Dynamic Probabilistic Input Output Automata

PIERRE CIVIT, Sorbonne Université, CNRS, LIP6, France
MARIA POTOP-BUTUCARU, Sorbonne Université, CNRS, LIP6, France

We present *probabilistic dynamic I/O automata*, a framework to model dynamic probabilistic systems. Our work extends *dynamic I/O Automata* formalism of Attie & Lynch [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 construction uses a novel proof technique based on homomorphism that can be of independent interest. Our work lays down the foundations for extending *composable secure-emulation* of Canetti et al. [4] 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).

1 INTRODUCTION

Distributed computing area faces today important challenges coming from modern applications such as peer-to-peer networks, cooperative robotics, dynamic sensor networks, adhoc networks and more recently, cryptocurrencies and blockchains which have a tremendous impact in our society. These newly emerging fields of distributed systems are characterized by an extreme dynamism in terms of structure, content and load. Moreover, they have to offer strong guaranties over large scale networks which is usually impossible in deterministic settings. Therefore, most of these systems use probabilistic algorithms and randomized techniques in order to offer scalability features. However, the vulnerabilities of these systems may be exploited with the aim to provoke an unforeseen execution that diverges from the understanding or intuition of the developers. Therefore, formal validation and verification of these systems has to be realized before their industrial deployment.

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 by Segala in [8] with Markov decision processes [7]. In order to model randomized distributed systems Segala proposes *Probabilistic Input/Output Automata*. In this model each process in the system is an automaton with probabilistic transitions. The probabilistic protocol is the parallel composition of the automata modeling each participant.

The modelisation of dynamic behavior in distributed systems has been addressed by Attie & Lynch in [1] where they propose *Dynamic Input Output Automata* formalism. This formalism extends the *Input/Output Automata* 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 recent blockchains such as Algorand [5].

In order to respond to the need of formalisation in secure distributed systems, Canetti & al. proposed in [2] *task-structured probabilistic Input/Output automata* (TPIOA) specifically designed for the analysis of cryptographic protocols. Task-structured probabilistic Input/Output automata are Probabilistic Input/Output automata extended with tasks that are equivalence classes on the set of actions. The task-structure allows a generalisation of "off-line scheduling" where the non-determinism of the system is resolved in advance by a *task-scheduler*, i. e. a sequence of tasks chosen in advance that trigger the actions among the enabled ones. They define the parallel composition for this type of automata. Inspired by

the literature in security area they also define the notion of implementation for TPIOA. Informally, the implementation of a Task-structured probabilistic Input/Output automata should look "similar" to the specification whatever will be 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 of probabilistic dynamic systems such as peer-to-peer networks or blockchains systems where the set of participants dynamically changes.

Our contribution. In order to cope with dynamicity and probabilistic nature of modern distributed systems we propose an extension of the two formalisms introduced in [1] and [2]. Our extension uses 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. That is, if systems $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ differ only in that $X_{\mathcal{A}}$ dynamically creates and destroys automaton $\mathcal{A}$ instead of creating and destroying automaton $\mathcal{B}$ as $X_{\mathcal{B}}$ does, and if $\mathcal{A}$ implements $\mathcal{B}$ (in the sense they cannot be distinguished by any external observer), then $X_{\mathcal{A}}$ implements $X_{\mathcal{B}}$. This result enables a design and refinement methodology based solely on the notion of externally visible behavior and permits the refinement of components and subsystems in isolation from the rest of the system. In our construction we exhibit the need of considering only *creation-oblivious* schedulers in the implementation relation, i. e. a scheduler that, upon the (dynamic) creation of a sub-automaton $\mathcal{A}$, does not take into account the previous internal actions of $\mathcal{A}$ to output (randomly) a transition. Surprisingly, the task-schedulers introduced by Canetti & al. [2] are not creation-oblivious. Interestingly, an important contribution of the paper of independent interest is the proof technique we used in order to obtain our results. Differently from [1] and [2] which build their constructions mainly on induction techniques, we developed an elegant homomorphism based technique which aim to render the proofs modular. This proof technique can be easily adapted in order to further extend our framework with cryptography and time.

It should be noted that our work is an intermediate step before extending composable secure-emulation [4] to dynamic settings. This extension is necessary for formal verification of secure dynamic distributed systems (e.g. blockchain systems).

Paper organization. The paper is organized as follow. Section 3 is dedicated to a brief introduction of the notion of probabilistic measure and recalls notations used in defining Signature I/O automata of [1]. Section 4 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 5 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 the end of Section 6 and demonstrated in the remaining sections, up to Section 11). Section 12 explains why the off-line scheduler introduced by Canetti & al. [4] is not creation-oblivious and therefore cannot be used to obtain our key result.

2 WARM UP

In this section we describe the paper in the a very informal way. We aim to give some intuitions on the role of each section. The section 3 gives some preliminaries on probability and measure.

2.1 Probabilistic Signature Input/Output Automata (PSIOA)

The section 4 defines the notion of probabilistic signature Input/Output automata (PSIOA). A PSIOA $\mathcal{A}$ is an automaton that can move from one *state* to another through *actions*. The set of states of $\mathcal{A}$ is then denoted $states(\mathcal{A})$, while we note $start(\mathcal{A}) \in states(\mathcal{A})$ the unique start state of $\mathcal{A}$. At each state $q \in states(\mathcal{A})$ some actions can be triggered in its signature $sig(\mathcal{A})(q)$. Such an action leads to a new state with a certain probability. The measure of probability triggered by an action a in a state q is denoted $\eta_{(\mathcal{A},q,a)}$. The model aims to allow the composition (noted $\mathcal{A}_1||...||\mathcal{A}_n$) of several automata to capture the idea of an interaction between them. That is why a signature is composed by three categories of actions: the input actions, the output actions and the internal actions. In practice the input actions of an automaton potentially aim to be the ouput action of another automaton and vice-versa. Hence an automaton can influence another one through a shared action. The comportment of the entire system is formalised by the automaton issued from the compostion of the automata of the system.

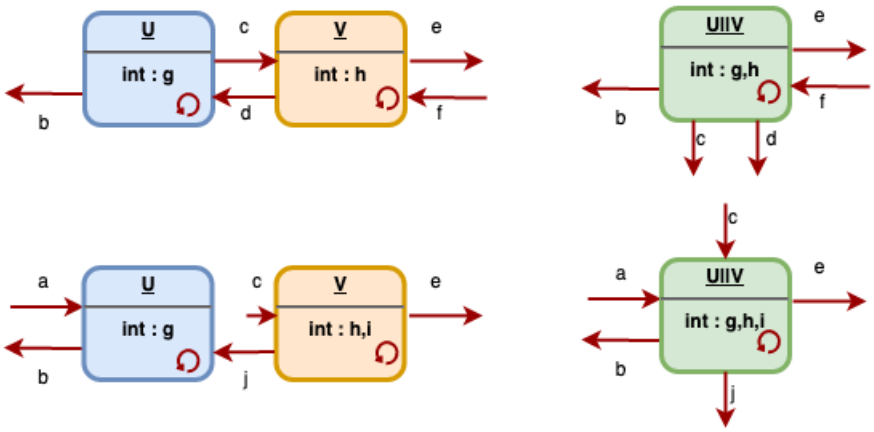


Fig. 1. A representation of two automata U and V. In the top line, we see the PSIOA U in a state q_U^1, s. t. $sig(U)(q_U^1) = (out(U)(q_U^1), in(U)(q_U^1), int(U)(q_U^1)) = (\{b,c\},\{d\},\{g\})$, the PSIOA V in a state q_V^1, s. t. $sig(V)(q_V^1) = (out(V)(q_V^1), in(V)(q_V^1), int(V)(q_V^1)) = (\{d,e\},\{c,f\},\{h\})$ and the result of their composition, the PSIOA $U||V$ in a state (q_U^1, q_V^1), s. t. $sig(U||V)((q_U^1, q_V^1)) = (out(U||V)((q_U^1, q_V^1)), in(U||V)((q_U^1, q_V^1)), int(U||V)((q_U^1, q_V^1) = (\{b,c,d,e\},\{f\},\{g,h\})$. In the second line we see the same PSIOA but in different states. We see the PSIOA U in a state q_U^2, s. t. $sig(U)(q_U^2) = (out(U)(q_U^2), in(U)(q_U^2), int(U)(q_U^2)) = (\{b\},\{a,j\},\{g\})$, the PSIOA V in a state q_V^2, s. t. $sig(V)(q_V^2) = (out(V)(q_V^2), in(V)(q_V^2), int(V)(q_V^2)) = (\{e,j\},\{c\},\{h,i\})$ and the result of their composition, the PSIOA $U||V$ in a state (q_U^2, q_V^2), s. t. $sig(U||V)((q_U^2, q_V^2)) = (out(U||V)((q_U^2, q_V^2)), in(U||V)((q_U^2, q_V^2)), int(U||V)((q_U^2, q_V^2) = (\{b,e,j\},\{a,c\},\{g,h,i\})$.

After this, we can speak about an execution of an automaton, which is an alternating sequence of states and actions. We can also speak about a trace of an automaton, which is the projection of an execution on the external actions uniquely. This allows us to speak about external behaviour of a system, that is, what can we observe from an outside point of view.

2.2 Scheduler

We remarked in the example of figure 2 that an inherent non-determinism has to be solved to be able to define a measure of probability on the executions, and so on the traces. This is the role of the scheduler which is a function

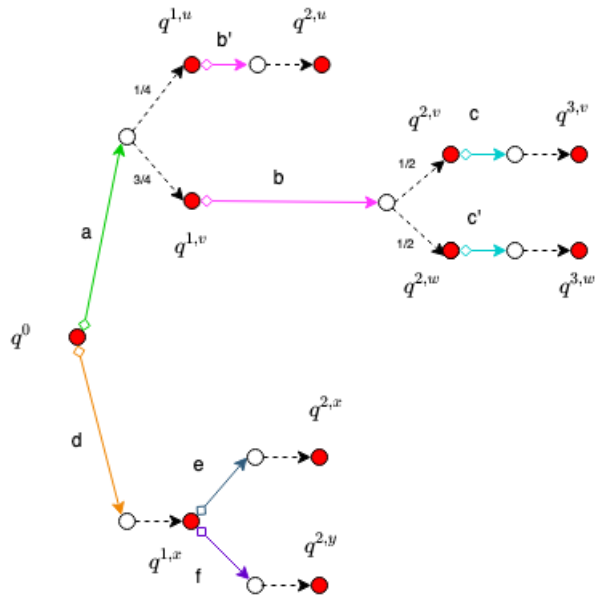


Fig. 2. The figure represents a tree of possible executions for a PSIOA $\mathcal{A}$. The red dots ($q^0, q^{1,\cdot}, q^{2,\cdot}, q^{3,\cdot}$) represents some states of the PSIOA. The PSIOA can move from on state to another through actions ($a, b, c, d, e, f, ...$) represented with colored solid arrows. Such an action act, triggered from a specif state q does not lead directly to another state q' but to a probabilistic distribution on states $\eta_{(\mathcal{A},q,act)}$ represented by a white dot and as many dashed black arrows as states in the support of $\eta_{(\mathcal{A},q,act)}$, i. e. the subset of states with non-zero probability for $\eta_{(\mathcal{A},q,act)}$. For example, the PSIOA $\mathcal{A}$ can be in state q^0, trigger the action a that leads him to $\eta_{(\mathcal{A},q,a)}$ and hence to $q^{1,u}$ with probability $1/4$ and to $q^{1,v}$ with probability $3/4$. Some executions are $q^0, a, q^{1,v}$; $q^0, a, q^{1,v}, b, q^{2,w}$; $q^0, a, q^{1,v}, b, q^{2,w}$; $q^0, a, q^{1,v}, b, q^{2,w}, c, q^{3,w}$; $q^0, a, q^{1,v}, b, q^{2,w}, c, q^{3,w}$; $q^0, a, q^{1,v}, b, q^{2,w}, c, q^{3,w}$. Let assume that b and b' are internal actions. The set of traces is then $\{a; a, c; a, c'; d; d, e; d, f\}$. Typically $a, c = trace(q^0, a, q^{1,v}, b, q^{2,w}, c, q^{3,v})$ We can already remark that a non-determinism is appearing since nothing states for the moment how two choose an action. How to know which action to take at state q^0 among a and d. This non-determinism will be solved by the *scheduler*, introduced later.

