condition (1) is met since $(\tilde{\alpha}^{n+1})$ is the $\mathcal{A}$-partition of $\tilde{\alpha}^{n+1}$ and (α^{n+1}) is the $\mathcal{A}$-partition of α^{n+1}. Hence $\tilde{\alpha}^{n+1} \bar{S}_{\mathcal{AB}} \alpha^{n+1}$. Thus $\underline{\alpha'} \subset \{\tilde{\alpha}^1 \frown \tilde{\alpha}^2 \frown ... \tilde{\alpha}^n \frown \tilde{\alpha}^{n+1} | \tilde{\alpha}^i \bar{S}_{\mathcal{A}} \alpha^i \forall i \in [1:n+1]\}$.

Let $\tilde{\alpha}' = \tilde{\alpha}^1 \frown \tilde{\alpha}^2 \frown ... \tilde{\alpha}^n \frown \tilde{\alpha}^{n+1}$ with $\tilde{\alpha}^i \bar{S}_{\mathcal{A}} \alpha^i \forall i \in [1:n+1]$. Thus $(\tilde{\alpha}^1, \tilde{\alpha}^2, ..., \tilde{\alpha}^n, \tilde{\alpha}^{n+1})$ is the $\mathcal{A}$-partition of $\tilde{\alpha}'$. By construction, the conditions (2), (3), (4), (5), (6) of definition of $\bar{S}_{\mathcal{AB}}$ are met for each i for $\tilde{\alpha}^i$ and α^i. The condition (1) is also met by construction with a size of $n+1$. Thus $\tilde{\alpha}' \in \underline{\alpha'}$. We have shown that if the claim was true for a partition of size n, it was also true for a partition of size $n+1$. Furthermore, the claim is true for $n=1$. Thus, by induction this is true for every integer n which ends the proof.

◀

▶ **Lemma 130** (μ_e preserves the equivalence relation intra automaton)**.** *Let $\mathcal{A}$ be a PSIOA. Let $X_{\mathcal{A}}$ be a $\mathcal{A}$-conservative PCA. Let $\mathcal{E}$ be an environment of $X_{\mathcal{A}}$. Let $\tilde{\alpha}, \tilde{\alpha}'$ be execution fragments of PCA $X_A||\mathcal{E}$ s. t. no creation of $\mathcal{A}$ occurs in $\tilde{\alpha}$. We note $\mathcal{E}' = (X_{\mathcal{A}} \setminus \mathcal{A})||\mathcal{E} = (X_{\mathcal{A}}||\mathcal{E}) \setminus \mathcal{A}$. We have $\mu_e(\tilde{\alpha}), \mu_e(\tilde{\alpha}') \in frags(\tilde{A}^{sw}||\mathcal{E}')$ and*

$\tilde{\alpha}\bar{S}_{\mathcal{A}}\tilde{\alpha}' \iff \mu_e(\tilde{\alpha})\bar{S}_{\mathcal{A}}\mu_e(\tilde{\alpha}')$.

Proof. For every state $\tilde{q}^j = (\tilde{q}^j_{X_{\mathcal{A}}}, \tilde{q}^j_{X_{\mathcal{A}}})$ and $q^j = \mu_z^{\mathcal{A}}(\tilde{q}^j) = (\tilde{q}^j_{\tilde{\mathcal{A}}^{sw}}, \tilde{q}^j_{\mathcal{E}'})$, $config(X_A||\mathcal{E})(\tilde{q}^j) = config(\tilde{A}^{sw}||\mathcal{E}')(q^j)$. Namely $\mathcal{A} \in auts(config(X_A||\mathcal{E})(\tilde{q}^j)) \iff \mathcal{A} \in auts(config(\tilde{A}^{sw}||\mathcal{E}')(q^j))$. Thus the respect of condition (1) is equivalent and we can reason by segment of the partition. For the same reason, the respect of condition (1) is equivalent. Since the configuration are the same and the actions are the same, the respect of condition (3) is equivalent. Since the actions are the same, then the external actions are the same and the respect of condition (4) is equivalent. Since the configuration are the same, the external signature of $\mathcal{A}$ in case of presence is the same and the respect of condition (5) is equivalent. Thus for every $i \in \{1,2,3,4,5\}$, $\tilde{\alpha}$ and $\tilde{\alpha}'$ respect the condition i of $\bar{S}_{\mathcal{A}} \iff \mu_e(\tilde{\alpha})$ and $\mu_e(\tilde{\alpha}')$ respect the condition i of $\bar{S}_{\mathcal{A}}$. This gives a fortiori $\tilde{\alpha}\bar{S}_{\mathcal{A}}\tilde{\alpha}' \iff \mu_e(\tilde{\alpha})\bar{S}_{\mathcal{A}}\mu_e(\tilde{\alpha}')$. ◀

▶ **Lemma 131** (γ preserves the equivalence relation intra automata)**.** *Let $\mathcal{A}$ be a PSIOA. Let $\tilde{\mathcal{A}}^{sw}$ be its simpleton wrapper. Let $\mathcal{E}$ be an environment of $\tilde{\mathcal{A}}^{sw}$ and $\mathcal{E}' = psioa(\mathcal{E})$.*

Let $\tilde{\alpha}, \tilde{\alpha}'$ be execution fragments of PCA $\tilde{\mathcal{A}}^{sw}||\mathcal{E}$ We have $\gamma_e(\tilde{\alpha}), \gamma_e(\tilde{\alpha}') \in frags(\mathcal{A}||\mathcal{E}')$ and

$\tilde{\alpha}\bar{S}_{\mathcal{A}}\tilde{\alpha}' \iff \gamma_e(\tilde{\alpha})\bar{S}_{\mathcal{A}\mathcal{E}'}\gamma_e(\tilde{\alpha}')$.

Proof. We have to deal with 4 cases:

- $(\tilde{\alpha}^1)$ is a $\mathcal{A}$-partition of $\tilde{\alpha}$ where $\mathcal{A}$ is permanently absent in $\tilde{\alpha}^1$. This is equivalent to $\mathcal{A}$ is permanently off in $\gamma_e(\tilde{\alpha}^1)$. We have $\tilde{\alpha}\bar{S}_{\mathcal{A}}\tilde{\alpha}' \iff \tilde{\alpha} = \tilde{\alpha}' \iff \gamma_e(\tilde{\alpha}) = \gamma_e(\tilde{\alpha}') \iff \gamma_e(\tilde{\alpha})S_{\mathcal{A}\mathcal{E}'}\gamma_e(\tilde{\alpha})'$.
- $(\tilde{\alpha}^1)$ is a $\mathcal{A}$-partition of $\tilde{\alpha}$ where $\mathcal{A}$ is permanently present in $\tilde{\alpha}^1$. This is equivalent to $\mathcal{A}$ is permanently on in $\gamma_e(\tilde{\alpha}^1)$.
  $\mathcal{A}$ is permanently present in $\tilde{\alpha}'$ because they have the same size of partition. Thus $\mathcal{A}$ is permanently on in both $\gamma_e(\tilde{\alpha}')$ and $\gamma_e(\tilde{\alpha})$, which implies that conditions (1) and (2) are met for $S_{\mathcal{A}}$. Also if the conditions (1) and (2) are met for $S_{\mathcal{A}\mathcal{B}}$, with $\mathcal{A}$ permanently on in $\gamma_e(\tilde{\alpha})$ and $\gamma_e(\tilde{\alpha})'$, then the second condition is met for $\bar{S}_{\mathcal{A}}$ with (**) true , while the condition (1) is verified with size 1. So the conditions (1) and (2) for $\bar{S}_{\mathcal{A}}$ are equivalent to the conditions (1) and (2) for $S_{\mathcal{A}\mathcal{E}'}$. The conditions (3) and (4) for $\bar{S}_{\mathcal{A}}$ are equivalent to the condition (3) for $S_{\mathcal{A}\mathcal{E}'}$. The condition (5) for $\bar{S}_{\mathcal{A}}$ is equivalent to the condition (4) for $S_{\mathcal{A}\mathcal{E}'}$ since the actions are not modified by γ_e. The condition (6) for $\bar{S}_{\mathcal{A}}$ is equivalent to the condition (5) for $S_{\mathcal{A}\mathcal{E}'}$.
  Thus $\tilde{\alpha}\bar{S}_{\mathcal{A}}\tilde{\alpha}' \iff \gamma_e(\tilde{\alpha})S_{\mathcal{A}}\gamma_e(\tilde{\alpha})'$.
- $(\tilde{\alpha}^1)$ is a $\mathcal{A}$-partition of $\tilde{\alpha}$ where $\tilde{\alpha}^1$ ends on $\mathcal{A}$ destruction.
  This is the same than in the previous point, excepting that the fact that $\tilde{\alpha}^1$ is a $\mathcal{A}$-filled-segment is equivalent to the fact that $\gamma_e(\tilde{\alpha}^1)$ is a $\mathcal{A}$-segment and the conditions the conditions (1) and (2) for $\bar{S}_{\mathcal{A}}$ are equivalent to the conditions (1) and (2) for $S_{\mathcal{A}\mathcal{E}'}$ with (**) false.
- $(\tilde{\alpha}^1, \tilde{\alpha}^2)$ is a $\mathcal{A}$-partition of $\tilde{\alpha}$ where $\tilde{\alpha}^1$ ends on $\mathcal{A}$ destruction and $\mathcal{A}$ is permanently absent in $\tilde{\alpha}^2$.
  This is the conjonction of the two last points.

◀

▶ **Lemma 132** (μ_e preserves the equivalence relation intra automaton)**.** *Let $\mathcal{A}$ be a PSIOA. Let $X_{\mathcal{A}}$ be a $\mathcal{A}$-conservative PCA. Let $\mathcal{E}$ be an environment of $X_{\mathcal{A}}$. Let $\tilde{\alpha}, \tilde{\alpha}'$ be execution fragments of PCA $X_A||\mathcal{E}$ s. t. no creation of $\mathcal{A}$ occurs in $\tilde{\alpha}$. We note $\mathcal{E}' = psioa(X_{\mathcal{A}} \setminus \mathcal{A}||\mathcal{E})$.*

We have $\gamma_e(\mu_e(\tilde{\alpha})), \gamma_e(\mu_e(\tilde{\alpha}')) \in frags(\mathcal{A}||\mathcal{E}')$ *and*
$\tilde{\alpha}\bar{S}_{\mathcal{A}}\tilde{\alpha}' \iff \gamma_e(\mu_e(\tilde{\alpha}))\bar{S}_{\mathcal{A}\mathcal{E}'}\gamma_e(\mu_e(\tilde{\alpha}')).$

Proof. By conjonction of the two last lemma. ◀

▶ **Lemma 133** (μ_e preserves the equivalence class)**.** *Let* $\mathcal{A}$ *be a PSIOA. Let* $X_{\mathcal{A}}$ *be a* $\mathcal{A}$*-conservative configuration-equivalence-free PCA. Let* $\mathcal{E}$ *be an environment of* $X_{\mathcal{A}}$*.*

Let $\tilde{\alpha}$ *be an execution fragments of PCA* $X_A||\mathcal{E}$ *s. t. no creation of* $\mathcal{A}$ *occurs in* $\tilde{\alpha}$*.*

Then $\underline{\mu_e(\tilde{\alpha})} = \mu_e(\underline{\tilde{\alpha}})$*.*

Proof. We have

$\mu_e(\underline{\tilde{\alpha}}) = \mu_e(\{\tilde{\alpha}' \in frags(X_{\mathcal{A}}||\mathcal{E})|\tilde{\alpha}'\bar{S}_{\mathcal{A}}\tilde{\alpha}\}) = \{\mu_e(\tilde{\alpha}')|\tilde{\alpha}' \in frags(X_{\mathcal{A}}||\mathcal{E}), \tilde{\alpha}'\bar{S}_{\mathcal{A}}\tilde{\alpha}\})$

and

$\underline{\mu_e(\tilde{\alpha})} = \{\alpha' \in frags(\tilde{\mathcal{A}}^{sw}||\mathcal{E}')|\alpha'\bar{S}_{\mathcal{A}}\mu_e(\tilde{\alpha})\}$ with $\mathcal{E}' = X_{\mathcal{A}} \setminus \{\mathcal{A}\}||\mathcal{E}$.

Since $\tilde{\alpha}'$ does not create $\mathcal{A}$, because of partial bijectivity, $\underline{\mu_e(\tilde{\alpha})} = \{\mu_e(\tilde{\alpha}')|\tilde{\alpha}' \in frags(X_{\mathcal{A}}||\mathcal{E}), \mu_e(\tilde{\alpha}')\bar{S}_{\mathcal{A}}\mu_e(\tilde{\alpha})\}$

Furthermore, $\tilde{\alpha}\bar{S}_{\mathcal{A}}\tilde{\alpha}' \iff \mu_e(\tilde{\alpha})S_{\mathcal{A}}\mu_e(\tilde{\alpha}')$ from the lemma. of preservation of $\bar{S}$ relation by μ_e.

So $\mu_e(\underline{\tilde{\alpha}}) = \underline{\mu_e(\tilde{\alpha})}$.

◀

▶ **Lemma 134** (γ_e preserves the equivalence class)**.** *Let* $\mathcal{A}$ *be a PSIOA. Let* $\tilde{\mathcal{A}}^{sw}$ *be its simpleton wrapper. Let* $\mathcal{E}$ *be an environment of* $\tilde{\mathcal{A}}^{sw}$ *and* $\mathcal{E}' = psioa(\mathcal{E})$*. Let* $\tilde{\alpha} \in frags(\tilde{\mathcal{A}}^{sw}||\mathcal{E})$

Then $\gamma_e(\underline{\tilde{\alpha}}) = \underline{\gamma_e(\tilde{\alpha})}$*.*

Proof. We have

$\gamma_e(\underline{\tilde{\alpha}}) = \gamma_e(\{\tilde{\alpha}' \in frags(\tilde{\mathcal{A}}^{sw}||\mathcal{E})|\tilde{\alpha}'\bar{S}_{\mathcal{A}}\tilde{\alpha}\}) = \{\gamma_e(\tilde{\alpha}')|\tilde{\alpha}' \in frags(\tilde{\mathcal{A}}^{sw}||\mathcal{E}), \tilde{\alpha}'\bar{S}_{\mathcal{A}}\tilde{\alpha}\})$

and

$\underline{\gamma_e(\tilde{\alpha})} = \{\alpha' \in frags(\mathcal{A}||\mathcal{E}')|\alpha' S_{\mathcal{A}\mathcal{E}'}\gamma_e(\tilde{\alpha})\}$.