$\sigma : Frags^*(\mathcal{A}) \rightarrow SubDisc(dtrans(\mathcal{A}))$ that (consistently) maps an execution fragment to a discrete sub-probability distributions on set of discrete transitions of the concerned PSIOA $\mathcal{A}$. Loosely speaking, the scheduler σ decides (probabilistically) which transition to take after each finite execution fragment α. Since this decision is a discrete sub-probability measure, it may be the case that σ chooses to halt after α with non-zero probability: $1-\sigma(\alpha)(dtrans(\mathcal{A})) > 0$.

A scheduler σ and a probabilistic distribution μ on the set of finite execution fragments $Frags^*(\mathcal{A})$ generate a measure $\epsilon_{\sigma,\mu}$ on the sigma-field $\mathcal{F}_{Execs(\mathcal{A})}$ generated by cones of execution fragments (of the form $C_{\alpha^x} = \{\alpha^z \in Frags(\mathcal{A}) | \alpha^z = \alpha^x \frown \alpha^y | \alpha^y \in Frags(\mathcal{A})\}$), and so a measure on the measurable space $(G, \mathcal{F}_G)$ for any measurable function f from $(Execs(\mathcal{A}), \mathcal{F}_{Execs(\mathcal{A})})$ to $(G, \mathcal{F}_G)$. Hence, when a scheduler is made explicit, we can state the probability that a cone of execution is reached and that a property holds. By default, the probabilistic distribution μ on the set of finite execution fragments $Frags^*(\mathcal{A})$ is $\delta_{start(\mathcal{A})}$, i. e. the Dirac distribution that has a measure of 1 for the start state of the concerned automaton $\mathcal{A}$. We denote by $\epsilon_\sigma = \epsilon_{\sigma,\delta_{start(\mathcal{A})}} : Execs(\mathcal{A}) :\rightarrow [0,1]$ the execution distribution generated by the scheduler σ and $\delta_{start(\mathcal{A})}$.

2.3 Environment, external behavior, implementation

Now it is possible to define the crucial concept of implementation that captures the idea that an automaton $\mathcal{A}$ "mimics" another automaton $\mathcal{B}$. To do so, we define an environment $\mathcal{E}$ which takes on the role of a "distinguisher" for $\mathcal{A}$ and $\mathcal{B}$. In general, an environment of an automaton $\mathcal{A}$ is just an automaton compatible with $\mathcal{A}$ but some additional minor technical properties can be assumed. The set of environments of the automaton $\mathcal{A}$ is denoted $env(\mathcal{A})$. The information used by an environment to attempt a distinction between two automata $\mathcal{A}$ and $\mathcal{B}$ s. t. $\mathcal{E} \in env(\mathcal{A}) \cap env(\mathcal{B})$ is captured by a function $f_{(.,.)}$ that we call *insight function*. In the literature, we very often deal with $f_{(\mathcal{E},\mathcal{A})} = trace_{(\mathcal{E},\mathcal{A})}$ but in our case, the theorem of implementation monotonicity (stated later) holds for a slightly different function that we call $print_{(\mathcal{E},\mathcal{A})}$ which takes into account the entire execution of the environment itself. The philosophy of the approach remains nevertheless the same.

For any insight function $f_{(.,.)}$, we denote by $f\text{-}dist_{\mathcal{E},\mathcal{A}}(\sigma)$ the image measure of ϵ_σ under $f_{(\mathcal{E},\mathcal{A})}$. From here, this is classic to define the f-external behaviour of $\mathcal{A}$, denoted $ExtBeh^f_{\mathcal{A}} : \mathcal{E} \in env(\mathcal{A}) \mapsto \{f\text{-}dist_{\mathcal{A}||\mathcal{E}}(\sigma) | \sigma \in schedulers(\mathcal{E}||\mathcal{A})\}$. Such an object capture all the possible measures of probability on the external interaction of the concerned automaton $\mathcal{A}$ and an arbitrary environment $\mathcal{E}$. Finally we can say that $\mathcal{A}$ f-implements $\mathcal{B}$ if $\forall \mathcal{E} \in env(\mathcal{A}) \cap env(\mathcal{B}), ExtBeh^f_{\mathcal{A}}(\mathcal{E}) \subseteq ExtBeh^f_{\mathcal{B}}(\mathcal{E})$, i. e. for any "distinguisher" $\mathcal{E}$ for $\mathcal{A}$ and $\mathcal{B}$, for any possible distribution $f\text{-}dist_{(\mathcal{E},\mathcal{A})}(\sigma)$ of the interaction between $\mathcal{E}$ and $\mathcal{A}$ generated by a scheduler $\sigma \in schedulers(\mathcal{E}||\mathcal{A})$, there exists a scheduler $\sigma' \in schedulers(\mathcal{E}||\mathcal{B})$ s. t. the distribution $f\text{-}dist_{(\mathcal{E},\mathcal{B})}(\sigma')$ of the interaction between $\mathcal{E}$ and $\mathcal{B}$ generated by σ' is the same, i. e. for every external perception $\zeta \in range(f_{(\mathcal{E}||\mathcal{A})}) \cup range(f_{(\mathcal{E}||\mathcal{B})})$, $f\text{-}dist_{\mathcal{E}||\mathcal{A}}(\sigma)(\zeta) = f\text{-}dist_{\mathcal{E}||\mathcal{B}}(\sigma')(\zeta)$, noted $f\text{-}dist_{\mathcal{E}||\mathcal{A}}(\sigma) \equiv f\text{-}dist_{\mathcal{E}||\mathcal{B}}(\sigma')$. This a way to formalise that there is no way to distinguish $\mathcal{A}$ from $\mathcal{B}$. (see figure 3).

However, as already mentioned in [8], the correctness of an algorithm may be based on some specific assumptions on the scheduling policy that is used. Thus, in general, we are interested only in a subset of $schedulers(\mathcal{E}||\mathcal{A})$. A function that maps any automaton W to a subset of $schedulers(W)$ is called a *scheduler schema*. Among the most noteworthy examples are the fair schedulers, the off-line, a.k.a. oblivious schedulers, defined in opposition with the online-schedulers. So, we note $ExtBeh^{f,Sch}_{\mathcal{A}} : \mathcal{E} \in env(\mathcal{A}) \mapsto \{f\text{-}dist_{\mathcal{A}||\mathcal{E}}(\sigma) | \sigma \in Sch(\mathcal{E}||\mathcal{A})\}$ where Sch is a scheduler schema and we say that $\mathcal{A}$ f-implements $\mathcal{B}$ according to a scheduler schema Sch if $\forall \mathcal{E} \in env(\mathcal{A}) \cap env(\mathcal{B}), ExtBeh^{f,Sch}_{\mathcal{A}}(\mathcal{E}) \subseteq ExtBeh^{f,Sch}_{\mathcal{B}}(\mathcal{E})$. In the remaining, we will have a great interest for two certain classes of oblivious schedulers, i. e. i) the creation-oblivious scheduler (introduced later) and ii) the task-scheduler: an off-line scheduler already introduced in [2], which is relevant for cryptographic analysis. The previous notions can be adapted with a particular class of scheduler schema.

2.4 Probabilistic Configuration Automata (PCA)

The section 5 introduces the notion of probabilistic configuration automata (PCA). (see figure 4). A PCA is very closed to a PSIOA, but each state is mapped to a *configuration* $C = (\mathbf{A}, \mathbf{S})$ which is a pair constituted by a set $\mathbf{A}$ of PSIOA and the current states of each member of the set (with a mapping function $\mathbf{S} : \mathcal{A} \in \mathbf{A} \mapsto q_{\mathcal{A}} \in states(\mathcal{A})$). The idea is that the composition of the attached set can change during the execution of a PCA, which allows us to formalise the notion of dynamicity, that is the potential creation and potential destruction of a PSIOA in a dynamic system. Some particular precautions have to be taken to make it consistent.

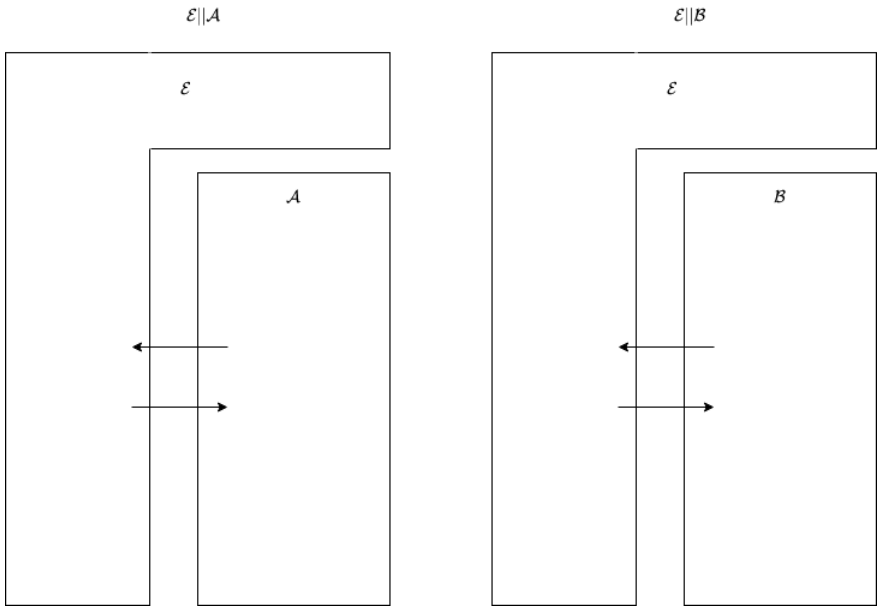


Fig. 3. An environment $\mathcal{E}$, which is nothing more than a PSIOA compatible with both $\mathcal{A}$ and $\mathcal{B}$, tries to distinguish $\mathcal{A}$ from $\mathcal{B}$. We say that $\mathcal{A}$ implements $\mathcal{B}$ if for every environment $\mathcal{E}$, $\mathcal{E}$ is unable to distinguish $\mathcal{A}$ from $\mathcal{B}$. To formalise it a little bit more, but not totally, we say that $\mathcal{A}$ implements $\mathcal{B}$ if for every environment $\mathcal{E} \in env(\mathcal{A}) \cap env(\mathcal{B})$, for every scheduler $\sigma \in schedulers(\mathcal{E}||\mathcal{A})$ applied to $\mathcal{E}||\mathcal{A}$ it exists a scheduler $\sigma' \in schedulers(\mathcal{E}||\mathcal{B})$ applied to $\mathcal{E}||\mathcal{B}$ s. t. $tdist_{\mathcal{E}||\mathcal{A}}(\sigma) \equiv tdist_{\mathcal{E}||\mathcal{B}}(\sigma')$, i. e. for every trace $\beta \in trace(\mathcal{E}||\mathcal{A}) \cup trace(\mathcal{E}||\mathcal{B})$, $tdist_{\mathcal{E}||\mathcal{A}}(\sigma)(\beta) = tdist_{\mathcal{E}||\mathcal{B}}(\sigma')(\beta)$.

2.5 Road to monotonicity

The rest of the paper is dedicated to the proof of implementation monotonicity. We show that, under certain technical conditions, automaton creation is monotonic with respect to external behavior inclusion, i. e. if a system X creates automaton $\mathcal{A}$ instead of (previously) creating automaton $\mathcal{B}$ and the external behaviors of $\mathcal{A}$ are a subset of the external behaviors of $\mathcal{B}$, then the set of external behaviors of the overall system is possibly reduced, but not increased. Such an external behavior inclusion result enable a design and refinement methodology based solely on the notion of externally visible behavior, and which is therefore independent of specific methods of establishing external behavior inclusion. It permits the refinement of components and subsystems in isolation from the entire system. To do so, we develop different mathematical tools.

2.5.1 Execution-matching. First, we define the notion of executions-matching (see figure 5) to capture the idea that two automata have the same comportment along some corresponding executions. Basically an execution-matching from a PSIOA $\mathcal{A}$ to a PSIOA $\mathcal{B}$ is a morphism $f^{ex} : Execs'_{\mathcal{A}} \rightarrow Execs(\mathcal{B})$ where $Execs'_{\mathcal{A}} \subseteq Execs(\mathcal{A})$. This morphism preserves some properties along the pair of matched executions: signature, transition, ... in such a way that for every pair $(\alpha, \alpha') \in Execs(\mathcal{A}) \times Execs(\mathcal{B})$ s. t. $\alpha' = f^{ex}(\alpha)$, $\epsilon_{\sigma}(\alpha) = \epsilon_{\sigma'}(\alpha')$ for every pair of scheduler (σ, σ') (so-called *alter ego*) that are "very similar" in the sense they take into account only the "structure" of the argument to return a sub-probability distribution, i. e. $\alpha' = f^{ex}(\alpha)$ implies $\sigma(\alpha) = \sigma'(\alpha')$. When the executions-matching is a bijection function from $Execs(\mathcal{A})$ to $Execs(\mathcal{B})$, we say $\mathcal{A}$ and $\mathcal{B}$ are semantically-equivalent (they differ only syntactically).

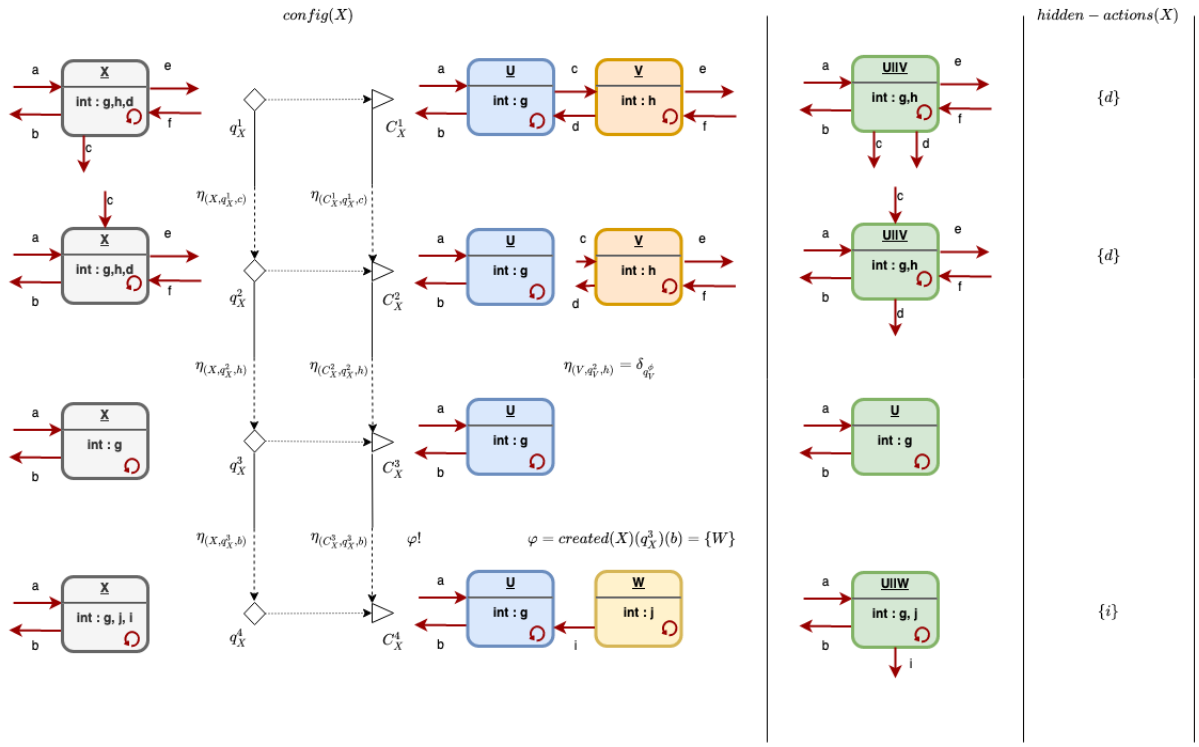


Fig. 4. The figure represents an execution fragment $(q_X^1, c, q_X^2, h, q_X^3, b, q_X^4)$ of a PCA X. In the left column, we see different states q_X^1, q_X^2, q_X^3 and q_X^4 of the PCA X, represented with white diamonds ($\diamond$). Each of these states q_X^i is mapped through the mapping $config(X)$ (represented with right dotted arrows) to a *configuration* C_X^i, represented with a white triangle ($\triangleright$). For example the state q_X^1 is mapped with the configuration $C_X^1 = (\mathbf{A}^1, \mathbf{S}^1)$ with $\mathbf{A}^1 = \{U, V\}$, $\mathbf{S}^1(U) = q_U^1$ and $\mathbf{S}^1(V) = q_V^1$. The signature of the PCA X at state q_X^i is the one of the composition of automata, in their current states in the attached configuration C_X^i, modulo some external actions $hidden\text{-}actions(X)(q_X^i)$ for C_X^i that are hidden and become internal for X. For example, the configuration C_X^1 has a signature $sig(C_X^1) = (out(C_X^1), in(C_X^1), int(C_X^1)) = (\{b, e, c, d\}, \{a, f\}, \{g, h\})$, while the signature of X at corresponding state is $sig(X)(q_X^1) = (out(X)(q_X^1), in(X)(q_X^1), int(X)(C_X^1)) = (\{b, e, c\}, \{a, f\}, \{g, h, d\})$ since the unique action $d \in hidden\text{-}actions(X)(q_X^1)$ is hidden and hence becomes an internal action. We can define discrete transitions for configurations in a similar way as what we do for PSIOA, but adding some tools (formally defined in section 5) to allow the creation and the destruction of automata. For example, the automaton V is destroyed during the step (q_X^2, h, q_X^3), while W is created during the step (q_X^3, b, q_X^4) which is made explicit by the fact that $created(X)(q_X^3)(b) = \{X\}$ where $created(X)$ is a mapping function defined for any PCA X. Some intuitive consistency rules have to be respected by pair of "corresponding transitions" $((q_X^i, act, \eta_{(X,q_X^i,act)}); (C_X^i, act, \eta_{(C_X^i,q_X^i,act)}))$ represented by pair of parallel downward arrows (one from two diamonds $\diamond$ and one from two triangles $\triangleright$) . For example, the probability $\eta_{(X,q_X^1,c)}(q_X^2)$ of reaching q_X^2 by triggering c from q_X^1 is equal to the probability $\eta_{(C_X^1,q_X^1,act)}(C_X^2)$ of reaching C_X^2 by triggering c from C_X^1. Moreover, a configuration transition has to respect some of other consistency rules with respect to the sub-automata that compose the configuration. Typically, the destruction of V in step (C_X^2, h, C_X^3) comes from the fact that the action h triggered from state q_V^2 of sub-automaton V lead to a probabilistic states distribution $\eta_{(V,q_V^2,h)}$ equal to $\delta_{q_V^\phi}$ which is a Dirac distribution for a special state q_V^ϕ with $sig(V)(q_V^\phi) = (\emptyset, \emptyset, \emptyset)$ that means V "has been destroyed".