Because of bijectivity of γ_e, $\underline{\gamma_e(\tilde{\alpha})} = \{\gamma(\tilde{\alpha}')|\tilde{\alpha'} \in frags(\tilde{\mathcal{A}}^{sw}||\mathcal{E}), \gamma_e(\tilde{\alpha}')S_{\mathcal{A}\mathcal{E}'}\gamma_e(\tilde{\alpha})\}$

Furthermore, $\tilde{\alpha}\bar{S}_{\mathcal{A}}\tilde{\alpha}' \iff \gamma_e(\tilde{\alpha})S_{\mathcal{A}\mathcal{E}'}\gamma_e(\tilde{\alpha}')$ from the lemma of preservation of S relation by γ_e.

So $\gamma_e(\underline{\tilde{\alpha}}) = \underline{\gamma_e(\tilde{\alpha})}$.

◀

▶ **Lemma 135** ($\gamma_e \circ \mu_e$ preserves the equivalence class)**.** *Let* $\mathcal{A}$ *be a PSIOA. Let* $X_{\mathcal{A}}$ *be a* $\mathcal{A}$*-conservative configuration-equivalence-free PCA. Let* $\mathcal{E}$ *be an environment of* $X_{\mathcal{A}}$*.*

Let $\tilde{\alpha}$ *be an execution fragments of PCA* $X_A||\mathcal{E}$ *s. t. no creation of* $\mathcal{A}$ *occurs in* $\tilde{\alpha}$*.*

Then $\underline{\gamma_e(\mu_e(\tilde{\alpha}))} = \gamma_e(\mu_e(\underline{\tilde{\alpha}}))$*.*

Proof. By conjonction of the two last lemma. ◀

▶ **Theorem 136** (Preserving probabilistic distribution without creation for equivalence class)**.** *Let $\mathcal{A} \in Autids$. Let X be a $\mathcal{A}$-conservative PCA. Let X' be a $\mathcal{A}$-twin of $\mathcal{A}$. Let $Y' = X' \setminus \mathcal{A}$. Let $\mathcal{Z} = \tilde{A}^{sw}||Y'$. Let $\mathcal{E}$ be an environment of X'. Let $\mathcal{E}' = psioa(Y'||\mathcal{E})$. Let ρ be a schedule.*

For every execution fragment $\alpha = q^0 a^1 q^1 ... q^k \in frags(X||\mathcal{E})$, verifying:

- *No creation of $\mathcal{A}$: For every $s \in [0, k-1]$, if $\mathcal{A} \notin auts(config(X)(q^s_X))$ then $\mathcal{A} \notin created(X)(q^s_X)(a^{s+1})$.*
- *No creation from $\mathcal{A}$: $\forall s \in [0, k-1]$, verifying $a^{s+1} \notin sig(config(X)(q^s_X) \setminus \mathcal{A}) \wedge a^{s+1} \in sig(\mathcal{A})(q^s_{\mathcal{A}})$, with $\mu_z(q^s_X) = q_{\mathcal{Z}} = (q^s_Y, q^s_{\mathcal{A}})$, $created(X)(q^s_X)(a) = \emptyset$.*

then $apply_{X||\mathcal{E}}(\delta_{(q_X,q_{\mathcal{E}})}, \rho)(\underline{\alpha}) = apply_{(\mathcal{Z}||\mathcal{E})}(\delta_{(\mu_z(q_X),q_{\mathcal{E}})}, \rho)(\underline{\mu_e(\alpha)}) = apply_{(\mathcal{A}||\mathcal{E}')}(\delta_{(\gamma_s(\mu_z(q_X),q_{\mathcal{E}'}))}, \rho)(\underline{\gamma_e(\mu_e(\alpha))})$

Proof. We already have $apply_{X||\mathcal{E}}(\delta_{(q_X,q_{\mathcal{E}})}, \rho)(\alpha) = apply_{(\mathcal{Z}||\mathcal{E})}(\delta_{(\mu_z(q_X),q_{\mathcal{E}})}, \rho)(\mu_e(\alpha))$. Thus $\sum_{\alpha' \in \underline{\alpha}} apply_{X||\mathcal{E}}(\delta_{(q_X,q_{\mathcal{E}})}, \rho)(\alpha') = \sum_{\alpha' \in \underline{\alpha}} apply_{(\mathcal{Z}||\mathcal{E})}(\delta_{(\mu_z(q_X),q_{\mathcal{E}})}, \rho)(\mu_e(\alpha'))$. Hence, $apply_{X||\mathcal{E}}(\delta_{(q_X,q_{\mathcal{E}})}, \rho)(\underline{\alpha}) = apply_{(\mathcal{Z}||\mathcal{E})}(\delta_{(\mu_z(q_X),q_{\mathcal{E}})}, \rho)(\mu_e(\underline{\alpha'}))$. Furthermore, we know that $\mu_e(\underline{\tilde{\alpha}}) = \underline{\mu_e(\tilde{\alpha})}$, thus $apply_{X||\mathcal{E}}(\delta_{(q_X,q_{\mathcal{E}})}, \rho)(\underline{\alpha}) = apply_{(\mathcal{Z}||\mathcal{E})}(\delta_{(\mu_z(q_X),q_{\mathcal{E}})}, \rho)(\underline{\mu_e(\alpha')})$.

In the same manner, we obtain the second result with $\gamma_e(\mu_e(\underline{\tilde{\alpha}})) = \underline{\gamma_e(\mu_e(\tilde{\alpha}))}$.

◀

7.4 Implementation monotonicity without creation

▶ **Lemma 137** ($\bar{S}_{\mathcal{AB}}$-balanced distribution witout creation)**.** *Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $K_{\mathcal{A}}$, $K_{\mathcal{B}}$ be PCA corresponding w. r. t. $\mathcal{A}$ and $\mathcal{B}$. Let $K'_{\mathcal{A}}, K'_{\mathcal{B}}$ be $\mathcal{AB}$-co-twin of $K_{\mathcal{A}}$ and$K_{\mathcal{B}}$. Let $\mathcal{E}'_{\mathcal{A}} = K'_{\mathcal{A}} \setminus \mathcal{A}$, $\mathcal{E}'_{\mathcal{B}} = K'_{\mathcal{B}} \setminus \mathcal{B}$, $\mathcal{E}''_{\mathcal{A}} = psioa(\mathcal{E}'_{\mathcal{A}})$ and $\mathcal{E}''_{\mathcal{B}} = psioa(\mathcal{E}'_{\mathcal{B}})$. Let $\mathcal{E}'' = \mathcal{E}''_{\mathcal{A}}$ (or $\mathcal{E}'' = \mathcal{E}''_{\mathcal{B}}$, it does not matter).*

Let ρ, ρ' be schedule s. t. for every executions α, π of $\mathcal{A}||\mathcal{E}''$ and $\mathcal{B}||\mathcal{E}''$, verifying $\alpha S_{\mathcal{ABE}''}\pi$, $apply_{\mathcal{A}||\mathcal{E}''}(\delta_{(\bar{q}_{\mathcal{A}},\bar{q}_{\mathcal{E}''})}, \rho)(\underline{\alpha}) = apply_{\mathcal{B}||\mathcal{E}'}(\delta_{(\bar{q}_{\mathcal{B}},\bar{q}_{\mathcal{E}''})}, \rho')(\underline{\pi})$.

Let $q_{K_{\mathcal{A}}}$ s. t. $\mu^{\mathcal{A}}_z(q_{K_{\mathcal{A}}}) = (ren_{sw}(\bar{q}_{\mathcal{A}}), \bar{q}_{\mathcal{E}'_{\mathcal{A}}})$. Let $q_{K_{\mathcal{B}}}$ s. t. $\mu^{\mathcal{B}}_z(q_{K_{\mathcal{B}}}) = (ren_{sw}(\bar{q}_{\mathcal{B}}), \bar{q}_{\mathcal{E}'_{\mathcal{B}}})$.

Then for every execution fragments $\tilde{\alpha}$, $\tilde{\pi}$ of $K'_{\mathcal{A}}$ and $K'_{\mathcal{B}}$, verifying $\tilde{\alpha}\bar{S}_{\mathcal{AB}}\tilde{\pi}$ and $\tilde{\alpha}$ does not create $\mathcal{A}$, we have:

$apply_{K'_{\mathcal{A}}}(\delta_{q_{K_{\mathcal{A}}}}, \rho)(\underline{\tilde{\alpha}}) = apply_{K'_{\mathcal{B}}}(\delta_{q_{K_{\mathcal{B}}}}, \rho')(\underline{\tilde{\pi}})$.

Proof. Let $\tilde{\alpha}$, $\tilde{\pi}$ be execution fragments of $K'_{\mathcal{A}}$ and $K'_{\mathcal{B}}$, verifying $\tilde{\alpha}\bar{S}_{\mathcal{AB}}\tilde{\pi}$ with $\tilde{\alpha}$ that does not create $\mathcal{A}$.

We have

- $apply_{K'_{\mathcal{A}}}(\delta_{q_{K_{\mathcal{A}}}}, \rho)(\underline{\tilde{\alpha}}) = apply_{\tilde{A}^{sw}||\mathcal{E}'_{\mathcal{A}}}(\delta_{\mu^{\mathcal{A}}_z(q_{K_{\mathcal{A}}})}, \rho)(\mu^{\mathcal{A}}_e(\underline{\tilde{\alpha}})) = apply_{\mathcal{A}||\mathcal{E}''}(\delta_{(\bar{q}_{\mathcal{A}},\bar{q}_{\mathcal{E}''})}, \rho')(\gamma^{\mathcal{A}}_e(\mu^{\mathcal{A}}_e(\underline{\tilde{\alpha}}))) = apply_{\mathcal{A}||\mathcal{E}''}(\delta_{(\bar{q}_{\mathcal{A}},\bar{q}_{\mathcal{E}''})}, \rho')(\underline{\gamma^{\mathcal{A}}_e(\mu^{\mathcal{A}}_e(\tilde{\alpha}))})$
- $apply_{K'_{\mathcal{B}}}(\delta_{q_{K_{\mathcal{B}}}}, \rho')(\underline{\tilde{\pi}}) = apply_{\tilde{B}^{sw}||\mathcal{E}'_{\mathcal{B}}}(\delta_{\mu^{\mathcal{B}}_z(q_{K_{\mathcal{B}}})}, \rho')(\mu^{\mathcal{B}}_e(\underline{\tilde{\pi}})) = apply_{\mathcal{B}||\mathcal{E}''}(\delta_{(\bar{q}_{\mathcal{A}},\bar{q}_{\mathcal{E}''})}, \rho')(\gamma^{\mathcal{B}}_e(\mu^{\mathcal{B}}_e(\underline{\tilde{\pi}}))) = apply_{\mathcal{B}||\mathcal{E}''}(\delta_{(\bar{q}_{\mathcal{A}},\bar{q}_{\mathcal{E}''})}, \rho')(\underline{\gamma^{\mathcal{B}}_e(\mu^{\mathcal{B}}_e(\tilde{\pi}))})$.

Hence we have $apply_{K'_{\mathcal{A}}}(\delta_{q_{K_{\mathcal{A}}}}, \rho)(\underline{\tilde{\alpha}}) = apply_{K'_{\mathcal{B}}}(\delta_{q_{K_{\mathcal{B}}}}, \rho')(\underline{\tilde{\pi}})$

◀

▶ **Definition 138** ($S^s_{\mathcal{ABE}}$ relation for schedules)**.** Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $\mathcal{E}$ be an environment of both $\mathcal{A}$ and $\mathcal{B}$. Let ρ and ρ' be two schedules. We say that $\rho S^s_{(\mathcal{A},\mathcal{B},\mathcal{E})}\rho'$ if :

for every executions α, π of $\mathcal{A}||\mathcal{E}$ and $\mathcal{B}||\mathcal{E}$ respectively, s. t. $\alpha S_{\mathcal{ABE}}\pi$,

$apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A,\bar{q}_\mathcal{E})},\rho)(\underline{\alpha}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})},\rho')(\underline{\pi})$.

This definition says that each member of each pair of corresponding classes of equivalence deserve the same probability measure.

▶ **Theorem 139** (Monotonicity of S^s relation without creation). *Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $X_\mathcal{A}$, $X_\mathcal{B}$ be PCA corresponding w. r. t. $\mathcal{A}$ and $\mathcal{B}$. Let $\mathcal{E}$ be an environment for both $X_\mathcal{A}$, $X_\mathcal{B}$. Let $X'_\mathcal{A}||\mathcal{E}'$, $X'_\mathcal{B}||\mathcal{E}'$ be $\mathcal{AB}$-co-twin of $X_\mathcal{A}||\mathcal{E}$ and $X_\mathcal{B}||\mathcal{E}$. Let $\mathcal{E}''_\mathcal{A} = psioa(X'_\mathcal{A} \setminus \mathcal{A}||\mathcal{E}')$ and $\mathcal{E}''_\mathcal{B} = psioa(X'_\mathcal{B} \setminus \mathcal{B}||\mathcal{E}')$ Let $\mathcal{E}'' = \mathcal{E}''_\mathcal{A}$ (or $\mathcal{E}'' = \mathcal{E}''_\mathcal{B}$, it does not matter).*

Let ρ, ρ' be schedule s. t. $\rho S^s_{(\mathcal{A},\mathcal{B},\mathcal{E}'')}\rho'$. Then for every $(\alpha,\pi) \in execs(X'_\mathcal{A}||\mathcal{E}') \times execs(X'_\mathcal{B}||\mathcal{E}')$ that does not create $\mathcal{A}$ and $\mathcal{B}$ s. t. $\alpha S_{(X'_\mathcal{A}X'_\mathcal{B}\mathcal{E}')}\pi$

$apply_{X'_\mathcal{A}||\mathcal{E}'}(\delta_{(\bar{q}_{X'_\mathcal{A}},\bar{q}_{\mathcal{E}'})},\rho)(\underline{\alpha}) = apply_{X'_\mathcal{B}||\mathcal{E}'}(\delta_{(\bar{q}_{X'_\mathcal{B}},\bar{q}_{\mathcal{E}'})},\rho')(\underline{\pi})$.