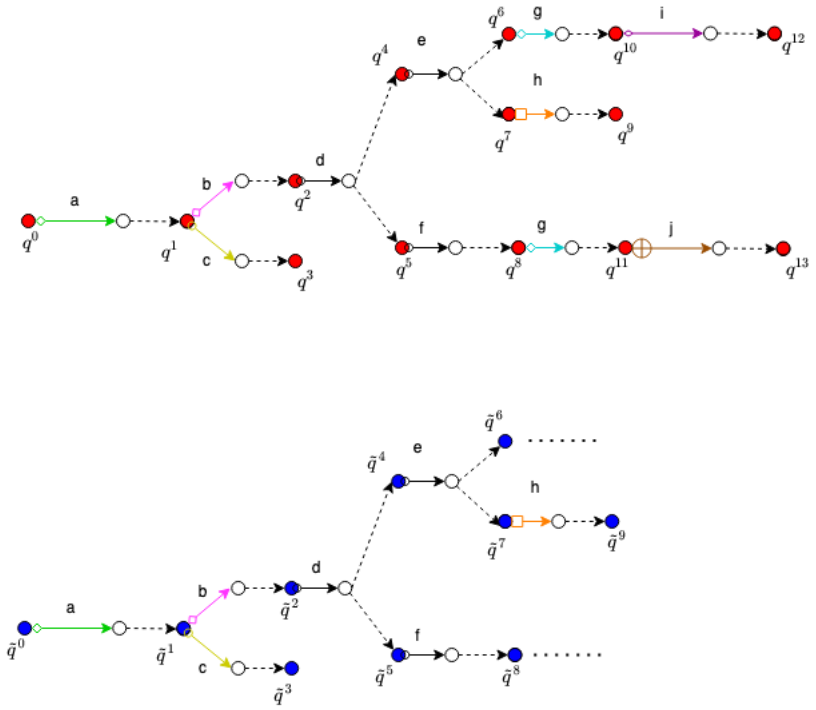


Fig. 5. The figure represents the respective executions tree of two automata $\mathcal{A}$ and $\mathcal{B}$ with some strong similarities. The states of $\mathcal{A}$ (resp. $\mathcal{B}$) are represented with red (resp. blue) dots. The actions are represented with solid arrows. An action leads to a discrete probability distribution on states η, represented with a white dot and dashed arrows reaching the different states of the support of η. In section 7, we define these strong similarities with what we call an executions-matching (f, f^{tr}, f^{ex}) where $f : states'_{\mathcal{A}} \to states(\mathcal{B})$, $f^{tr} : dtrans'_{\mathcal{A}} \to dtrans(\mathcal{B})$, $f^{ex} : Execs'_{\mathcal{A}} \to Execs(\mathcal{B})$ with $states'_{\mathcal{A}} \subseteq states(\mathcal{A})$, $dtrans'_{\mathcal{A}} \subseteq dtrans(\mathcal{A})$, $Execs'_{\mathcal{A}} \subseteq Execs(\mathcal{A})$. The mappings f, f^{tr} and f^{ex} preserves the important properties: signature for corresponding states, name of the action and measure of probability of corresponding states for corresponding transitions, etc. In the example the similarities exist until the states q^6, q^8 and q^9, hence we have $states'_{\mathcal{A}} = \{q^0, q^1, ..., q^9\} \subsetneq Q_{\mathcal{A}}$. The *states-matching* f is then defined s. t. $\forall k \in [1, 9], f(q^k) = \tilde{q}^k$. Thereafter, we define define $Act = \{a, b, c, d, e, f, h\}$ and f^{trans}, s. t. $\forall k \in [1, 9]$, $\forall act \in Act$, for every transition $(q^k, act, \eta_{(\mathcal{A},q^k,act)})$, $f^{trans}((q^k, act, \eta_{(\mathcal{A},q^k,act)})) = (\tilde{q}^k, act, \eta_{(\mathcal{B},q^k,act)})$. Each pair of mapped transition gives the same probability to pair of mapped states, e. g. $\eta_{(\mathcal{A},q^2,d)}(q^4) = \eta_{(\mathcal{B},\tilde{q}^2,d)}(\tilde{q}^4)$. Then we can define $Execs'_{\mathcal{A}} \subset Execs(\mathcal{A})$ the set of executions composed only with states in $Q'_{\mathcal{A}}$ and actions in Act. Finally $f^{ex} : \alpha = q^0 a^1 ... a^n q^n \in Execs'_{\mathcal{A}} \mapsto f(q^0) a^1 ... a^n f(q^n)$ is an execution-matching. The Point is that if two schedulers σ and σ' only look at the preserved properties to output a measure of probability on the actions to take, the attached measures of probability will be equal, i. e. $\epsilon_\sigma(\alpha) = \epsilon_{\sigma'}(\alpha')$

2.5.2 A PCA $X_{\mathcal{A}}$ deprived from a PSIOA $\mathcal{A}$. Second, we define in section 8 the notion of a PCA $X_{\mathcal{A}}$ deprived from a PSIOA $\mathcal{A}$ noted $(X_{\mathcal{A}} \setminus \{\mathcal{A}\})$. Such an automaton corresponds to the intuition of a similar automaton where $\mathcal{A}$ is systematically removed from the configuration of the original PCA (see figure 6 and 7).

2.5.3 Reconstruction: $(X_{\mathcal{A}} \setminus \{\mathcal{A}\})||\tilde{\mathcal{A}}^{sw}$. Thereafter we show in section 9 that under technical minor assumptions $X_{\mathcal{A}} \setminus \{\mathcal{A}\}$ and $\tilde{\mathcal{A}}^{sw}$ are composable where $\tilde{\mathcal{A}}^{sw}$ and $\mathcal{A}$ are semantically equivalent in the sense loosely introduced in the section 2.5.1 . In fact $\tilde{\mathcal{A}}^{sw}$ is the simpleton wrapper of $\mathcal{A}$, that is a PCA that only owns $\mathcal{A}$ in its attached configuration (see figure 8). Let us note that if $\mathcal{A}$ implements $\mathcal{B}$, then $\tilde{\mathcal{A}}^{sw}$ implements $\tilde{\mathcal{B}}^{sw}$.

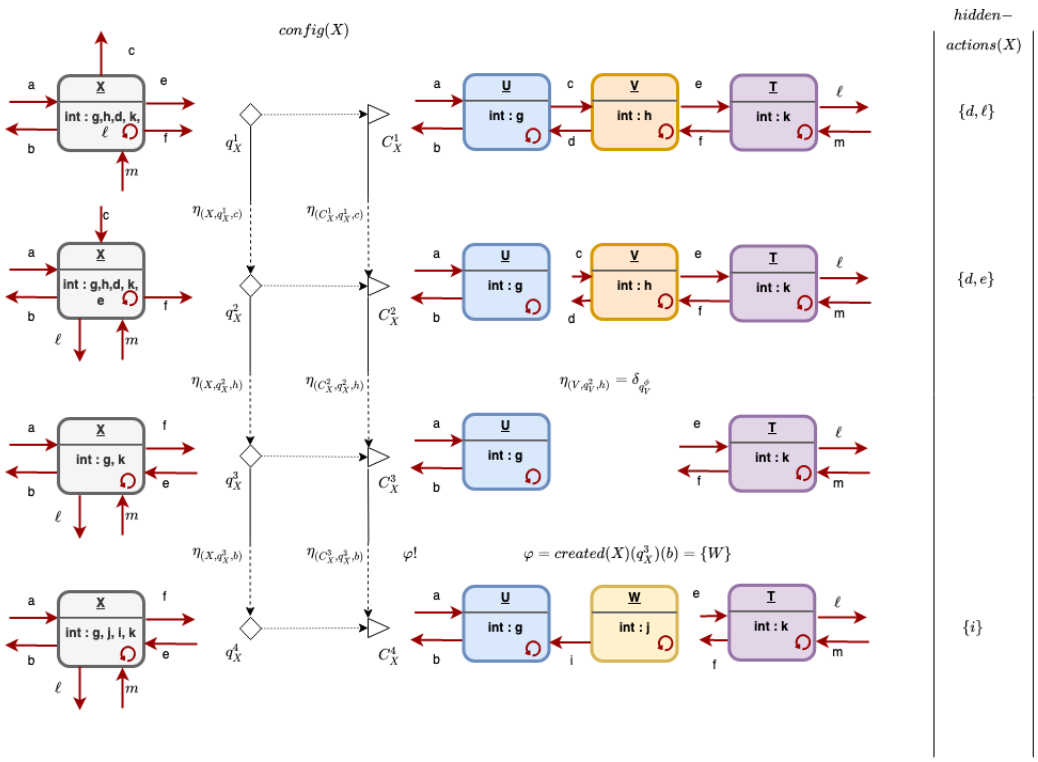


Fig. 6. Projection on PCA, part 1/2: The figure represents a PCA X like in figure 4. A sub-automaton T (in purple) appears in the configurations attached to the states visited by X. The PCA $Y = X \setminus \{T\}$ where the sub-automaton T is systematically removed is represented in figure 7.

Then we show that there is an (incomplete) execution-matching from $X_{\mathcal{A}}$ to $(X_{\mathcal{A}} \setminus \{\mathcal{A}\})||\tilde{\mathcal{A}}^{sw}$ (see figure 9). The domain of this executions-matching is the set of executions where $\mathcal{A}$ is not (re-)created.

After this, we always try to reduce any reasoning on $X_{\mathcal{A}}$ (resp. $X_{\mathcal{B}}$) on a reasoning on $(X_{\mathcal{A}} \setminus \{\mathcal{A}\})||\tilde{\mathcal{A}}^{sw}$ (resp. $(X_{\mathcal{B}} \setminus \{\mathcal{B}\})||\tilde{\mathcal{B}}^{sw}$).

2.5.4 Corresponding PCA. We show in section 10 that under certain reasonable technical assumptions (captured in the definition of corresponding PCA w. r. t. $\mathcal{A}$, $\mathcal{B}$) that $(X_{\mathcal{A}} \setminus \{\mathcal{A}\})$ and $(X_{\mathcal{B}} \setminus \{\mathcal{B}\})$ are semantically-equivalent. We can note Y an arbitrary PCA semantically-equivalent to $(X_{\mathcal{A}} \setminus \{\mathcal{A}\})$ and $(X_{\mathcal{B}} \setminus \{\mathcal{B}\})$. Finally, a reasoning on $\mathcal{E}||X_{\mathcal{A}}$ (resp. $\mathcal{E}||X_{\mathcal{B}}$) can be reduced to a reasoning on $(\mathcal{E}||Y)||\tilde{\mathcal{A}}^{sw} = \mathcal{E}'||\tilde{\mathcal{A}}^{sw}$ (resp. $\mathcal{E}'||\tilde{\mathcal{B}}^{sw}$) with $\mathcal{E}' = \mathcal{E}||Y$. Since $\tilde{\mathcal{A}}^{sw}$ implements $\tilde{\mathcal{B}}^{sw}$, we have already some results on $\mathcal{E}'||\tilde{\mathcal{A}}^{sw}$ and $\mathcal{E}'||\tilde{\mathcal{B}}^{sw}$ and so on $\mathcal{E}||X_{\mathcal{A}}$ and $\mathcal{E}||X_{\mathcal{B}}$ but only in a subset of executions (as long as neither $\mathcal{A}$ nor $\mathcal{B}$ is (re-)created, their comportment are the same.). This reduction is represented in the figure 10.

Cut-paste execution fragments with $\mathcal{A}$ (resp. $\mathcal{B}$) creation at the endpoints. The reduction roughly described in figure 10 holds only for executions fragments that do not create the automata $\mathcal{A}$ and $\mathcal{B}$ after their destruction (or at very last action). Some technical precautions have to be taken to be allowed to paste these fragments together to finally say that

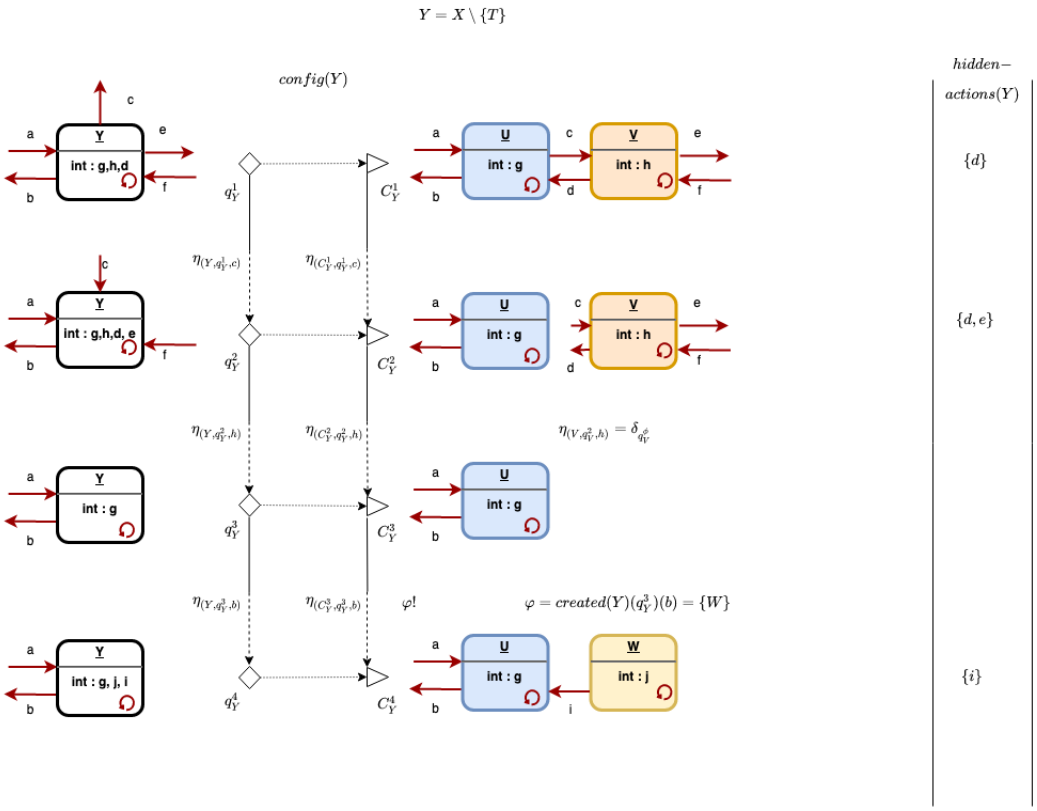


Fig. 7. Projection on PCA, part 2/2: the figure represents the PCA $Y = X \setminus \{T\}$ while the original PCA X is represented in figure 6. We can see that the sub-automaton T (in purple in figure 6) has been systematically removed from the configurations attached to the states visited by Y.