Proof. By application of previous lemma with $K_\mathcal{A} = X_\mathcal{A}||\mathcal{E}$ and $K_\mathcal{B} = X_\mathcal{B}||\mathcal{E}$, since projection and composition are commutative. ◀

8 Monotonicity of implementation w. r. t. PSIOA creation and destruction

In last section we have shown a weak version of our final monotonicity theorem (160), where we only consider executions that do not create $\mathcal{A}$ (see theorem 139).

Here we want to show this is also true with the creation of $\mathcal{A}$ and $\mathcal{B}$.

8.1 schedule notations

▶ **Definition 140** (simple schedule notation). Let $\rho = T^\ell, T^{\ell+1}, ..., T^h, ...$ be a schedule, i. e. a sequence of tasks, beginning with T^ℓ and terminating by T^h if ρ is finite with $\ell, h \in \mathbb{N}^*$. For every $q, q' \in [\ell, h], q \leq q'$, we note:

- $hi(\rho) = h$ the highest index in ρ ($hi(\rho) = \omega$ if ρ is infinite)
- $li(\rho) = \ell$ the lowest index in ρ
- $\rho[q] = T^q$
- $\rho|_q = T^\ell ... T^q$
- $_q|\rho = T^q ... T^h ...$
- $_q|\rho|_{q'} = T^q ... T^{q'}$

By doing so, we implicitly assume an indexation of ρ, $ind(\rho) : ind \in [li(\rho), hi(\rho)] \mapsto T^{ind} \in \rho$. Hence if $\rho = T^1, T^2, ..., T^k, T^{k+1}, ..., T^q, T^{q+1} ..., T^h, ...$, $\rho' =_k |\rho$, $\rho'' =_q |\rho'$, then $\rho'' =_q |\rho$.

▶ **Definition 141** (Schedule partition and index). Let ρ be a schedule. A partition p of ρ is a sequence of schedules (finite or infinite) $p = (\rho^m, \rho^{m+1}, ..., \rho^n, ...)$ so that ρ can be written $\rho = \rho^m, \rho^{m+1}, ..., \rho^n, ...$ We note $min(p) = m$ and $max(p) = card(p) + m - 1$ (if p is infinite, $max(p) = \omega$).

A total ordered set $(ind(\rho,p), \prec) \subset \mathbb{N}^2$ is defined as follows :

$ind(\rho,p) = \{(k,q) \in (\mathbb{N}^*)^2 | k \in [min(p), max(p)], q \in [li(\rho^k), hi(\rho^k)]\}$ For every $\ell = (k,q), \ell' = (k',q') \in ind(\rho,p)$:

- If $k < k'$, then $\ell \prec \ell'$
- If $k = k', q < q'$, then $\ell \prec \ell'$
- If $k = k'$ and $q = q'$, then $\ell = \ell'$. If either $\ell \prec \ell'$ or $\ell = \ell'$, we note $\ell \preceq \ell'$.

For every $\ell = (k, q) \in ind(\rho, p)$, we note $\ell + 1$ the smaller element (according to $\prec$) of $ind(\rho, p)$ that is greater than ℓ. For convenience, we extend $ind(\rho, p)$ with $\{(k, 0) \in (\mathbb{N}^*)^2 | k \leq card(p)\}$, where $(k + 1, 0) \triangleq (k, card(\rho^k))$.

▶ **Definition 142** (Schedule notation)**.** Let ρ be a schedule. Let p be a partition of ρ. For every $\ell = (k, q), \ell' = (k', q') \in ind(\rho, p)^2$, we note (when this is allowed):

- $\rho[p, \ell] = \rho^k[q]$
- $\rho|_{(p,\ell)} = \rho^1, ..., \rho^k|_q$
- $_{(p,\ell)}|\rho = (_q|\rho^k), ...$
- $_\ell|\rho|_{(p,\ell')} = (_q|\rho^k), ..., (\rho^{k'}|_q)$

The symbol p of the partition is removed when it is clear in the context.

▶ **Definition 143** (Environment)**.** Let $\mathcal{V}$ be a PCA (resp a PSIOA). An environment $\mathcal{E}$ for $\mathcal{V}$ is a PCA (resp. a PSIOA) partially-compatible with $\mathcal{V}$ s. t. $UA(\mathcal{E}) \cap UA(\mathcal{V}) = \emptyset$

▶ **Definition 144** ($\mathcal{V}$-partition of a schedule)**.** Let $\mathcal{V}$ be a PCA or a PSIOA. Let $\rho_{\mathcal{V}\mathcal{E}}$ be a schedule. Let $p = (\rho^1_{\mathcal{V}}, \rho^2_{\mathcal{E}}, \rho^3_{\mathcal{V}}, \rho^4_{\mathcal{E}}...)$ be a partition of $\rho_{\mathcal{V}\mathcal{E}}$ where each $\rho^{2k+1}_{\mathcal{V}}$ is a sequence of tasks of $UA(\mathcal{V})$ only and each $\rho^{2k}_{\mathcal{E}}$ does not contain any task of $UA(\mathcal{V})$. We call such a partition, a *$\mathcal{V}$-partition* of $\rho_{\mathcal{V}\mathcal{E}}$.

▶ Proposition 3. Let $\rho_{\mathcal{V}\mathcal{E}}$ be a schedule. It exists a unique $\mathcal{V}$-partition of $\rho_{\mathcal{V}\mathcal{E}}$.

Proof. Since $UA(\mathcal{E}) \cap UA(\mathcal{V}) = \emptyset$ the partition exists. The uniqueness is also due to the fact that $UA(\mathcal{E}) \cap UA(\mathcal{V}) = \emptyset$. ◀

Thus, in the remaining we say *the* $\mathcal{V}$-partition of a schedule.

▶ **Definition 145** (Environment corresponding schedule)**.** Let $\mathcal{V}$ and $\mathcal{W}$ be two PCA or two PSIOA. Let $\rho_{\mathcal{V}\mathcal{E}}$ and $\rho_{\mathcal{W}\mathcal{E}}$ be two schedules. Let $(\rho^1_{\mathcal{V}}, \rho^2_{\mathcal{E}}, \rho^3_{\mathcal{V}}, \rho^4_{\mathcal{E}}...)$ (resp. $\rho_{\mathcal{W}\mathcal{E}}$: $(\rho^1_{\mathcal{W}}, \rho^{2'}_{\mathcal{E}}, \rho^3_{\mathcal{W}}, \rho^{4'}_{\mathcal{E}}, ...)$) be the $\mathcal{V}$-partition (resp. $\mathcal{W}$-partition) of $\rho_{\mathcal{V}\mathcal{E}}$ (resp. $\rho_{\mathcal{W}\mathcal{E}}$). We say that $\rho_{\mathcal{V}\mathcal{E}}$ and $\rho_{\mathcal{W}\mathcal{E}}$ are *$\mathcal{V}\mathcal{W}$-environment-corresponding* if for every k, $\rho^{2k}_{\mathcal{E}} = \rho^{2k'}_{\mathcal{E}}$.

Environment corresponding schedules only differ on the tasks that do not concerns the environment.

▶ **Definition 146** ($S^s_{\mathcal{A}\mathcal{B}\mathcal{E}}$ relation for schedules)**.** Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $\mathcal{E}$ be an environment of both $\mathcal{A}$ and $\mathcal{B}$. Let ρ and ρ' be two schedule. We say that $\rho S^s_{(\mathcal{A},\mathcal{B},\mathcal{E})} \rho'$ if :

for every executions α, π of $\mathcal{A}||\mathcal{E}$ and $\mathcal{B}||\mathcal{E}$ respectively, s. t. $\alpha S_{\mathcal{A}\mathcal{B}\mathcal{E}} \pi$,

$apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_{\mathcal{E}})}, \rho)(\underline{\alpha}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B, \bar{q}_{\mathcal{E}})}, \rho')(\underline{\pi})$.

This definition says that each member of each pair of corresponding classes of equivalence deserve the same probability measure.

8.2 sub-classes according to the schedule

▶ **Definition 147.** Let X be an automaton, let α be an execution of X, and $\rho = \rho' T$ be a schedule of X. We say that α match ρ iff $\alpha \in supp(apply_X(\delta_{fstate(\alpha)}, \rho))$ but $\alpha \notin supp(apply_X(\delta_{fstate(\alpha)}, \rho'))$.

If $\alpha \in supp(apply_X(\delta_{fstate(\alpha)}, \lambda))$, we say that α match λ (the empty sequence).

▶ **Definition 148.** Let α be an execution. Let ρ be a schedule, p be a fixed partition of ρ, $\ell_1, \ell_2, \ell_1^-, \ell_2^-, \ell_1^+, \ell_2^+ \in ind(\rho, p)$, we note :

- $\underline{\alpha}_{(\ell_1,\rho)} = \{\tilde{\alpha} \in \underline{\alpha} | \tilde{\alpha}$ matches $\rho|_{\ell_1}\}$
- $\underline{\alpha}_{(\ell_1,\ell_2,\rho)} = \{\tilde{\alpha} \in \underline{\alpha} | \tilde{\alpha}$ matches $_{\ell_1}|\rho|_{\ell_2}\}$
- $\underline{\alpha}_{(\ell_1,[\ell_2^-,\ell_2^+],\rho)} = \{\tilde{\alpha} \in \underline{\alpha} | \exists \ell^2 \in [\ell_2^-, \ell_2^+], \tilde{\alpha}$ matches $_{\ell_1}|\rho|_{\ell_2}\}$

▶ **Lemma 149.** *Let X be a PSIOA, α be an execution of X , ρ be a schedule of$X$, p be a fixed partition of ρ. $\{\underline{\alpha}_{\ell^+,\rho} \cap supp(apply_X(\delta_{fstate(\alpha)}, \rho)) | \ell^+ \in ind(\rho, p)\}$ is a partition of $\underline{\alpha} \cap supp(apply_X(\delta_{fstate(\alpha)}, \rho))\}$.*

Proof.
- empty intersection: Let $\ell, \ell' \in ind(\rho, p)$. Let $\alpha \in \underline{\alpha}_{\ell,\rho}$, we show that $\alpha \notin \underline{\alpha}_{\ell',\rho}$. By contradiction, we assume the contrary: thus, $\alpha \in supp(apply_X(\delta_{fstate(\alpha)}, \rho|_\ell))$, $\alpha \in supp(apply_X(\delta_{fstate(\alpha)}, \rho|_{\ell'}))$ but $\alpha \notin supp(apply_X(\delta_{fstate(\alpha)}, \rho|_{\ell-1}))$ and $\alpha \notin supp(apply_X(\delta_{fstate(\alpha)}, \rho|_{\ell'-1}))$.
If $\ell = \ell' + 1$ or $\ell' = \ell + 1$, the contradiction is immediate.
Without lost of generality, we assume $\ell' \prec \ell + 1$. Since $\alpha \in supp(apply_X(\delta_{fstate(\alpha)}, \rho|_\ell))$, $\alpha \in supp(apply_X(\delta_{fstate(\alpha)}, \rho|_{\ell'}))$, all the tasks in $_{\ell'+1}|\rho|_\ell$ are not enabled in $lstate(\alpha)$, but this is in contradiction with the fact that both $\alpha \in supp(apply_X(\delta_{fstate(\alpha)}, \rho|_{\ell'}))$ and $\alpha \notin supp(apply_X(\delta_{fstate(\alpha)}, \rho|_{\ell-1}))$.
- complete union: Let $\alpha' = \alpha''^\frown aq' \in supp(apply_X(\delta_{fstate(\alpha)}, \rho))$, with $q'' = lstate(\alpha'')$. We show it exists $\ell \in ind(\rho, p)$, so that α' matches $\rho|_\ell$. By contradiction, it means α' matches $\rho|_\ell$ for every $\ell \in ind(\rho, p)$, namely α' matches $\rho|_0 = \lambda$ (the empty sequence) and that for every task T in ρ, T is not enabled in q''. Thus $apply_X(\delta_{fstate(\alpha)}, \lambda)(\alpha') > 0$, which is in contradiction with $\alpha' \neq fstate(\alpha)$. If $\alpha' = q^0$ and for every task T in ρ, T is not enabled in q^0, then α' matches $\rho_0 = 0$.

◀

▶ **Lemma 150.** *Let X be a PSIOA, α be an execution of X , ρ be a schedule of X, p be a fixed partition of ρ.*

$$apply_X(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}) = \sum_{\ell^+ \in ind(\rho,p)} apply_X(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}_{\ell^+})$$

Proof. $\{\underline{\alpha}_{\ell^+,\rho} \cap supp(apply_X(\delta_{fstate(\alpha)}, \rho)) | \ell^+ \in ind(\rho, p)\}$ is a partition of $\underline{\alpha} \cap supp(apply_X(\delta_{fstate(\alpha)}, \rho))\}$, which gives $apply_X(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}) = \sum_{\ell^+ \in ind(\rho,p)} \sum_{\ell^+ \in ind(\rho,p)} apply_X(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}_{\ell^+})$ that is the result. ◀

▶ **Definition 151** ($\mathcal{A}$-brief-partition)**.** Let $\mathcal{A}$ be a PSIOA, X be PCA, Let ρ be a schedule of X. Let $\alpha \in frags(X)$. Let $p = (\tilde{\alpha}^{s^1}, \tilde{\alpha}^{s^2}, ...\tilde{\alpha}^{s^m}$ be the $\mathcal{A}$-partition of α A $\mathcal{A}$-brief-partition of α is a sequence $\alpha^1, \alpha^2, ..., \alpha^n$. s. t.