$\mathcal{A}$ implements $\mathcal{B}$ implies $X_{\mathcal{A}}$ implements $X_{\mathcal{B}}$. In fact, such a pasting is generally not possible for a fully information online scheduler. This observation motivated us to introduce the *creation-oblivious scheduler* that outputs (randomly) a transition without taking into account the internal actions of a sub-automaton $\mathcal{A}$ preceding its last destruction. Surprisingly, the fully-offline task-scheduler introduced in [2] (slightly modified to be adapted to dynamic setting) is not creation-oblivious and so does not verify monotonicity.

3 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 $\eta : \mathcal{F}_S \to \mathbb{R}^{\geq 0}$, 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. $\eta(S) \leq 1$). A measure space is denoted by $(S, \mathcal{F}_S, \eta)$ where η is a measure on $(S, \mathcal{F}_S)$.

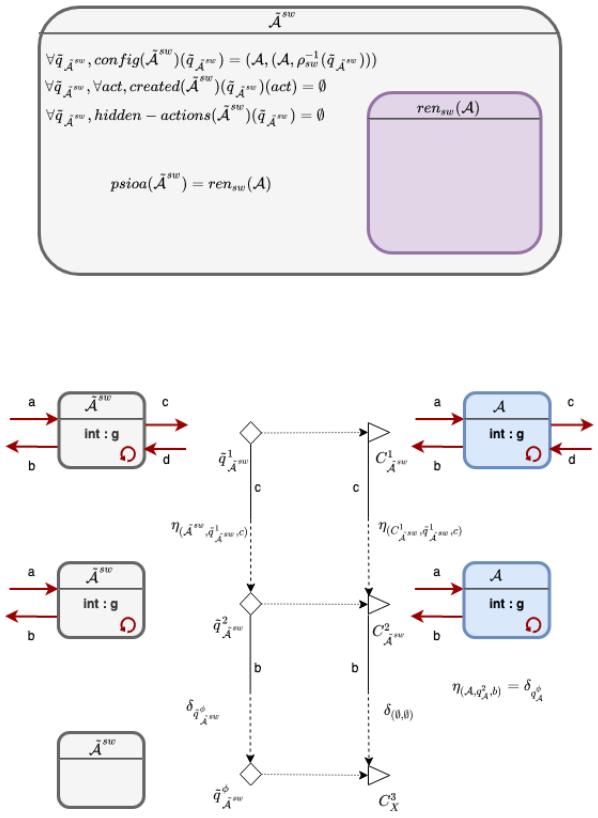


Fig. 8. The figure represents the simpleton wrapper $\tilde{\mathcal{A}}^{sw}$ of an automaton $\mathcal{A}$. The automaton $\tilde{\mathcal{A}}^{sw}$ is a PCA that only encapsulates one unique sub-automaton which is $\mathcal{A}$. We can confuse $\mathcal{A}$ and $\tilde{\mathcal{A}}^{sw}$ without impact. Intuitively, we can see $\tilde{\mathcal{A}}^{sw}$ as a wrapper of $\mathcal{A}$ that does not provide anything.

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 $C_1 \in \mathcal{F}_{s_1}, C_2 \in \mathcal{F}_{s_2}$, $\eta_1 \otimes \eta_2(C_1 \times C_2) = \eta_1(C_1) \cdot \eta_2(C_2)$. If S is countable, we note $\mathcal{P}(S) = 2^S$. If S_1 and S_2 are countable, we 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 $C \subset S, \eta(C) = \sum_{c \in C} \eta(\{c\})$. We define $Disc(S)$ and $SubDisc(S)$ to be respectively, the set of discrete probability and sub-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 $f : X \to Y$ is said to be measurable from $(X, \mathcal{F}_X) \to (Y, \mathcal{F}_Y)$ if the inverse image of each element of $\mathcal{F}_Y$ is an element

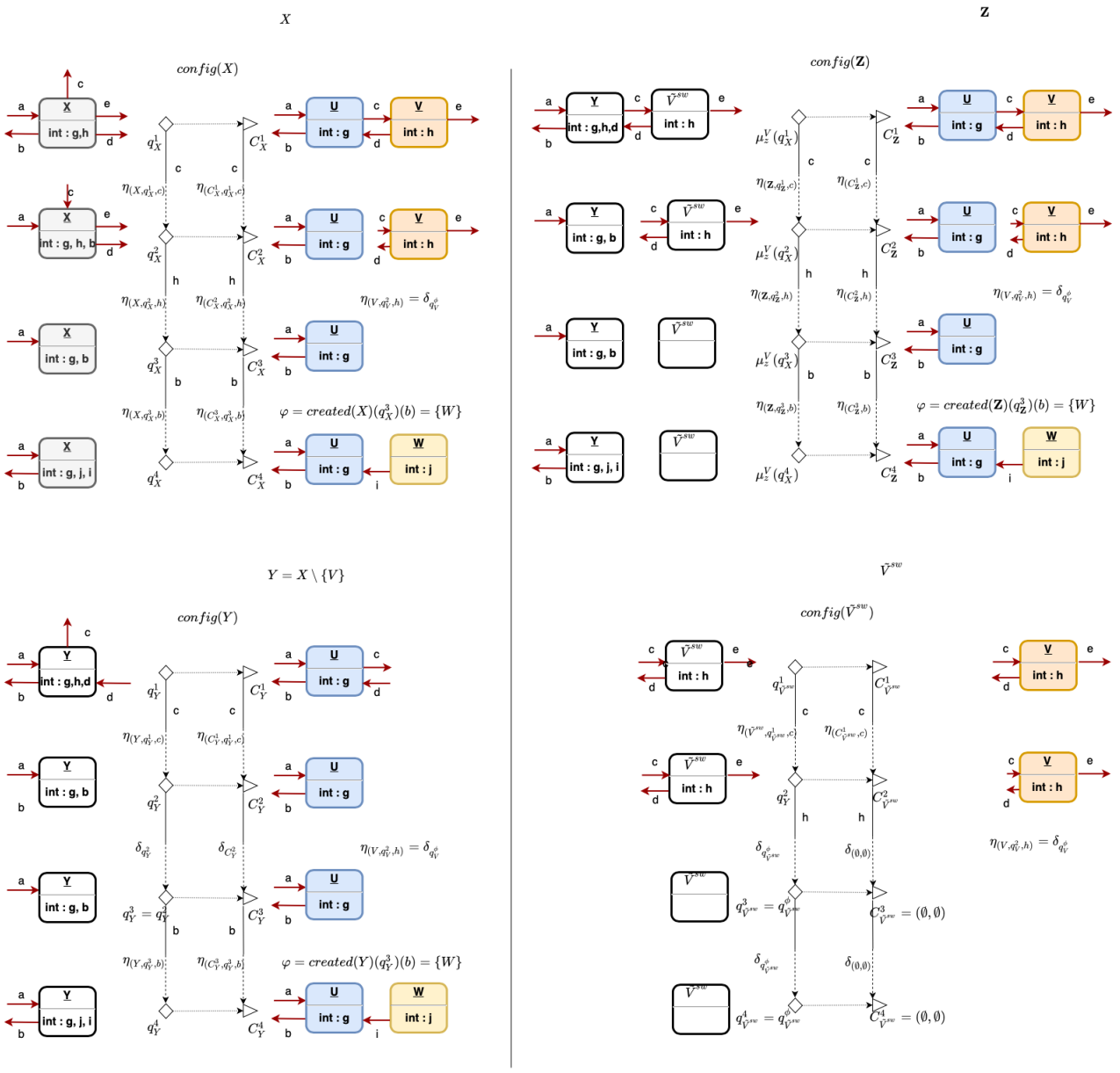


Fig. 9. The figure shows the similarities between two PCA X and $Z = (X \setminus \{V\})||\tilde{V}^{sw}$ represented in the top line. The two components of Z, i. e. $(X \setminus \{V\})$ and $\tilde{V}^{sw}$ are represented in the bottom line like in figure 7 and 8. These similarities are captured by the notions of executions-matching and hold as long as the the sub-automaton V is not created after a destruction. The idea is to reduce any reasoning on X to a reasoning on $(X \setminus \{V\})||\tilde{V}^{sw}$.

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.

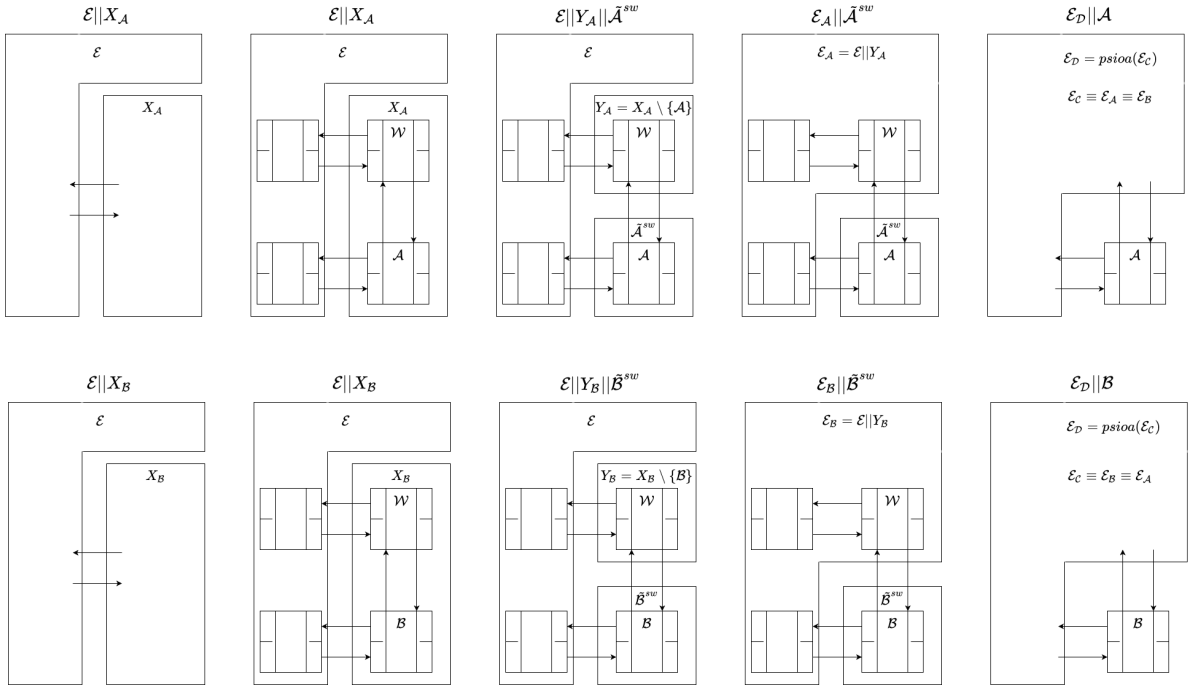


Fig. 10. The figure represents successive steps to reduce the problem of an environment $\mathcal{E}$ that tries to distinguish two PCA $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ (represented at first column) to a problem of an environment $\mathcal{E}_{\mathcal{D}}$ that tries to distinguish the automata $\mathcal{A}$ and $\mathcal{B}$ (represented at last column). The second column just remarks that the only difference between $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ is that $\mathcal{A}$ supplants $\mathcal{B}$ in $X_{\mathcal{A}}$. The third column consist in the steps of deprivation (see section 2.5.2) and reconstruction (see section 2.5.3). The fourth column rearrange the parenthesis by associativity of the parallel composition to highlight $\mathcal{E}_{\mathcal{A}} = \mathcal{E}||(X_{\mathcal{A}} \setminus \{\mathcal{A}\})$ and $\mathcal{E}_{\mathcal{B}} = \mathcal{E}||(X_{\mathcal{B}} \setminus \{\mathcal{B}\})$ as respective environments of $\tilde{\mathcal{A}}^{sw}$ and $\tilde{\mathcal{B}}^{sw}$. In last column, we remark that $\mathcal{E}_{\mathcal{A}}$ and $\mathcal{E}_{\mathcal{B}}$ are semantically equivalent so there it is like we deal with a common environment $\mathcal{E}_C$ for both $\tilde{\mathcal{A}}^{sw}$ and $\tilde{\mathcal{A}}^{sw}$. If we consider only the psioa components, the problem is reduced to a common environment $\mathcal{E}_{\mathcal{D}}$ that tries to distinguish the automata $\mathcal{A}$ and $\mathcal{B}$. The reasoning holds only as long as the automata $\mathcal{A}$ and $\mathcal{B}$ are not created after their destructions.

4 PROBABILISTIC SIGNATURE INPUT/OUTPUT AUTOMATA (PSIOA)

This section aims to introduce the first brick of our formalism, i. e. the probabilistic signature input/output automata (PSIOA). A PSIOA is the result of the generalization of probabilistic input/output automata (PIOA) [8] and signature input/output automata (SIOA) [1]. A PSIOA is thus an automaton that can randomly move from one *state* to another in response to some *actions*. The set of possible actions is the *signature* of the automaton and is partitioned into *input*, *output* and *internal* actions. An action can often be both the input of one automaton and the output of another one to captures the idea that the behavior of an automaton can influence the behavior of another one. As for the SIOA [1], the signature of a PSIOA can change according to the current state of the automaton, which allows us to formalise dynamicity later. The figure 11 gives a first intuition of what is a PSIOA.

4.1 Action Signature

We use the signature approach from [1].

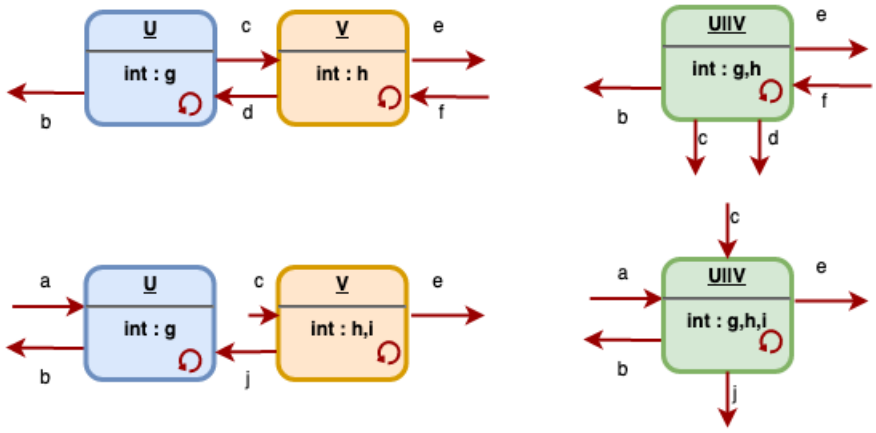


Fig. 11. A representation of two automata U and V. In the top line, we see the PSIOA U in a state q_U^1, s. t. $sig(U)(q_U^1) = (out(U)(q_U^1), in(U)(q_U^1), int(U)(q_U^1)) = (\{b, c\}, \{d\}, \{g\})$, the PSIOA V in a state q_V^1, s. t. $sig(V)(q_V^1) = (out(V)(q_V^1), in(V)(q_V^1), int(V)(q_V^1)) = (\{d, e\}, \{c, f\}, \{h\})$ and the result of their composition, the PSIOA $U||V$ in a state (q_U^1, q_V^1), s. t. $sig(U||V)((q_U^1, q_V^1)) = (out(U||V)((q_U^1, q_V^1)), in(U||V)((q_U^1, q_V^1)), int(U||V)((q_U^1, q_V^1) = (\{b, c, d, e\}, \{f\}, \{g, h\})$. In the second line we see the same PSIOA but in different states. We see the PSIOA U in a state q_U^2, s. t. $sig(U)(q_U^2) = (out(U)(q_U^2), in(U)(q_U^2), int(U)(q_U^2)) = (\{b\}, \{a, j\}, \{g\})$, the PSIOA V in a state q_V^2, s. t. $sig(V)(q_V^2) = (out(V)(q_V^2), in(V)(q_V^2), int(V)(q_V^2)) = (\{e, j\}, \{c\}, \{h, i\})$ and the result of their composition, the PSIOA $U||V$ in a state (q_U^2, q_V^2), s. t. $sig(U||V)((q_U^2, q_V^2)) = (out(U||V)((q_U^2, q_V^2)), in(U||V)((q_U^2, q_V^2)), int(U||V)((q_U^2, q_V^2) = (\{b, e, j\}, \{a, c\}, \{g, h, i\})$.

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 $aut : Autids \to Auts$. $aut(\mathcal{A})$ 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 $local(\mathcal{A})(q) = (out(\mathcal{A})(q), int(\mathcal{A})(q))$. 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 we 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 trigger, in any state. In the same way $UI(\mathcal{A}) = \bigcup_{q \in Q} in(\mathcal{A})(q)$, $UO(\mathcal{A}) = \bigcup_{q \in Q} out(\mathcal{A})(q)$, $UH(\mathcal{A}) = \bigcup_{q \in Q} int(\mathcal{A})(q)$, $UL(\mathcal{A}) = \bigcup_{q \in Q} \widehat{local}(\mathcal{A})(q)$, $UE(\mathcal{A}) = \bigcup_{q \in Q} \widehat{ext}(\mathcal{A})(q)$.

4.2 PSIOA

We combine the SIOA of [1] with the PIOA of [8]:

Definition 4.1 (PSIOA). A PSIOA $\mathcal{A} = (Q_{\mathcal{A}}, \bar{q}_{\mathcal{A}}, sig(\mathcal{A}), D_{\mathcal{A}})$, where:

- $Q_{\mathcal{A}}$ (a.k.a. $states(\mathcal{A})$) is a countable set of *states*, $(Q_{\mathcal{A}}, 2^{Q_{\mathcal{A}}})$ is a measurable space called the *state space*,
- $\bar{q}_{\mathcal{A}}$ (a. k. a. $start(\mathcal{A})$) is the unique *start state*.

- $sig(\mathcal{A}) : q \in Q_{\mathcal{A}} \mapsto sig(\mathcal{A})(q) = (in(A)(q), out(A)(q), int(A)(q))$ is the signature function that maps each state to a triplet of mutually disjoint countable set of *actions*, respectively called *input*, *output* and *internal* actions.
- $D_{\mathcal{A}} \subset Q_{\mathcal{A}} \times acts(\mathcal{A}) \times Disc(Q_{\mathcal{A}})$ ($D_{\mathcal{A}}$ a. k. a. $dtrans(\mathcal{A})$) is the set of *probabilistic discrete transitions* where $\forall (q, a, \eta) \in D_{\mathcal{A}} : a \in \widehat{sig}(\mathcal{A})(q)$. If (q, a, η) is an element of $D_{\mathcal{A}}$, we write $q \xrightarrow{a} \eta$ and action a is said to be *enabled* at q.

In addition $\mathcal{A}$ must satisfy the following conditions[1]:

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

Notation. For every PSIOA $\mathcal{A} \triangleq (Q_{\mathcal{A}}, \bar{q}_{\mathcal{A}}, sig(\mathcal{A}), D_{\mathcal{A}})$, we note $states(\mathcal{A}) \triangleq Q_{\mathcal{A}}$, $start(\mathcal{A}) \triangleq \bar{q}_{\mathcal{A}}$, $dtrans(\mathcal{A}) \triangleq D_{\mathcal{A}}$. We also note $steps(\mathcal{A}) \triangleq \{(q, a, q') \in Q_{\mathcal{A}} \times acts(\mathcal{A}) \times Q_{\mathcal{A}} | \exists (q, a, \eta) \in D_{\mathcal{A}}, q' \in supp(\eta)\}$.

4.3 Execution, Trace

We use the classic notions of execution and trace from [8] to speak about the comportment of a PSIOA.

Definition 4.2 (fragment, execution and trace of PSIOA). An *execution fragment* of a PSIOA $\mathcal{A} = (Q_{\mathcal{A}}, \bar{q}_{\mathcal{A}}, sig(\mathcal{A}), D_{\mathcal{A}})$ 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, $(q^i, a^i, q^{i+1}) \in steps(\mathcal{A})$

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 note $states(\alpha)$ (resp. $actions(\alpha)$) for the set of states (resp. actions) that compose α. The length $|\alpha|$ of a finite execution fragment α is the number of transitions along α. The length of an infinite execution fragment α is infinite, ($|\alpha| = \omega$). If We use $Frags(\mathcal{A})$ (resp., $Frags^*(\mathcal{A})$) 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 $\alpha \in Execs(\mathcal{A})$ with $\beta = trace(\alpha)$. $Traces(\mathcal{A})$ (resp., $Traces^*(\mathcal{A})$) denotes the set of all (resp., all finite) traces of $\mathcal{A}$. We define a concatenation operator $\frown$ for execution fragments as follows. If $\alpha = q^0 a^1 q^1 ... a^n q^n \in Frags^*(\mathcal{A})$ and $\alpha' = s^0 b^1 s^1 ... \in Frags^(\mathcal{A})$, we define $\alpha \frown \alpha' = q^0 a^1 q^1 ... a^n s^0 b^1 s^1 ...$ only if $s^0 = q^n$, otherwise $\alpha \frown \alpha'$ is undefined. Hence the notation $\alpha \frown s^0 b^1 s^1 ...$ implicitly means $s^0 = lstate(\alpha)$. We also note $\alpha \frown (b^1, s^1)$ to states $\alpha \frown lstate(\alpha) b^1 s^1$. Let $\alpha, \alpha' \in Frags(\mathcal{A})$, then α is a proper prefix of α' iff $\exists \alpha'' \in Frags(\mathcal{A})$ such that $\alpha' = \alpha \frown \alpha''$ with $\alpha \neq \alpha'$. In that case, we note $\alpha < \alpha'$. We note $\alpha \leq \alpha'$ if $\alpha < \alpha'$ or $\alpha = \alpha'$ and say that α is a prefix of α'. We also overload $\frown$ and use it for concatenating traces in the obvious manner.