- $\alpha = \alpha^{1\frown}\alpha^{2\frown}...\alpha^n$
- $\forall i \in [1, n], \exists!(\ell_i, h_i) \in [1, m]^2, \alpha^i = \tilde{\alpha}^{s^{\ell_i}}{}^\frown ... \tilde{\alpha}^{s^{h_i}}$
- $\forall i \in [1, n-1], \ell_{i+1} = h_i + 1$

▶ **Lemma 152.** *Let $\mathcal{A}$ be a PSIOA, X be PCA, Let ρ be a schedule of X. Let $\alpha^{12} = \alpha^{1\frown}\alpha^2$ a non single state execution of X that matches ρ, where (α^1, α^2) is a $\mathcal{A}$-brief-partition of α^{12}. Let $\ell_2 = max(ind(\rho, p))$ where p is any partition of ρ.*

$$apply_X(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}^{12}) = \sum_{0 \prec \ell_1 \prec \ell_2} apply(\rho|_{\ell_1})(\underline{\alpha}^1_{\ell_1,\rho}) \cdot apply(_{(\ell_1+1)}|\rho)(\underline{\alpha}^2)$$

Proof. $apply_X(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}) = \sum_{\alpha^{1\prime\frown}\alpha^{2\prime} \in \underline{\alpha}^{12}} apply(\rho|_{\ell_2})(\alpha^{1\prime\frown}\alpha^{2\prime}) =$

$$\sum_{\alpha^{1\prime} \in \underline{\alpha}^1} \sum_{\alpha^{2\prime} \in \underline{\alpha}^2} apply(\rho|_{\ell_2})(\alpha^{1\prime \frown} \alpha^{2\prime}) =$$
$$\sum_{\alpha^{1\prime} \in \underline{\alpha}^1} \sum_{\alpha^{2\prime} \in \underline{\alpha}^2} apply(\rho|_{\ell_1(\alpha^{1\prime})})(\alpha^{1\prime}) \cdot apply(_{(\ell_1(\alpha^{1\prime})+1)}|\rho|_{\ell_2})(\alpha^{2\prime}) =$$
$$\sum_{0 \prec \ell_1 \prec \ell_2} \sum_{\alpha^{1\prime} \in \underline{\alpha}^1_{(\ell_1,\rho)}} \sum_{\alpha^{2\prime} \in \underline{\alpha}^2} apply(\rho|_{\ell_1})(\alpha^{1\prime}) \cdot apply(_{(\ell_1+1)}|\rho|_{\ell_2})(\alpha^{2\prime}) =$$
$$\sum_{0 \prec \ell_1 \prec \ell_2} \sum_{\alpha^{1\prime} \in \underline{\alpha}^1_{(\ell_1,\rho)}} apply(\rho|_{\ell_1})(\alpha^{1\prime}) \cdot \sum_{\alpha^{2\prime} \in \underline{\alpha}^2} apply(_{(\ell_1+1)}|\rho|_{\ell_2})(\alpha^{2\prime}) =$$
$$\sum_{0 \prec \ell_1 \prec \ell_2} apply(\rho|_{\ell_1})(\underline{\alpha}^1_{\ell_1,\rho}) \cdot apply(_{(\ell_1+1)}|\rho|_{\ell_2})(\underline{\alpha}^2)$$ ◀

▶ **Lemma 153** (Total probability law with all the possible cuts). *Let $\mathcal{A}$ be a PSIOA, X be PCA, Let ρ be a schedule of X. Let $\alpha^{(1,n)} = \alpha^{1\frown}\alpha^{2\frown}...\alpha^{(n-1)\frown}\alpha^n$ an execution of X that matches ρ, where $(\alpha^1, \alpha^2, ..., \alpha^n)$ is a $\mathcal{A}$-brief-partition of $\alpha^{(1,n)}$. Let $\ell_n = max(ind(\rho, p))$ where p is any partition of ρ.*

$$apply_X(\delta_{fstate(\alpha^{(1,n)})}, \rho)(\underline{\alpha}^{(1,n)})) =$$

$$\sum_{\substack{\ell_1,\ell_2,...,\ell_{n-1} \\ 0 \prec \ell^1 \prec \ell^2 \prec ... \prec \ell^{n-1} \prec \ell_n}} \Gamma(\alpha^1, \ell^1, \rho)[\Pi_{i \in [2:n-1]} \Gamma'(\alpha^i, \ell^{i-1}, \ell^i, \rho)]\Gamma''(\alpha^n, \ell^{n-1}, \rho)$$

with

- $\Gamma(\alpha^1, \ell^1, \rho) = apply_X(\delta_{fstate(\alpha^1)}, \rho|_{\ell_1})(\underline{\alpha}^1_{\ell^1,\rho})$,
- $\Gamma'(\alpha^i, \ell^{i-1}, \ell^i, \rho) = apply_X(\delta_{fstate(\alpha^i)}, _{(\ell_{i-1}+1)} |\rho|_{\ell_i})(\underline{\alpha}^i_{(\ell^{i-1},\ell^i,\rho)}))$ *and*
- $\Gamma''(\alpha^n, \ell^{n-1}, \rho) = apply_X(\delta_{fstate(\alpha^n)}, _{(\ell_{n-1}+1)} |\rho)(\underline{\alpha}^n)$

Proof. By induction on the size of the brief-partition. Basis is true by the previous lemma. We assume the predicate true for $n-1$ and we show this implies the predicate is true for integer n.

Let $(\alpha^1, ..., \alpha^{n-1}, \alpha^n)$ be a $\mathcal{A}$-brief-partition of α^{1n}.

We note $\alpha^{(1,n)} = \alpha^{1\frown}\alpha^{(2,n)}$. $(\alpha^2, ..., \alpha^n)$ is clearly a $\mathcal{A}$-brief-partition of $\alpha^{(2,n)}$ of size $n-1$, $(\alpha^1, \alpha^{(2,n)})$ is a $\mathcal{A}$-brief-partition of α^{1n} with size 2 lower or equal than n.

Now $apply_X(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}^{(1,n)}) =$

$$\sum_{\substack{\ell^1 \\ 0 \prec \ell^1 \prec \ell^n}} apply_X(\delta_{fstate(\alpha^1)}, \rho|_{\ell^1})(\underline{\alpha}^1_{(\ell^1,\rho)}) \cdot apply_X(\delta_{fstate(\alpha^{2n})}, (_{\ell^1+1}|\rho))(\underline{\alpha}^{(2,n)}))$$

by induction hypothesis.

We note $\rho' =_{\ell^1+1} |\rho$, and reuse the induction hypothesis, which gives

$$apply_X(\delta_{fstate(\alpha^{(2,n)})}, \rho')(\underline{\alpha}^{(2,n)})) =$$

$$\sum_{\substack{\ell_2,...,\ell_{n-1} \\ 0 \prec \ell^2 \prec ... \prec \ell^{n-1} \prec \ell_n}} \Gamma(\alpha^2, \ell^2, \rho')[\Pi_{i \in [3:n-1]} \Gamma'(\alpha^i, \ell^{i-1}, \ell^i, \rho')]\Gamma''(\alpha^n, \ell^{n-1}, \rho')$$

$$\sum_{\substack{\ell_2,...,\ell_{n-1} \\ 0 \prec \ell^2 \prec ... \prec \ell^{n-1} \prec \ell_n}} \Gamma'(\alpha^2, \ell^1, \ell^2, \rho)[\Pi_{i \in [3:n-1]} \Gamma'(\alpha^i, \ell^{i-1}, \ell^i, \rho)]\Gamma''(\alpha^n, \ell^{n-1}, \rho)$$

We compose the last two results to obtain

$apply_X(\delta_{fstate(\alpha^{(1,n)})}, \rho)(\underline{\alpha}^{(1,n)})) =$

$$\sum_{\substack{\ell_1, \ell_2, \dots, \ell_{n-1} \\ 0 \prec \ell^1 \prec \ell^2 \prec \dots \prec \ell^{n-1} \prec \ell_n}} \Gamma(\alpha^1, \ell^1, \rho)[\Pi_{i \in [2:n-1]} \Gamma'(\alpha^i, \ell^{i-1}, \ell^i, \rho)]\Gamma''(\alpha^n, \ell^{n-1}, \rho)$$

, which is the desired result.

◀

▶ **Lemma 154.** *Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $\mathcal{E}$ be an environment of both $\mathcal{A}$ and $\mathcal{B}$. Let ρ and ρ' be $\mathcal{AB}$-environment-corresponding schedule with p the $\mathcal{A}$-partition of ρ and p' the $\mathcal{B}$-partition of ρ' s. t. for every $(k, q) \in \mathbb{N}^2$, for every $\ell = (2k, q) \in ind(\rho, p) \cap ind(\rho', p')$, $(\rho|_\ell) S^s_{(\mathcal{A},\mathcal{B},\mathcal{E})} (\rho'|_\ell)$.*

Then

- *for every $\tilde{\ell} = (2\tilde{k}, \tilde{q}) \in ind(\rho, p) \cap ind(\rho', p')$ with $(\tilde{k}, \tilde{q}) \in \mathbb{N}^2$:* $\sum_{\ell \in ind(\rho,p)}^{\ell \preceq \tilde{\ell}} apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}})(\underline{\alpha}_{\ell, \rho|_{\tilde{\ell}}}) = \sum_{\ell \in ind(\rho',p')}^{\ell \preceq \tilde{\ell}} apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B, \bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}})(\underline{\pi}_{\ell, \rho'|_{\tilde{\ell}}})$ *and*
- *for every $\tilde{\ell} = (2\tilde{k}, \tilde{q}) \in ind(\rho, p) \cap ind(\rho', p')$ with $(\tilde{k}, \tilde{q}) \in \mathbb{N} \times \mathbb{N}^*$, for every $\ell = (2k, q) \in ind(\rho, p) \cap ind(\rho', p')$ with $(k, q) \in \mathbb{N} \times \mathbb{N}^*$ and $\ell \preceq \tilde{\ell}$:* $apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}})(\underline{\alpha}_{\ell, \rho|_{\tilde{\ell}}}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B, \bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}})(\underline{\pi}_{\ell, \rho'|_{\tilde{\ell}}})$

Proof. By induction on k.

We deal with two induction hypothesis for every $\tilde{\ell}^* = (2\tilde{k}^*, \tilde{q}^*) \in ind(\rho, p) \cap ind(\rho', p')$ with $(\tilde{k}^*, \tilde{q}^*) \in \mathbb{N} \times \mathbb{N}$.

$IH^1(\tilde{\ell}^*)$: for every $\tilde{\ell} = (2\tilde{k}, \tilde{q}) \in ind(\rho, p) \cap ind(\rho', p')$ with $(\tilde{k}, \tilde{q}) \in \mathbb{N} \times \mathbb{N}$ and $\tilde{\ell} \preceq \tilde{\ell}^*$

$\sum_{\ell \in ind(\rho,p)}^{\ell \preceq \tilde{\ell}} apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}})(\underline{\alpha}_{\ell, \rho|_{\tilde{\ell}}}) = \sum_{\ell \in ind(\rho',p')}^{\ell \preceq \tilde{\ell}} apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B, \bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}})(\underline{\pi}_{\ell, \rho'|_{\tilde{\ell}}})$ and

$IH^2(\tilde{\ell}^*)$: for every $\tilde{\ell} = (2\tilde{k}, \tilde{q}) \in ind(\rho, p) \cap ind(\rho', p')$ with $(\tilde{k}, \tilde{q}) \in \mathbb{N} \times \mathbb{N}$ $\forall (k, q) \in \mathbb{N} \times \mathbb{N}^*, \forall \ell = (2k, q) \in ind(\rho, p) \cap ind(\rho', p')$, s. t. $\ell \preceq \tilde{\ell} \preceq \tilde{\ell}^*$

$apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}})(\underline{\alpha}_{\ell, \rho|_{\tilde{\ell}}}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B, \bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}})(\underline{\pi}_{\ell, \rho'|_{\tilde{\ell}}})$

Basis: Let $\alpha' \in supp(apply(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, \lambda)) \cap \underline{\alpha}$, then $\{\alpha'\} = \underline{\alpha}_{(0,\rho)} = \{(\bar{q}_A, \bar{q}_\mathcal{E})\}$. Similarly if $\pi' \in supp(apply(\delta_{(\bar{q}_B, \bar{q}_\mathcal{E})}, \lambda)) \cap \underline{\pi}$, then $\{\pi'\} = \underline{\pi}_{(0,\rho)} = \{(\bar{q}_B, \bar{q}_\mathcal{E})\}$.

Thus $apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, _0|\rho)(\underline{\alpha}_{0,\rho}) = apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, _0|\rho)(\underline{\alpha})$ and $apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B, \bar{q}_\mathcal{E})}, _0|\rho')(\underline{\pi}_{0,\rho'}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B, \bar{q}_\mathcal{E})}, \underline{\pi}_{0,\rho'})(\underline{\pi})$.

Hence $apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, _0|\rho)(\underline{\alpha}_{0,\rho}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B, \bar{q}_\mathcal{E})}, _0|\rho')(\underline{\pi}_{0,\rho'})$, which means that $IH^1(0)$ and $IH^2(0)$ are true.

Induction:

Let $\tilde{\ell} = (2\tilde{k}, \tilde{q}), \tilde{\ell}' = (2\tilde{k}', \tilde{q}') \in ind(p, \rho) \cap ind(p', \rho')$ with $\tilde{k}, \tilde{q}, \tilde{k}', \tilde{q}' \in \mathbb{N}$ and $\tilde{\ell} \prec \tilde{\ell}'$.