Definition 4.3 (reachable state). Let $\mathcal{A} = (Q_{\mathcal{A}}, \bar{q}_{\mathcal{A}}, sig(\mathcal{A}), D_{\mathcal{A}})$ be a PSIOA. A state $q \in Q_{\mathcal{A}}$ is said *reachable* if it exists a finite execution that ends with q. We note $reachable(\mathcal{A})$ the set of reachable states of $\mathcal{A}$. We also note for every $s \in \mathbb{N}$, $reachable_{\leq s}(\mathcal{A}) = \{q^* \in reachable(\mathcal{A}) | \exists z \leq s, \exists \alpha = q^0 a^1 q^1 ... q^{z-1} a^z q^*\}$. This is the set of states reachable in

[1]The conjunction of conditions $\mathbf{E_1}$ and $\mathbf{T_1}$ could allow us to model $\mathcal{D}_{\mathcal{A}}$ as a partial function from $Q_{\mathcal{A}} \times acts(\mathcal{A})$ to $Disc(Q_{\mathcal{A}})$. However, we keep this presentation to stay as close as possible to the usual notation of the literature. For the same reasons, we use both $\mathcal{A} \triangleq (Q_{\mathcal{A}}, \bar{q}_{\mathcal{A}}, sig(\mathcal{A}), D_{\mathcal{A}})$ and $\mathcal{A} \triangleq (states(\mathcal{A}), start(\mathcal{A}), sig(\mathcal{A}), dtrans(\mathcal{A}))$

less than s actions. We also note for every $s \in \mathbb{N}^*$, $reachable_s(\mathcal{A}) = reachable_{\leq s}(\mathcal{A}) \setminus reachable_{\leq s-1}(\mathcal{A})$. This is the set of states reachable in s actions but not less. By convention, $reachable_0(\mathcal{A}) = \{\bar{q}_{\mathcal{A}}\}$.

The set of sets $\{reachable_s(\mathcal{A}) | s \in \mathbb{N}\}$ is clearly a partition of $reachable(\mathcal{A})$.

4.4 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 4.4 (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 4.5 (Composition of Signatures). Let $\Sigma = (in, out, int)$ and $\Sigma' = (in', out', int')$ be compatible signatures. Then we define their composition $\Sigma \times \Sigma = (in \cup in' - (out \cup out'), out \cup out', int \cup int')$[2].

Signature composition is clearly commutative and associative.

Now we can define the compatibility of several automata with the compatibility of their attached signatures. First we define compatibility at a state, and discrete transition for a set of automata for a particular compatible state.

Definition 4.6 (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 $q = (q_1, ..., q_n) \in Q_{\mathbf{A}} = Q_{\mathcal{A}_1} \times ... \times Q_{\mathcal{A}_n}$. We say $\mathcal{A}_1, ..., \mathcal{A}_n$ are *partially-compatible* at state q (or $\mathbf{A}$ is *partially-compatible* at state q) if $\{sig(\mathcal{A}_1)(q_1), ..., sig(\mathcal{A}_n)(q_n)\}$ is a set of compatible signatures. In this case we note $sig(\mathbf{A})(q) = sig(\mathcal{A}_1)(q_1) \times ... \times sig(\mathcal{A}_n)(q_n)$ as per definition 4.5 and we note $\eta_{(\mathbf{A},q,a)} \in Disc(Q_{\mathbf{A}})$, s. t. for every action $a \in \widehat{sig}(\mathbf{A})(q)$, $\eta_{(\mathbf{A},q,a)} = \eta_1 \otimes ... \otimes \eta_n \in Disc(Q_{\mathbf{A}})$ 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, $\eta_j = \delta_{q_j}$ (where δ_{q_j} the Dirac distribution with $supp(\delta_{q_j}) = \{q_j\}$ and $\delta_{q_j}(q_j) = 1$)

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

We will say a set of automata is partially-compatible, if any reachable state is compatible. This motivates the two following definitions.

Definition 4.7 (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 state of $\mathbf{A}$.
- 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 4.8 (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.

Now we are able to define compatibility for a set of PSIOA.

[2] not to be confused with Cartesian product. We keep this notation to stay as close as possible to the literature.

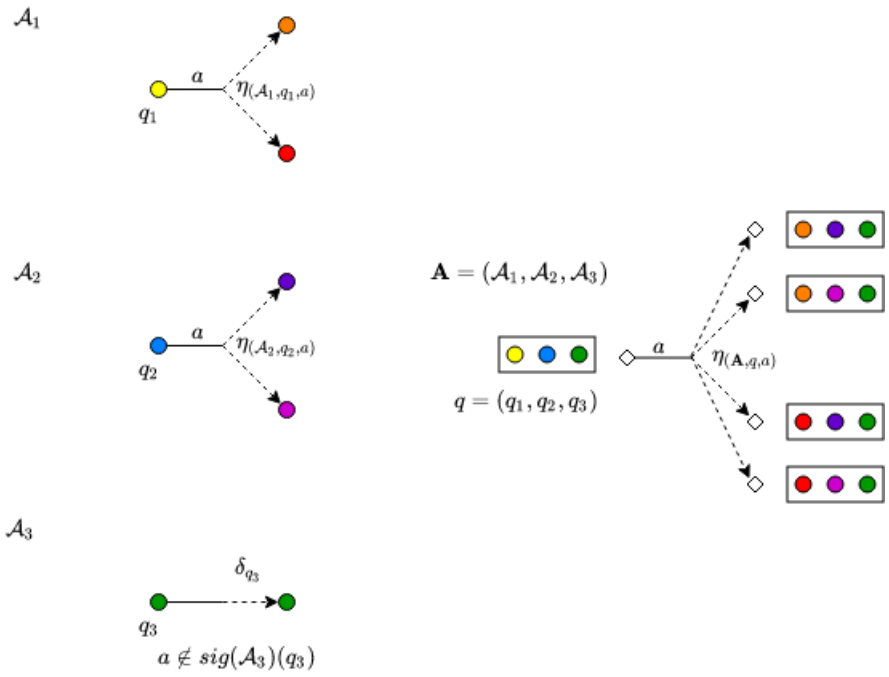


Fig. 12. The family transition is obtain by the transitions of the automata of the family.

Definition 4.9 (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. if $\mathbf{A}$ is *ℓ-partially-compatible* for every $\ell \in \mathbb{N}$.

Finally, we can formally define our operation of composition. This is the central operation of any IOA formalism.

Definition 4.10 (partially-compatible PSIOA composition). If $\mathbf{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ is a partially-compatible set of PSIOA, with $\mathcal{A}_i = (Q_{\mathcal{A}_i}, \bar{q}_{\mathcal{A}_i}, sig(\mathcal{A}_i), D_{\mathcal{A}_i})$, then their partial-composition $\mathcal{A}_1||...||\mathcal{A}_n$, is defined to be $\mathcal{A} = (Q_{\mathcal{A}}, \bar{q}_{\mathcal{A}}, sig(\mathcal{A}), D_{\mathcal{A}})$, where:

- $Q_{\mathcal{A}} = \{q \in Q_{\mathcal{A}_1} \times ... \times Q_{\mathcal{A}_n} | q \text{ is a reachable state of } \mathbf{A}\}$.
- $\bar{q}_{\mathcal{A}} = (\bar{q}_{\mathcal{A}_1}, ..., \bar{q}_{\mathcal{A}_n})$
- $sig(\mathcal{A}) : q = (q_1, ..., q_n) \in Q_{\mathcal{A}} \mapsto sig(\mathcal{A})(q) = sig(\mathcal{A}_1)(q_1) \times ... \times sig(\mathcal{A}_n)(q_n)$ as per definition 4.5.
- $D_{\mathcal{A}} \subset Q_{\mathcal{A}} \times acts(\mathcal{A}) \times Disc(Q_{\mathcal{A}})$ is the set of triples $(q, a, \eta_{(\mathbf{A},q,a)})$ so that $q \in Q_{\mathcal{A}}$ and $a \in \widehat{sig}(\mathbf{A})(q)$

This formalism extends the one proposed in [1] where it is required that all (potentially non-reachable) states are compatible. In addition to being slightly less restrictive, this notion of composability i) may facilitate the expression of mobile agents moving from one system to another (see section 5) and ii) will allow the proof of theorem of implementation monotonicity w.r.t. PSIOA creation (see section 6).

Given a parallel composition $\mathcal{A} = \mathcal{A}_1||...||\mathcal{A}_n$ of n PSIOA, we define the projection of an execution fragments of $\mathcal{A}$ onto one of the $\mathcal{A}_i$, $i \in [1:n]$, in the usual way: the state components for all PSIOA other than $\mathcal{A}_i$ are removed, and so are all actions in which $\mathcal{A}_i$ does not participate.

Definition 4.11 (Execution projection for PSIOA). Let $\mathcal{A} = \mathcal{A}_1||...||\mathcal{A}_n$ be a PSIOA. Let $\alpha = q^0 a^1 ... a^n q^n ... \in Frags(\mathcal{A})$. Then, $\forall i \in [1:n]$, we define $\alpha \upharpoonright \mathcal{A}_i$ to be the sequence resulting from:

(1) replacing each $q^j = (q_1^j, ..., q_n^j)$ by its i'th component q_i^j and then
(2) removing all $a^j q_i^j$ s. t. $a^j \notin \widehat{sig}(\mathcal{A}_i)(q_i^{j-1})$.

The idea behind execution projection is to retain only the state of $\mathcal{A}_i$, and only the actions which $\mathcal{A}_i$ participates in. It has been shown in [1] (theorem 4, page 11), that for every PSIOA $\mathcal{A} = \mathcal{A}_1||...||\mathcal{A}_n$, $\forall \alpha \in Execs(\mathcal{A})$, $\forall i \in [1:n]$, $\alpha \upharpoonright \mathcal{A}_i \in Execs(\mathcal{A}_i)$.

4.5 Scheduler: define a measure on executions and traces

An inherent non-determinism appears for composable input/output (I/O) automata. Indeed, after composition (or even before), it is natural to obtain a state with several enabled actions. The most common case is the reception of two concurrent messages in flight from two different processes. This non-determinism must be solved if we want to define a probability measure on the automata executions and be able to say that a situation is likely to occur or not. To solve the non-determinism, we use a scheduler that chooses an enabled action from a signature.

4.5.1 Scheduler: general definition. A scheduler is hence a function that takes an execution fragment as input and outputs the probability distribution on the set of transitions that will be triggered. We reuse the formalism from [8] with the syntax from [2].

Definition 4.12 (scheduler). A *scheduler* of a PSIOA $\mathcal{A}$ is a function
$\sigma : Frags^*(\mathcal{A}) \rightarrow SubDisc(dtrans(\mathcal{A}))$ such that $(q, a, \eta) \in supp(\sigma(\alpha))$ implies $q = lstate(\alpha)$. Here $SubDisc(dtrans(\mathcal{A}))$ is the set of discrete sub-probability distributions on $dtrans(\mathcal{A})$. Loosely speaking, σ decides (probabilistically) which transition to take after each finite execution fragment α. Since this decision is a discrete sub-probability measure, it may be the case that σ chooses to halt after α with non-zero probability: $1 - \sigma(\alpha)(dtrans(\alpha)) > 0$. We note $schedulers(\mathcal{A})$ the set of schedulers of $\mathcal{A}$.