We note that

$\sum_{\ell \in ind(\rho,p)}^{\ell \preceq \tilde{\ell}'} apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}'})(\underline{\alpha}_{\ell, \rho|_{\tilde{\ell}'}}) =$

$apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}'})(\underline{\alpha}) - \sum_{\ell \in ind(\rho,p)}^{\ell \preceq \tilde{\ell}} apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A, \bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}})(\underline{\alpha}_{\ell, \rho|_{\tilde{\ell}}})$ (*)

and

$\sum_{\ell \in ind(\rho',p')}^{\ell \preceq \tilde{\ell}'} apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}'})(\underline{\pi}_{\ell,\rho'|_{\tilde{\ell}'}}) =$

$apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}'})(\underline{\pi}) - \sum_{\ell \in ind(\rho,p)}^{\ell \preceq \tilde{\ell}} apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}})(\underline{\pi}_{\ell,\rho'|_{\tilde{\ell}}})$ (**)

We assume $IH^1(\ell)$ and $IH^2(\ell)$ to be true for every $\ell = (2k, q)$ with $k, q \in \mathbb{N}$ s. t. $\ell \in ind(\rho, p) \cap ind(\rho', p')$ and $\ell \preceq \tilde{\ell}$.

We need to consider two cases:

Case 1: $\tilde{\ell} + 1 = (2\tilde{k}, \tilde{q} + 1)$: Case 2: $\tilde{\ell} + 1 \neq (2\tilde{k}, \tilde{q} + 1)$

Case 1: We evaluate (*) and (**) with $\tilde{\ell}' = \tilde{\ell} + 1$

$apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A,\bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}+1})(\underline{\alpha}_{\tilde{\ell}+1,\rho|_{\tilde{\ell}+1}}) = apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A,\bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}+1})(\underline{\alpha}) - \sum_{\ell \in ind(\rho,p)}^{\ell \preceq \tilde{\ell}} apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A,\bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}})(\underline{\alpha}_{\ell,\rho|_{\tilde{\ell}}})$
and similarly

$apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}+1})(\underline{\pi}_{\tilde{\ell}+1,\rho'|_{\tilde{\ell}+1}}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}+1})(\underline{\pi}) - \sum_{\ell \in ind(\rho',p')}^{\ell \preceq \tilde{\ell}} apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}})(\underline{\pi}_{\ell,\rho'}$

Thus, we apply $IH^1(\tilde{\ell})$ and the equality $apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A,\bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}+1})(\underline{\alpha}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}+1})(\underline{\pi})$ by assumption to obtain both $IH^1(\tilde{\ell}')$ and $IH^2(\tilde{\ell}')$.

Case 2: We evaluate (*) and (**) with $\tilde{\ell}' = (2(k+1), 0)$,

We apply $IH^1(\tilde{\ell})$ and the equality $apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A,\bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}'})(\underline{\alpha}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}'})(\underline{\pi})$ by assumption to obtain $IH^1(\tilde{\ell}')$.

Then, we can evaluate (*) and (**) with $\tilde{\ell}' = (2(k+1), 0)$ and $\tilde{\ell}'' = (2(k+1), 1)$, apply $IH^1(\tilde{\ell}')$ and the equality $apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A,\bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}''})(\underline{\alpha}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}''})(\underline{\pi})$ by assumption to obtain both $IH^1(\tilde{\ell}'')$ and $IH^2(\tilde{\ell}'')$

By induction, we obtain the desired result:

- for every $\tilde{\ell} = (2\tilde{k}, \tilde{q}) \in ind(\rho, p) \cap ind(\rho', p')$ with $(\tilde{k}, \tilde{q}) \in \mathbb{N}^2$: $\sum_{\ell \in ind(\rho,p)}^{\ell \preceq \tilde{\ell}} apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A,\bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}})(\underline{\alpha}_{\ell,\rho|_{\tilde{\ell}}}) = \sum_{\ell \in ind(\rho',p')}^{\ell \preceq \tilde{\ell}} apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}})(\underline{\pi}_{\ell,\rho'|_{\tilde{\ell}}})$ and
- for every $\tilde{\ell} = (2\tilde{k}, \tilde{q}) \in ind(\rho, p) \cap ind(\rho', p')$ with $(\tilde{k}, \tilde{q}) \in \mathbb{N} \times \mathbb{N}^*$, for every $\ell = (2k, q) \in ind(\rho, p) \cap ind(\rho', p')$ with $(k, q) \in \mathbb{N} \times \mathbb{N}^*$ and $\ell \preceq \tilde{\ell}$: $apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A,\bar{q}_\mathcal{E})}, \rho|_{\tilde{\ell}})(\underline{\alpha}_{\ell,\rho|_{\tilde{\ell}}}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, \rho'|_{\tilde{\ell}})(\underline{\pi}_{\ell,\rho'|_{\tilde{\ell}}})$

◀

▶ **Lemma 155** (subdivision in sub-classes of probability distribution correspondence)**.** *Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $\mathcal{E}$ be an environment of both $\mathcal{A}$ and $\mathcal{B}$. Let ρ and ρ' be $\mathcal{AB}$-environment-corresponding schedule with p the $\mathcal{A}$-partition of ρ and p' the $\mathcal{B}$-partition of ρ' s. t. for every $(k, q), (k', q') \in \mathbb{N}^2$, for every $\ell = (2k, q), \ell' = (2k', q') \in ind(\rho, p) \cap ind(\rho', p')$, $({}_\ell|\rho|_{\ell'}) S^s_{(\mathcal{A},\mathcal{B},\mathcal{E})} ({}_\ell|\rho'|_{\ell'})$.*

Then

for every $\ell = (2k, q), \ell' = (2k', q') \in ind(\rho, p) \cap ind(\rho', p')$ with $(k, q), (k', q') \in \mathbb{N} \times \mathbb{N}^*$ and $\ell \preceq \tilde{\ell}$:

$apply_{\mathcal{A}||\mathcal{E}}(\delta_{(\bar{q}_A,\bar{q}_\mathcal{E})}, {}_\ell|\rho|_{\ell'})(\underline{\alpha}_{(\ell,\ell',{}_\ell|\rho|_{\ell'})}) = apply_{\mathcal{B}||\mathcal{E}}(\delta_{(\bar{q}_B,\bar{q}_\mathcal{E})}, {}_\ell|\rho'|_{\ell'})(\underline{\pi}_{(\ell,\ell',{}_\ell|\rho'|_{\ell'})})$

Proof. We apply the previous lemma with $\tilde{\rho} = {}_\ell|\rho$ and $\tilde{\rho}' = {}_\ell|\rho'$. ◀

8.3 Implementation

▶ **Definition 156** (Strong implementation)**.** Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. We say that $\mathcal{A}$ *strongly implements* $\mathcal{B}$ iff for every environment $\mathcal{E}$ of both $\mathcal{A}$ and $\mathcal{B}$, for every schedule ρ , it exists an $\mathcal{AB}$-environment-corresponding schedule ρ', s. t. for every $\ell = (2k, q)$: $(\rho|_\ell) S^s_{(\mathcal{A},\mathcal{B},\mathcal{E})} (\rho'|_\ell)$.

The impementation says that for each schedule dedicated to $\mathcal{A}||\mathcal{E}$ there is a counterpart dedicated to $\mathcal{B}||\mathcal{E}$ so that each corresponding equivalence classes have the same probability measure. Hence there is no statistical experimentation for an environment to distinguish $\mathcal{A}$ from $\mathcal{B}$. Also the definition requires that the relationship stays true for every prefix cut at an environment's task at an arbitrary (even) index.

▶ **Definition 157** (Tenacious implementation)**.** Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. We say that $\mathcal{A}$ *tenaciously implements* $\mathcal{B}$, noted $\mathcal{A} \leq^{ten} \mathcal{B}$, iff for every schedule ρ, it exists a $\mathcal{AB}$-environment-corresponding schedule ρ' s. t. for every environment $\mathcal{E}$ of both $\mathcal{A}$ and $\mathcal{B}$, for every $\ell = (2k, q)$, $\ell' = (2k', q') \in ind(\rho, p) \cap ind(\rho', p')$, $({}_\ell|\rho|_{\ell'}) S^s_{(\mathcal{A},\mathcal{B},\mathcal{E})} ({}_\ell|\rho'|_{\ell'})$

The tenacious implementation is a variant of strong implementation where the relationship stays true for every suffix cut at an environment's task at an arbitrary index. Moreover, the choice of the corresponding schedule does not depend of the environment. Hence, to stay indistinguishable by the environment $\mathcal{A}$ and $\mathcal{B}$ do not need to change their "strategy", the same pair of corresponding schedule is enough to prevent the distinction of $\mathcal{A}$ and $\mathcal{B}$ by any environment with any "strategy".

8.4 Implementation Monotonicity

▶ **Lemma 158** (Corresponding-environment relation is preserved in the upper level)**.** *Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $X_\mathcal{A}$, $X_\mathcal{B}$ be PCA corresponding w.r.t. $\mathcal{A}$, $\mathcal{B}$. Let ρ, ρ' be $\mathcal{AB}$-environment-corresponding schedules. ρ, ρ' are also $X_\mathcal{A} X_\mathcal{B}$-environment-corresponding schedules.*

Proof. We note $Y_\mathcal{A} = X_\mathcal{A} \setminus \mathcal{A}$ and $Y_\mathcal{B} = X_\mathcal{B} \setminus \mathcal{B}$. It is sufficient to partition each sub-schedule $\rho^{2k}_\mathcal{E}$ into tasks with id in $UA(Y_\mathcal{A}) = UB(Y_\mathcal{B})$ and tasks with id not in $UA(Y_\mathcal{A}) = UB(Y_\mathcal{B})$. If $\rho^{2k}_\mathcal{E}$ begins (resp. ends) by a sequence of tasks with ids in $UA(Y_\mathcal{A})$, we can combine them with tasks of $\rho^{2k-1}_\mathcal{A}$ (resp. $\rho^{2k+1}_\mathcal{A}$) to obtain a sequence of tasks in $UA(X_\mathcal{A})$. The other tasks are not in $UA(X_\mathcal{A})$. If $\rho^{2k}_\mathcal{E}$ begins (resp. ends) by a sequence of tasks with ids in $UA(Y_\mathcal{B})$, we can combine them with tasks of $\rho^{2k-1}_\mathcal{B}$ (resp. $\rho^{2k+1}_\mathcal{B}$) to obtain a sequence of tasks in $UA(X_\mathcal{B})$. The other tasks are not in $UA(X_\mathcal{B})$. ◀

▶ **Lemma 159** (S^s monotonocity wrt creation and destruction)**.** *Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $X_\mathcal{A}$, $X_\mathcal{B}$ be PCA corresponding w.r.t. $\mathcal{A}$, $\mathcal{B}$. Let ρ, ρ' be $\mathcal{AB}$-environment-corresponding schedules s. t. for every environment $\mathcal{E}''$ of both $\mathcal{A}$ and $\mathcal{B}$, for every $\ell = (2k, q)$, $\ell' = (2k', q') \in ind(\rho, p) \cap ind(\rho', p')$, $({}_\ell|\rho|_{\ell'}) S^s_{(\mathcal{A},\mathcal{B},\mathcal{E}'')} ({}_\ell|\rho'|_{\ell'})$.*

Then for every environment $\mathcal{E}$ of both $X_\mathcal{A}$ and $X_\mathcal{B}$, for every $\ell = (2k, q)$, $\ell' = (2k', q') \in ind(\rho, p) \cap ind(\rho', p')$, $({}_\ell|\rho|_{\ell'}) S^s_{(X_\mathcal{A},X_\mathcal{B},\mathcal{E})} ({}_\ell|\rho'|_{\ell'})$.

Proof. By induction.

We assume this is true up to $\ell^+ \leq 2k$ and we show this is also true for $2k + 1$ and $2k + 2$.

We have two cases: The first case is $\mathcal{A}$ never created, where the results is true because of homorphism without creation. Thus we investigate only the second case $\mathcal{A}$ is created at least once :

We note $\ell_4 = (2k_4, q_4) = max(ind(\rho, p)) = max(ind(\rho', p'))$ (potentially $q_4 = 0$), $\alpha = \alpha^{13} \frown \alpha^4$ (resp. $\pi = \pi^{13} \frown \pi^4$)where α^{13} (resp π^{13}) ends on $\mathcal{A}$ (resp. $\mathcal{B}$) creation.

Because of lemma 153, we have both

$apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}) =$

$\sum_{\ell_3 \in ind(\rho,p)}^{\ell_3 \prec \ell_4} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_3})(\underline{\alpha}^{13}_{(\ell_3,\rho)}) \cdot apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^4)}, (_{\ell_3+1}|\rho))(\underline{\alpha}^4)$ and

$apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho')(\underline{\pi}) =$

$\sum_{\ell^3 \in ind(\rho',p')}^{\ell_3 \prec \ell_4} apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_3})(\underline{\pi}^{13}_{(\ell_3,\rho)}) \cdot apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^4)}, (_{\ell_3+1}|\rho'))(\underline{\pi}^4)$.

Since α^{13} (resp. π^{13}) ends on $\mathcal{A}$ ($\mathcal{B}$) creation, $\underline{\alpha}^{13}_{\ell_3,\rho} \neq \emptyset$ only if $\ell_3 = (2k_3, q_3)$ with $(k_3, q_3) \in \mathbb{N} \times \mathbb{N}^*$.

We already have for every $\ell_3 = (2k_3, q_3)$,

$apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^4)}, (_{\ell_3+1}|\rho))(\underline{\alpha}^4) = apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^4)}, (_{\ell_3+1}|\rho'))(\underline{\pi}^4)$ for every $\ell_3 = (2k_3, q_3) \in ind(\rho, p)$ by the theorem 139 of preservation of probabilistic distribution without creation.

Indeed, we note $Y'^3_\mathcal{A}$ (resp. $Y'^3_\mathcal{B}$) the $\mathcal{A}$-twin (resp. $\mathcal{B}$-twin) PCA of $Y_\mathcal{A} = X_\mathcal{A} \setminus \mathcal{A}$ (resp. $Y_\mathcal{B} = X_\mathcal{B} \setminus \mathcal{B}$) where the initial state is $\mu_s^\mathcal{A}(lstate(\alpha^{13}) \restriction X_\mathcal{A})$ (resp. $\mu_s^\mathcal{B}(lstate(\pi^{13}) \restriction X_\mathcal{B})$), we note $\mathcal{E}^3_\prime$ the PCA equal to $\mathcal{E}$ except that its initial state is $(lstate(\pi^{13}) \restriction \mathcal{E}$ and we note $\mathcal{E}^{3\prime\prime}_\mathcal{A} = Y'^3_\mathcal{A}||psioa(\mathcal{E}^{3\prime})$, $\mathcal{E}^{3\prime\prime}_\mathcal{B} = Y'^3_\mathcal{B}||psioa(\mathcal{E}^{3\prime})$ and $\mathcal{E}^{3\prime\prime} = \mathcal{E}^{3\prime\prime}_\mathcal{A}$ or $\mathcal{E}^{3\prime\prime} = \mathcal{E}^{3\prime\prime}_\mathcal{B}$ arbitrarily.

The premises of the lemma give

$apply_{\mathcal{A}||\mathcal{E}^{3\prime\prime}}(\delta_{fstate(\gamma_e(\mu_e(\alpha^4))}, (_{\ell_3+1}|\rho))(\underline{\gamma_e(\mu_e(\alpha^4)}) = apply_{\mathcal{B}||\mathcal{E}^{3\prime\prime}}(\delta_{fstate(\gamma_e(\mu_e(\pi^4))}, (_{\ell_3+1}|\rho'))(\underline{\gamma_e(\mu_e(\pi^4)})$ for every $\ell_3 = (2k_3, q_3) \in ind(\rho, p)$. And the theorem 139 of preservation of probabilistic distribution without creation gives for every $\ell_3 = (2k_3, q_3) \in ind(\rho, p)$:

$apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^4)}, (_{\ell_3+1}|\rho))(\underline{\alpha}^4) = apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^4)}, (_{\ell_3+1}|\rho'))(\underline{\pi}^4)$ for every $\ell_3 = (2k_3, q_3) \in ind(\rho, p)$.

Then we consider several cases:

Case 1: $\mathcal{A}$ (resp. $\mathcal{B}$) not destroyed (originally absent) in α^{13} (resp. π^{13})

In this case $\underline{\alpha}^{13} = \{\alpha^{13}\}$ and $\underline{\pi}^{13} = \{\pi^{13}\}$ with $\alpha^{13} \simeq \pi^{13}$. Since $\mathcal{A}$ and $\mathcal{B}$ are absent, all the tasks of odd index are ignored, hence

$apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_3})(\underline{\alpha}^{13}_{(\ell_3,\rho)}) = apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho''|_{\ell_3})(\underline{\alpha}^{13}_{(\ell_3,\rho'')})$ and

$apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_3})(\underline{\pi}^{13}_{(\ell_3,\rho')}) = apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho''|_{\ell_3})(\underline{\pi}^{13}_{(\ell_3,\rho'')})$ with $\rho'' = \rho^0_\mathcal{E}\rho^2_\mathcal{E}...\rho^{2*\lfloor card(p)/2 \rfloor}$.

Since $\alpha^{13} \simeq \pi^{13}$, $apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho''|_{\ell_3})(\underline{\alpha}^{13}_{(\ell_3,\rho'')}) = apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho''|_{\ell_3})(\underline{\pi}^{13}_{(\ell_3,\rho'')})$ for every $\ell_3 = (2k_3, q_3) \in ind(\rho, p)$ (Moreover it exists at most one $\ell_3^* = (2k_3^*, q_3^*)$, s. t. $apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho''|_{\ell_3^*})(\underline{\alpha}^{13}_{(\ell_3^*,\rho'')}) = apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho''|_{\ell_3^*})(\alpha^{13}) \neq 0$).

Hence either $apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}) = apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho')(\underline{\pi}) = 0$ or $apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}) = apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_3^*})(\underline{\alpha}^{13}_{(\ell_3^*,\rho)}) \cdot apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^4)}, (_{\ell_3^*+1}|\rho))(\underline{\alpha}^4)$ and

$apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho')(\underline{\pi}) = apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_3^*})(\underline{\pi}^{13}_{(\ell_3^*,\rho)}) \cdot apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^4)}, (_{\ell_3^*+1}|\rho'))(\underline{\pi}^4)$.

In both cases $apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}) = apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho')(\underline{\pi})$ which terminates case 1.

Case 2: $\mathcal{A}$ (resp. $\mathcal{B}$) destroyed.

We note $\alpha^{13} = \alpha^{12} \frown \alpha^3$ (resp. $\pi^{13} = \pi^{12} \frown \pi^3$) where α^{12} (resp. π^{12}) ends on $\mathcal{A}$ (resp. $\mathcal{B}$) destruction.

Here again, since α^{13} (resp. π^{13}) ends on $\mathcal{A}$ (resp. $\mathcal{B}$) creation , if $\underline{\alpha}^{13}_{\ell_3,\rho} \neq \emptyset$ (resp. $\underline{\pi}^{13}_{\ell_3,\rho'} \neq \emptyset$), then $\ell_3 = (2k_3, q_3)$ with $(k_3, q_3) \in \mathbb{N} \times \mathbb{N}^*$.

Let $\ell_3 = (2k_3, q_3)$ with $(k_3, q_3) \in \mathbb{N} \times \mathbb{N}^*$. Because of lemma 153, we have

$apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_3})(\underline{\alpha}^{13}_{(\ell_3,\rho)}) =$

$\sum_{\ell_2 \in ind(\rho,p)}^{\ell_2 \prec \ell_3} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}_{(\ell_2,\rho)}) \cdot apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^3)}, {}_{(\ell_2+1}|\rho|_{\ell_3}))(\underline{\alpha}^3_{(\ell_2+1,\ell_3,\rho)})$
and

$apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_3})(\underline{\pi}^3_{(\ell_3,\rho')}) =$

$\sum_{\ell_2 \in ind(\rho',p')}^{\ell_2 \prec \ell_3} apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_2})(\underline{\pi}^{12}_{(\ell_2,\rho')}) \cdot apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, {}_{(\ell_2+1}|\rho'|_{\ell_3}))(\underline{\pi}^3_{(\ell_2+1,\ell_3,\rho)})$.

Since α^{12} (resp. π^{12}) ends on $\mathcal{A}$ (resp. $\mathcal{B}$) destruction, all task of $\mathcal{A}$ (resp. $\mathcal{B}$) are ignored after the destruction. Thus, if $\underline{\alpha}^{12}_{\ell_2,\rho} \neq \emptyset$ (resp. $\underline{\pi}^{12}_{\ell_2,\rho'} \neq \emptyset$), then $\ell_2 = (2k_2 + 1, q_3)$ with $(k_2, q_2) \in \mathbb{N} \times \mathbb{N}^*$.

For the same reason, for every $\ell_2 = (2k_2 + 1, q_2) \in \mathbb{N} \times \mathbb{N}^*$, $\ell_2^+ = (2k_2 + 2, 0)$, we have

- $(\underline{\alpha}^3_{(\ell_2,\ell_3,\rho)}) = (\underline{\alpha}^3_{(\ell_2^+,\ell_3,\rho)})$,
- $(\underline{\pi}^3_{(\ell_2,\ell_3,\rho)}) = (\underline{\pi}^3_{(\ell_2^+,\ell_3,\rho)})$

Thus we obtain

$apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_3})(\underline{\alpha}^{13}_{(\ell_3,\rho)}) = \sum_{k_2}^{k_2<k_3} \sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho,p)}^{\ell_2 \preceq \ell_2^+} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}_{(\ell_2,\rho)}) \cdot$
$apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^3)}, {}_{(\ell_2^++1}|\rho|_{\ell_3}))(\underline{\alpha}^3_{((\ell_2^++1),\ell_3,\rho)}) = \sum_{k_2}^{k_2<k_3} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^3)}, {}_{(\ell_2^++1}|\rho|_{\ell_3}))(\underline{\alpha}^3_{((\ell_2^++1),\ell_3,\rho)}) \cdot$
$\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho,p)}^{\ell_2 \preceq \ell_2^+} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}_{(\ell_2,\rho)})$.

We obtain the symetric result for π^{13}, hence :

- $apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_3})(\underline{\alpha}^{13}_{(\ell_3,\rho)}) = \sum_{k_2}^{k_2<k_3} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^3)}, {}_{((2k_2+2,1)}|\rho|_{\ell_3}))(\underline{\alpha}^3_{((2k_2+2,1),\ell_3,\rho)}) \cdot$
$\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho,p)}^{\ell_2 \preceq \ell_2^+} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}_{(\ell_2,\rho)})$.
- $apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_3})(\underline{\pi}^{13}_{(\ell_3,\rho')}) = \sum_{k_2}^{k_2<k_3} apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^3)}, {}_{((2k_2+2,1)}|\rho'|_{\ell_3}))(\underline{\pi}^3_{((2k_2+2,1),\ell_3,\rho')}) \cdot$
$\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho',p')}^{\ell_2 \preceq \ell_2^+} apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_2})(\underline{\pi}^{12}_{(\ell_2,\rho')})$.

In this case $\underline{\alpha}^3 = \{\alpha^3\}$, $\underline{\pi}^3 = \{\pi^3\}$ and $\alpha^3 \simeq \pi^3$. Since $\mathcal{A}$ and $\mathcal{B}$ are absent in α^3 and π^3 respectively (excepting at the last state) all the tasks of odd index are ignored. Thus, for each $(2k_2 + 2, 1) \prec (2k_3, q_3)$,

$apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^3)}, {}_{((2k_2+2,1)}|\rho|_{\ell_3}))(\underline{\alpha}^3_{((2k_2+2,1),\ell_3,\rho)}) = apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^3)}, {}_{((2k_2+2,1)}|\rho'|_{\ell_3}))(\underline{\pi}^3_{((2k_2+2,1),\ell_3,\rho')})$.

So we still need to show that for every k_2 s. t. $(2k_2 + 2, 1) \prec (2k_3, q_3)$,

$$\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho,p)}^{\ell_2 \preceq \ell_2^+} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}_{(\ell_2,\rho)}) = \sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho',p')}^{\ell_2 \preceq \ell_2^+} apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_2})(\underline{\pi}^{12}_{(\ell_2,\rho')}) \quad (1)$$

Case 2a: $\mathcal{A}$ (resp. $\mathcal{B}$) created only once (in $lstate(\alpha^3)$ and in $lstate(\pi^3)$) (originally present).

In this case $\underline{\alpha}^{12} = \underline{\pi}^{12}$ and the result is immediate by the theorem 139 of preservation of probabilistic distribution without creation.

Indeed, we note $Y_{\mathcal{A}} = X_{\mathcal{A}} \setminus \mathcal{A}$ and$Y_{\mathcal{B}} = X_{\mathcal{B}} \setminus \mathcal{B}$ and we note $\mathcal{E}''_{\mathcal{A}} = Y_{\mathcal{A}}||psioa(\mathcal{E})$, $\mathcal{E}''_{\mathcal{B}} = Y_{\mathcal{B}}||psioa(\mathcal{E})$ and $\mathcal{E}'' = \mathcal{E}''_{\mathcal{A}}$ or $\mathcal{E}'' = \mathcal{E}''_{\mathcal{B}}$ arbitrarily.

The premises of the lemma give

$apply_{\mathcal{A}||\mathcal{E}''}(\delta_{fstate(\gamma_e^{\mathcal{A}}(\mu_e^{\mathcal{A}}(\alpha))}, \rho|_{\ell_2})(\underline{\gamma_e^{\mathcal{A}}(\mu_e^{\mathcal{A}}(\alpha^{12}))}) = apply_{\mathcal{B}||\mathcal{E}''}(\delta_{fstate(\gamma_e^{\mathcal{B}}(\mu_e^{\mathcal{B}}(\pi))}, \rho'|_{\ell_2}))(\underline{\gamma_e^{\mathcal{B}}(\mu_e^{\mathcal{B}}(\pi^{12}))})$
for every $\ell_2 = (2k_2, q_2) \in ind(\rho, p)$ with no creation of $\mathcal{A}$ and $\mathcal{B}$ in α^{12} and π^{12} respectively. Thus we can apply the theorem 139 of preservation of probabilistic distribution to obtain $apply_{X_{\mathcal{A}}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}_{(\ell_2,\rho)}) = apply_{X_{\mathcal{B}}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_2})(\underline{\pi}^{12}_{(\ell_2,\rho')})$ for every $\ell_2 = (2k_2, q_2) \in ind(\rho, p)$, which allows to verify the equation 1, which terminates the induction and the proof for case 2a.

Case 2b: $\mathcal{A}$ (resp. $\mathcal{B}$) created twice. We note $\alpha^{12} = \alpha^1 {}^\frown \alpha^2$ (resp. $\pi^{12} = \pi^1 {}^\frown \pi^2$) where α^1 (resp. π^1) ends on $\mathcal{A}$ (resp. $\mathcal{B}$) creation. For every k_2, we note $\ell_2^-(k_2) = (2k_2+1, 1)$ and $\ell_2^+(k_2) = (2k_2+2, 0)$. We fix k_2. Let ℓ_2, s. t. $\ell_2^-(k_2) \preceq \ell_2 \preceq \ell_2^+(k_2)$.

Because of lemma 153, we have:

$apply_{X_{\mathcal{A}}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}_{(\ell_2,\rho)}) =$

$\sum_{\ell_1 \in ind(\rho,p)}^{\ell_1 \prec \ell_2} apply_{X_{\mathcal{A}}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_1})(\underline{\alpha}^{1}_{(\ell_1,\rho)}) \cdot apply_{X_{\mathcal{A}}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2}))(\underline{\alpha}^{2}_{(\ell_1+1,\ell_2,\rho)})$.

and

$apply_{X_{\mathcal{B}}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_2})(\underline{\pi}^{12}_{(\ell_2,\rho')}) =$

$\sum_{\ell_1 \in ind(\rho',p')}^{\ell_1 \prec \ell_2} apply_{X_{\mathcal{B}}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_1})(\underline{\pi}^{1}_{(\ell_1,\rho')}) \cdot apply_{X_{\mathcal{B}}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2}))(\underline{\pi}^{2}_{(\ell_1+1,\ell_2,\rho')})$.

Hence,

- $\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho,p)}^{\ell_2 \preceq \ell_2^+} apply_{X_{\mathcal{A}}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}_{(\ell_2,\rho)}) =$
$\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho,p)}^{\ell_2 \preceq \ell_2^+} \sum_{\ell_1 \in ind(\rho,p)}^{\ell_1 \prec \ell_2} apply_{X_{\mathcal{A}}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_1})(\underline{\alpha}^{1}_{(\ell_1,\rho)}) \cdot$
$apply_{X_{\mathcal{A}}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2}))(\underline{\alpha}^{2}_{(\ell_1+1,\ell_2,\rho)})$
- $\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho',p')}^{\ell_2 \preceq \ell_2^+} apply_{X_{\mathcal{B}}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_2})(\underline{\pi}^{12}_{(\ell_2,\rho')}) =$
$\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho',p')}^{\ell_2 \preceq \ell_2^+} \sum_{\ell_1 \in ind(\rho',p')}^{\ell_1 \prec \ell_2} apply_{X_{\mathcal{B}}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_1})(\underline{\pi}^{1}_{(\ell_1,\rho')}) \cdot$
$apply_{X_{\mathcal{B}}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2}))(\underline{\pi}^{2}_{(\ell_1+1,\ell_2,\rho')})$

Since α^1 (resp. π^1) ends on $\mathcal{A}$ (resp. $\mathcal{B}$) creation, it can match $\rho|_{\ell_1}$ only if $\ell_1 = (2k_1, q_1)$. Thus $apply(\delta_{fstate(\alpha)}, \rho|_{\ell_1})(\underline{\alpha}^{1}_{(\ell_1,\rho)}) \neq 0$ and $\ell_1 \prec \ell_2$ implies $\ell_1 \prec \ell_2^-$ and $apply(\delta_{fstate(\pi)}, \rho'|_{\ell_1})(\underline{\pi}^{1}_{(\ell_1,\rho')}) \neq 0$ and $\ell_1 \prec \ell_2$ implies $\ell_1 \prec \ell_2^-$.

Thus,

- $\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho,p)}^{\ell_2 \preceq \ell_2^+} apply_{X_{\mathcal{A}}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}_{(\ell_2,\rho)}) =$
$\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho,p)}^{\ell_2 \preceq \ell_2^+} \sum_{\ell_1 \in ind(\rho,p)}^{\ell_1 \prec \ell_2^-} apply_{X_{\mathcal{A}}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_1})(\underline{\alpha}^{1}_{(\ell_1,\rho)}) \cdot$
$apply_{X_{\mathcal{A}}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2}))(\underline{\alpha}^{2}_{(\ell_1+1,\ell_2,\rho)})$
- $\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho',p')}^{\ell_2 \preceq \ell_2^+} apply_{X_{\mathcal{B}}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_2})(\underline{\pi}^{12}_{(\ell_2,\rho')}) =$
$\sum_{\ell_2=(2k_2+1,q_2)\in ind(\rho',p')}^{\ell_2 \preceq \ell_2^+} \sum_{\ell_1 \in ind(\rho',p')}^{\ell_1 \prec \ell_2^-} apply_{X_{\mathcal{B}}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_1})(\underline{\pi}^{1}_{(\ell_1,\rho')}) \cdot$
$apply_{X_{\mathcal{B}}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2}))(\underline{\pi}^{2}_{(\ell_1+1,\ell_2,\rho')})$

which gives:

- $\sum_{\ell_2 \in ind(\rho,p)}^{\ell_2^- \prec \ell_2 \preceq \ell_2^+} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_2})(\underline{\alpha}^{12}_{(\ell_2,\rho)}) =$
$\sum_{\ell_1 \in ind(\rho,p)}^{\ell_1 \prec \ell_2^-} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_1})(\underline{\alpha}^{1}_{(\ell_1,\rho)})$.
$\sum_{\ell_2 \in ind(\rho,p)}^{\ell_2^- \preceq \ell_2 \preceq \ell_2^+} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2}))(\underline{\alpha}^{2}_{(\ell_1+1,\ell_2,\rho)})$
- $\sum_{\ell_2 \in ind(\rho',p')}^{\ell_2^- \preceq \ell_2 \preceq \ell_2^+} apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_2})(\underline{\pi}^{12}_{(\ell_2,\rho')}) =$
$\sum_{\ell_1 \in ind(\rho',p')}^{\ell_1 \prec \ell_2^-} apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_1})(\underline{\pi}^{1}_{(\ell_1,\rho')})$.
$\sum_{\ell_2 \in ind(\rho',p')}^{\ell_2^- \preceq \ell_2 \preceq \ell_2^+} apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2}))(\underline{\pi}^{2}_{(\ell_1+1,\ell_2,\rho')})$

By induction hypothesis, $apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho|_{\ell_1})(\underline{\alpha}^1_{(\ell_1,\rho)}) = apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho'|_{\ell_1})(\underline{\pi}^1_{(\ell_1,\rho')})$ for every $\ell_1 \prec (2k_2+1,1) \prec (2k_2+2,0) \prec (2k_3,q_3) \prec (2k_4,q_4)$.

So we need to show that for every $\ell_1 \prec \ell_2^-$

$$\sum_{\ell_2 \in ind(\rho,p)}^{\ell_2^- \preceq \ell_2 \preceq \ell_2^+} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2}))(\underline{\alpha}^{2}_{(\ell_1+1,\ell_2,\rho)}) =$$

$$\sum_{\ell_2 \in ind(\rho',p')}^{\ell_2^- \preceq \ell_2 \preceq \ell_2^+} apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2}))(\underline{\pi}^{2}_{(\ell_1+1,\ell_2,\rho')})$$

that is, $apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2^+}))(\underline{\alpha}^2) - apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2^- -1}))(\underline{\alpha}^2) =$
$apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2^+}))(\underline{\pi}^2) - apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2^-} - 1))(\underline{\pi}^2)$.

To do so, we will show that:

$$\begin{aligned} apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2^+}))(\underline{\alpha}^2) &= apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2^+}))(\underline{\pi}^2) \\ apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2^- -1}))(\underline{\alpha}^2) &= apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2^- -1}))(\underline{\pi}^2) \end{aligned} \quad (2)$$

We note $Y_\mathcal{A} = X_\mathcal{A} \setminus \mathcal{A}$ and $Y_\mathcal{B} = X_\mathcal{B} \setminus \mathcal{B}$. We note $Y'_\mathcal{A}$ (resp. $Y'_\mathcal{B}$) the $\mathcal{A}$-twin (resp. $\mathcal{B}$-twin) of $Y_\mathcal{A}$ (resp. $Y_\mathcal{B}$) with $\mu_s^\mathcal{A}(lstate(\alpha^1) \upharpoonright X_\mathcal{A})$ (resp. $\mu_s^\mathcal{B}(lstate(\pi^1) \upharpoonright X_\mathcal{B})$) as initial state. We note $\mathcal{E}'$ the PCA equal to $\mathcal{E}$ excepting that its initial state is $lstate(\alpha^1) \upharpoonright \mathcal{E}$.

We note $\mathcal{E}''_\mathcal{A} = Y'_\mathcal{A}||psioa(\mathcal{E}')$, $\mathcal{E}''_\mathcal{B} = Y'_\mathcal{B}||psioa(\mathcal{E}')$ and $\mathcal{E}'' = \mathcal{E}''_\mathcal{A}$ or $\mathcal{E}'' = \mathcal{E}''_\mathcal{B}$ arbitrarily.

Since $\ell_1 = (2k_1, q_1)$, $\ell_2^- - 1 = (2k_2, 0)$, $\ell_2^+ = (2k_2+1, 0)$, we have for every $\mathcal{E}''$, $apply_{\mathcal{A}||\mathcal{E}''}(\delta_{fstate(\gamma_e(\mu_e(\alpha^2)))}, ({}_{(\ell_1+1}|\rho|_{\ell_2^+}))(\underline{\gamma_e(\mu_e(\alpha^2))}) = apply_{\mathcal{B}||\mathcal{E}''}(\delta_{fstate(\gamma_e(\mu_e(\pi^2)))}, ({}_{(\ell_1+1}|\rho'|_{\ell_2^+}))(\underline{\gamma_e(\mu_e(\pi^2))})$ and $apply_{\mathcal{A}||\mathcal{E}''}(\delta_{fstate(\gamma_e(\mu_e(\alpha^2)))}, ({}_{(\ell_1+1}|\rho|_{\ell_2^- -1}))(\underline{\gamma_e(\mu_e(\alpha^2))}) = apply_{\mathcal{B}||\mathcal{E}''}(\delta_{fstate(\gamma_e(\mu_e(\pi^2)))}, ({}_{(\ell_1+1}|\rho'|_{\ell_2^-} - 1))(\underline{\gamma_e(\mu_e(\pi^2))})$.

Moreover, since α^2 (resp. π^2) does not create $\mathcal{A}$ (resp. $\mathcal{B}$) we can apply the theorem 139 of preservation of probabilistic distribution without creation to show 2.

Hence $apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2^+}))(\underline{\alpha}^2) - apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha^2)}, ({}_{(\ell_1+1}|\rho|_{\ell_2^- -1}))(\underline{\alpha}^2) =$
$apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2^+}))(\underline{\pi}^2) - apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi^2)}, ({}_{(\ell_1+1}|\rho'|_{\ell_2^-} - 1))(\underline{\pi}^2)$.

This implies that $apply_{X_\mathcal{A}||\mathcal{E}}(\delta_{fstate(\alpha)}, \rho)(\underline{\alpha}) = apply_{X_\mathcal{B}||\mathcal{E}}(\delta_{fstate(\pi)}, \rho')(\underline{\pi})$ in very case, which ends the induction and the proof.

◀

▶ **Theorem 160** (Implementation monotonicity wrt creation/destruction)**.** *Let $\mathcal{A}$, $\mathcal{B}$ be PSIOA. Let $X_\mathcal{A}$, $X_\mathcal{B}$ be PCA corresponding w.r.t. $\mathcal{A}$, $\mathcal{B}$.*

If $\mathcal{A}$ tenaciously implements $\mathcal{B}$ ($\mathcal{A} \leq^{ten} \mathcal{B}$) then $X_{\mathcal{A}}$ tenaciously implements $X_{\mathcal{B}}$ ($X_{\mathcal{A}} \leq^{ten} X_{\mathcal{B}}$).

Proof. Let ρ be a schedule, Since $\mathcal{A} \leq^{ten} \mathcal{B}$ it exists a schedule ρ' $\mathcal{AB}$-environment-corresponding with ρ s. t. for every $\mathcal{E}''$ environment of both $\mathcal{A}$ and $\mathcal{B}$, for every $\ell = (2k, q)$, $\ell' = (2k', q') \in ind(\rho, p) \cap ind(\rho', p')$, $(_{\ell}|\rho|_{\ell'}) S^s_{(\mathcal{A},\mathcal{B},\mathcal{E}'')} (_{\ell}|\rho'|_{\ell'})$.

Because of previous lemma 159 for every environment $\mathcal{E}$ of both $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$, for every $\ell = (2k, q)$, $\ell' = (2k', q') \in ind(\rho, p) \cap ind(\rho', p')$, (*) $(_{\ell}|\rho|_{\ell'}) S^s_{(X_{\mathcal{A}},X_{\mathcal{B}},\mathcal{E})} (_{\ell}|\rho'|_{\ell'})$, where p is the $\mathcal{A}$-partitition of ρ and p' is the $\mathcal{B}$-partitition of ρ'

Moreover ρ and ρ' are also $X_{\mathcal{A}}X_{\mathcal{B}}$-environment-corresponding because of lemma 158. Since the relation (*) is true for for every $\ell = (2k, q)$, $\ell' = (2k', q') \in ind(\rho, p) \cap ind(\rho', p')$, it is a fortiori true for every $\ell = (2k, q)$, $\ell' = (2k', q') \in ind(\rho, \tilde{p}) \cap ind(\rho', \tilde{p}')$ where $\tilde{p}$ is the $X_{\mathcal{A}}$-partitition of ρ and p' is the $X_{\mathcal{B}}$-partitition of ρ'.

Hence for every schedule ρ it exists a schedule ρ' $X_{\mathcal{A}}X_{\mathcal{B}}$-environment-corresponding with ρ s. t. for every $\mathcal{E}$ environment of both $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$, for every $\ell = (2k, q)$, $\ell' = (2k', q') \in ind(\rho, \tilde{p}) \cap ind(\rho', \tilde{p}')$, $(_{\ell}|\rho|_{\ell'}) S^s_{(X_{\mathcal{A}},X_{\mathcal{B}},\mathcal{E})} (_{\ell}|\rho'|_{\ell'})$ where $\tilde{p}$ is the $X_{\mathcal{A}}$-partitition of ρ and p' is the $X_{\mathcal{B}}$-partitition of ρ'.

This ends the proof.

◀

9 Conclusion

We formalised dynamic probabilistic setting. We exhibited the necessary and sufficient conditions to obtain implementation monotonicity w. r. t. Automata creation/destruction.

References

1. Paul C. Attie and Nancy A. Lynch. Dynamic input/output automata: A formal and compositional model for dynamic systems. 249:28–75.
2. Ran Canetti, Ling Cheung, Dilsun Kaynar, Moses Liskov, Nancy Lynch, Olivier Pereira, and Roberto Segala. Task-Structured Probabilistic {I/O} Automata. *Journal of Computer and System Sciences*, 94:63—-97, 2018.
3. Ran Canetti, Ling Cheung, Dilsun Kaynar, Nancy Lynch, and Olivier Pereira. Compositional security for task-PIOAs. *Proceedings - IEEE Computer Security Foundations Symposium*, pages 125–139, 2007.
4. Jing Chen and Silvio Micali. Algorand: A secure and efficient distributed ledger. *Theor. Comput. Sci.*, 777:155–183, 2019.
5. Maurice Herlihy. Blockchains and the future of distributed computing. In Elad Michael Schiller and Alexander A. Schwarzmann, editors, *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, page 155. ACM, 2017.
6. Nancy Lynch, Michael Merritt, William Weihl, and Alan Fekete. A theory of atomic transactions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 326 LNCS:41–71, 1988.
7. Martin L. Puterman. *Markov decision processes: discrete stochastic dynamic programming.* Wiley series in probability and mathematical statistics. John Wiley & Sons, 1 edition, 1994.

8 Alejandro Ranchal-Pedrosa and Vincent Gramoli. Platypus: Offchain protocol without synchrony. In Aris Gkoulalas-Divanis, Mirco Marchetti, and Dimiter R. Avresky, editors, *18th IEEE International Symposium on Network Computing and Applications, NCA 2019, Cambridge, MA, USA, September 26-28, 2019*, pages 1–8. IEEE, 2019.
9 Roberto Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusettes Institute of technology, 1995.

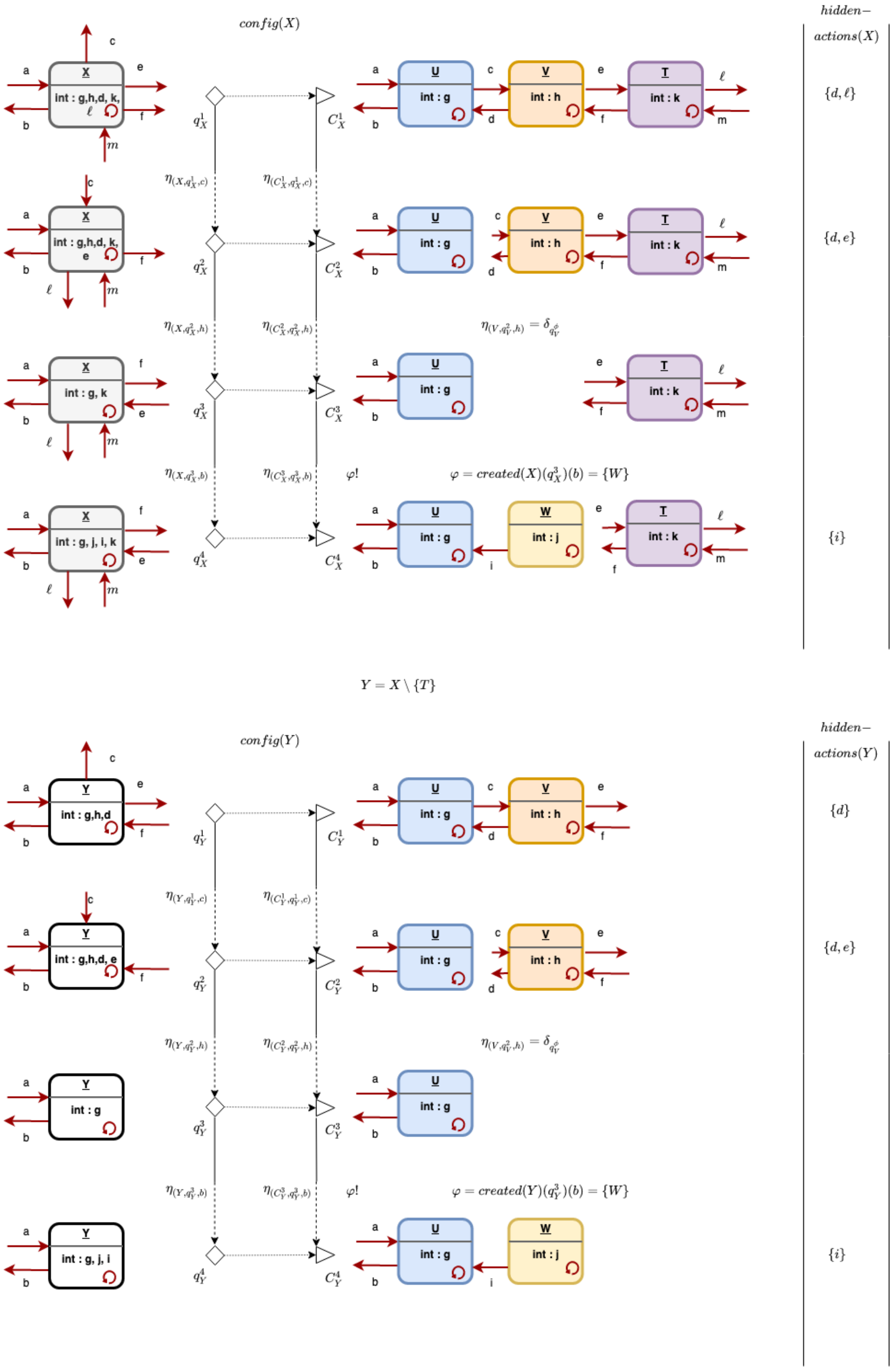
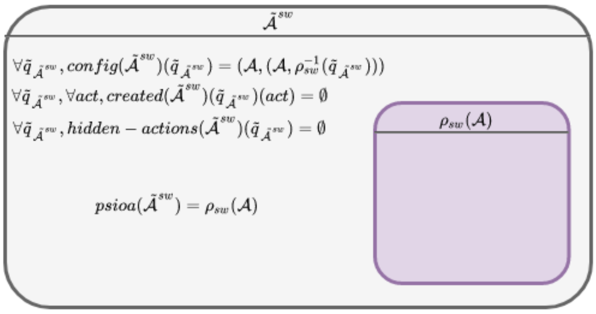


Figure 13 Projection on PCA



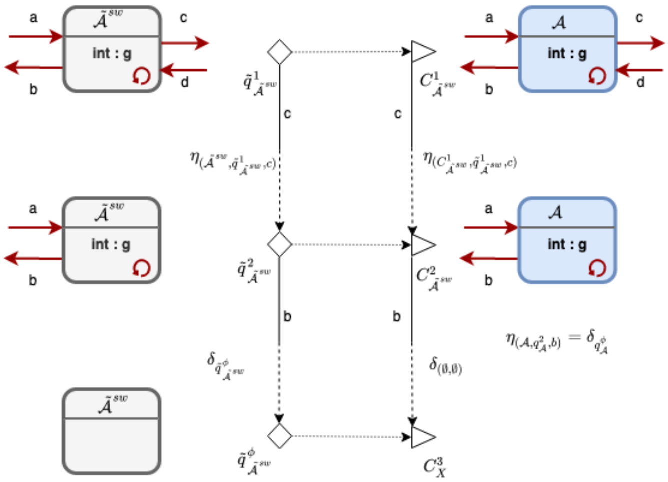


Figure 14 Simpleton wrapper

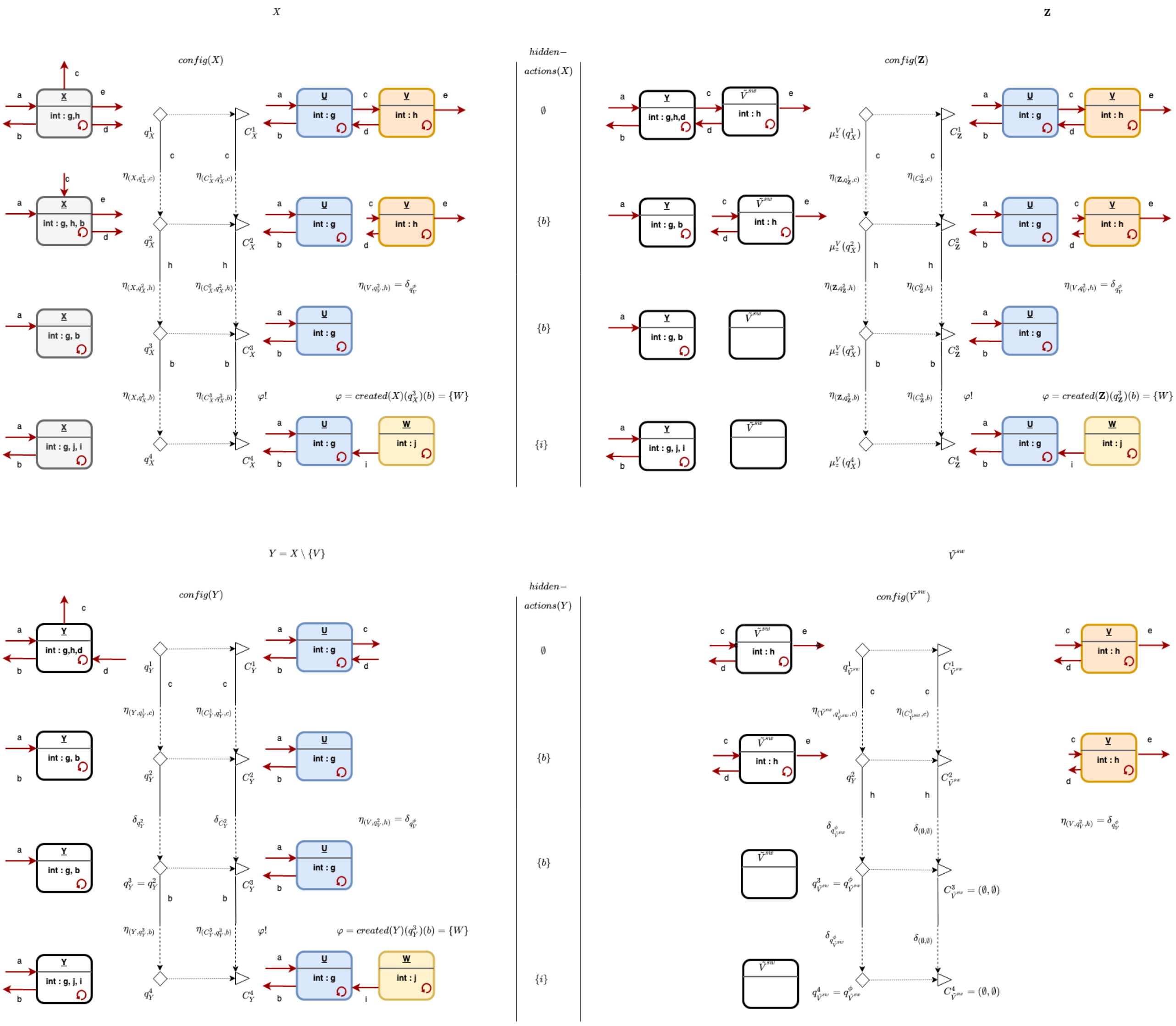


Figure 15 Reconstruction of a PCA

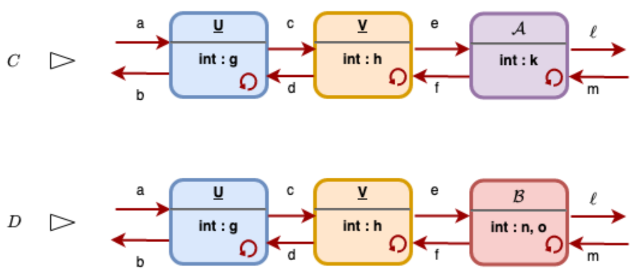


Figure 16 $\lhd_{\mathcal{AB}}$ corresponding-configuration

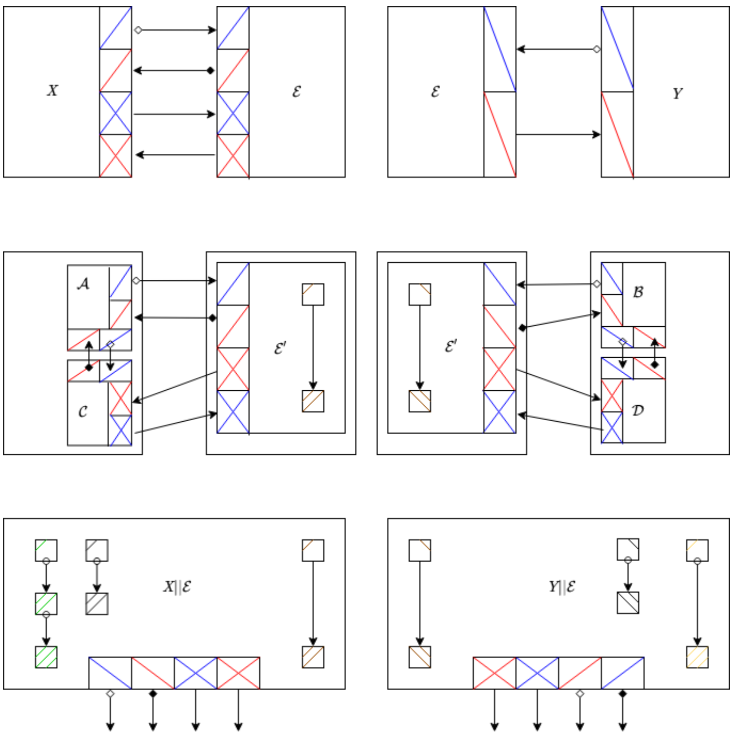


Figure 17 creation substitutivity for PCA. each blue or red box represents a set of actions. The one blue band ones are output actions for $\mathcal{A}$ or $\mathcal{B}$, the one red band ones are input actions for $\mathcal{A}$ or $\mathcal{B}$. The two blue bands ones are input actions for $\mathcal{E}'$ that do not come from $\mathcal{A}$ or $\mathcal{B}$, the two red bands ones are ouput actions for $\mathcal{E}'$ that do go into $\mathcal{A}$ or $\mathcal{B}$. The other squares represents internal states.