Definition 4.13 (measure $\epsilon_{\sigma,\alpha}$ generated by a scheduler and a fragment). A scheduler σ and a finite execution fragment α generate a measure $\epsilon_{\sigma,\alpha}$ on the sigma-field $\mathcal{F}_{Execs(\mathcal{A})}$ generated by cones of execution fragments, where each cone $C_{\alpha'}$ is the set of execution fragments that have α' as a prefix, i. e. $C_{\alpha'} = \{\alpha \in Execs(\mathcal{A}) | \alpha' \leq \alpha\}$. The measure of a cone $C_{\alpha'}$ is defined recursively as follows:

$$\epsilon_{\sigma,\alpha}(C_{\alpha'}) =: \begin{cases} 0 & \text{if both } \alpha' \not\leq \alpha \text{ and } \alpha \not\leq \alpha' \\ 1 & \text{if } \alpha' \leq \alpha \\ \epsilon_{\sigma,\alpha}(C_{\alpha''}) \cdot \sigma(\alpha'')(\eta_{(\mathcal{A},q',a)}) \cdot \eta_{(\mathcal{A},q',a)}(q) & \text{if } \alpha \leq \alpha'' \text{ and } \alpha' = \alpha'' \frown q' a q \end{cases}$$

Standard measure theoretic arguments ensure that $\epsilon_{\sigma,\alpha}$ is well-defined. We call the state $fstate(\alpha)$ the first state of $\epsilon_{\sigma,\alpha}$ and denote it by $fstate(\epsilon_{\sigma,\alpha})$. If α consists of the start state $start(\mathcal{A})$ only, we call $\epsilon_{\sigma,\alpha}$ a probabilistic execution of $\mathcal{A}$. Let μ be a discrete probability measure over $Frags^*(\mathcal{A})$. We denote by $\epsilon_{\sigma,\mu}$ the measure $\sum_{\alpha \in supp(\mu)} \mu(\alpha) \cdot \epsilon_{\sigma,\alpha}$ and we say that $\epsilon_{\sigma,\mu}$ is generated by σ and μ. We call the measure $\epsilon_{\sigma,\mu}$ a generalized probabilistic execution fragment of $\mathcal{A}$. If every execution fragment in $supp(\mu)$ consists of a single state, then we call $\epsilon_{\sigma,\mu}$ a probabilistic execution fragment of $\mathcal{A}$.

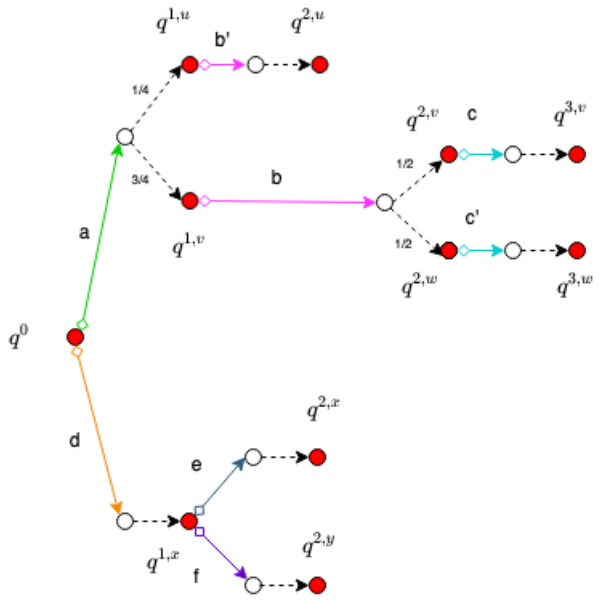


Fig. 13. Non-deterministic execution: The scheduler allows us to solve the non-determinism, by triggering an action among the enabled one. Typically after execution $\alpha = q^0\ d\ q^{1,x}$, the actions e and f are enabled and the probability to take one transition is given by the scheduler σ that computes $\sigma(\alpha)$.

4.5.2 Scheduler Schema. Without restriction, a scheduler could become a too powerful adversary for practical applications. Hence, it is common to only consider a subset of schedulers, called a *scheduler schema*. Typically, a classic limitation is often described by a scheduler with "partial online information". Some formalism has already been proposed in [8] (section 5.6) to impose the scheduler that its choices are correlated for executions fragments in the same equivalence class where both the equivalence relation and the correlation must to be defined. This idea has been reused and simplified in [3] that defines equivalence classes on actions, called *tasks*. Then, a task-scheduler (a.k.a. "off-line" scheduler) selects a sequence of tasks $T_1, T_2, \ldots$ in advance that it cannot modify during the execution of the automaton. After each transition, the next task T_i triggers an enabled action if there is no ambiguity and is ignored otherwise. One of our main contribution, the theorem of implementation monotonicity w.r.t. PSIOA creation, is ensured only for a certain scheduler schema, so-called *creation-oblivious*. However, we will see that the practical set of task-schedulers are not creation-oblivious.

Definition 4.14 (scheduler schema). A *scheduler schema* is a function that maps any PSIOA W to a subset of $schedulers(W)$.

4.6 Implementation

In last subsection, we defined a measure of probability on executions with the help of a scheduler to solve non-determinism. Now we can define the notion of implementation. The intuition behind this notion is the fact that any environment $\mathcal{E}$ that would interact with both $\mathcal{A}$ and $\mathcal{B}$, would not be able to distinguish $\mathcal{A}$ from $\mathcal{B}$. The classic use-case is to formally show that a (potentially very sophisticated) algorithm implements a specification.

For us, an environment is simply a partially-compatible automaton, but in practice, he will play the role of a "distinguisher".

Definition 4.15 (Environment). A probabilistic environment for PSIOA $\mathcal{A}$ is a PSIOA $\mathcal{E}$ such that $\mathcal{A}$ and $\mathcal{E}$ are partially-compatible. We note $env(\mathcal{A})$ the set of environments of $\mathcal{A}$.

Now we define *insight function* which is a function that captures the insights that could be obtained by an external observer to attempt a distinction.

Definition 4.16 (insight function). An *insight-function* is a function $f_{(.,.)}$ parametrized by a pair $(\mathcal{E}, \mathcal{A})$ of PSIOA where $\mathcal{E} \in env(\mathcal{A})$ so that for every PSIOA $\mathcal{E}$, it exists a measurable space $(G_{\mathcal{E}}, \mathcal{F}_{G_{\mathcal{E}}})$, s. t. for every pair $(\mathcal{A}, \mathcal{B})$ of PSIOA where $\mathcal{E} \in env(\mathcal{A}) \cap env(\mathcal{B})$, $f_{(\mathcal{E},\mathcal{A})}$ (resp. $f_{(\mathcal{E},\mathcal{B})}$) is a measurable function from $(Execs(\mathcal{E}||\mathcal{A}), \mathcal{F}_{Execs(\mathcal{E}||\mathcal{A})})$ (resp. $(Execs(\mathcal{E}||\mathcal{B}), \mathcal{F}_{Execs(\mathcal{E}||\mathcal{B})})$) to $(G_{\mathcal{E}}, \mathcal{F}_{G_{\mathcal{E}}})$.

The point is that the arrival space $(G_{\mathcal{E}}, \mathcal{F}_{G_{\mathcal{E}}})$ is the same for the two functions $f_{(\mathcal{E},\mathcal{A})}$ and $f_{(\mathcal{E},\mathcal{B})}$ to enable a comparison. Some examples of insight-functions are the trace function and the print function introduced later.

Since an insight-function $f_{(.,.)}$ is measurable, we can define the image measure of $\epsilon_{\sigma,\mu}$ under $f_{(\mathcal{E},\mathcal{A})}$, i. e. the probability to obtain a certain external perception under a certain scheduler σ and a certain probability distribution μ on the starting executions.

Definition 4.17 (f-dist). Let $f_{(.,.)}$ be an insight-function. Let $(\mathcal{E}, \mathcal{A})$ be a pair of PSIOA where $\mathcal{E} \in env(\mathcal{A})$. Let μ be a probability measure on $(Execs(\mathcal{E}||\mathcal{A}), \mathcal{F}_{Execs(\mathcal{E}||\mathcal{A})})$, and $\sigma \in schedulers(\mathcal{E}||\mathcal{A})$. We define $f\text{-}dist_{(\mathcal{E},\mathcal{A})}(\sigma, \mu)$, to be the image measure of $\epsilon_{\sigma,\mu}$ under $f_{(\mathcal{E},\mathcal{A})}$ (i. e. the function that maps any $C \in \mathcal{F}_{G_{\mathcal{E}}}$ to $\epsilon_{\sigma,\mu}(f^{-1}_{(\mathcal{E},\mathcal{A})}(C))$) . We note $f\text{-}dist_{(\mathcal{E},\mathcal{A})}(\sigma)$ for $f\text{-}dist_{(\mathcal{E},\mathcal{A})}(\sigma, \delta_{start(\mathcal{E},\mathcal{A})})$.

We can see next definition of f-implementation as the incapacity of an environment to distinguish two automata if it uses only information filtered by the insight function f.

Definition 4.18 (f-implementation). Let $f_{(.,.)}$ be an insight-function. Let *Sch* be a scheduler schema. We say that $\mathcal{A}$ *f-implements $\mathcal{B}$ according to Sch*, noted $\mathcal{A} \leq^{f}_{Sch} \mathcal{B}$, if $\forall \mathcal{E} \in env(\mathcal{A}) \cap env(\mathcal{B})$, $\forall \sigma \in Sch(\mathcal{E}||\mathcal{A})$, $\exists \sigma' \in Sch(\mathcal{E}||\mathcal{B})$, $f\text{-}dist_{(\mathcal{E},\mathcal{A})}(\sigma) = f\text{-}dist_{(\mathcal{E},\mathcal{B})}(\sigma')$.

We states a necessary and sufficient condition to obtain composability of f-implementation.

Definition 4.19. Let $f_{(.,.)}$ be an insight-function. We say that $f_{(.,.)}$ is *stable by composition* if for every quadruplet of PSIOA $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{B}, \mathcal{E})$, s. t. $\mathcal{B}$ is partially compatible with $\mathcal{A}_1$ and $\mathcal{A}_2$, $\mathcal{E} \in env(\mathcal{B}||\mathcal{A}_1) \cap env(\mathcal{B}||\mathcal{A}_2)$, for every $(C_1, C_2) \in \mathcal{F}_{Execs(\mathcal{E}||\mathcal{B}||\mathcal{A}_1)} \times \mathcal{F}_{Execs(\mathcal{E}||\mathcal{B}||\mathcal{A}_2)}$, $f_{(\mathcal{E}||\mathcal{B},\mathcal{A}_1)}(C_1) = f_{(\mathcal{E}||\mathcal{B},\mathcal{A}_2)}(C_2) \implies f_{(\mathcal{E},\mathcal{B}||\mathcal{A}_1)}(C_1) = f_{(\mathcal{E},\mathcal{B}||\mathcal{A}_2)}(C_2)$

We can restate classic theorem of composability of implementation in a quite general form.

Theorem 4.20 (Implementation composability). *Let $f_{(.,.)}$ be an insight-function stable by composition. Let Sch be a scheduler schema. Let $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{B}$ be PSIOA, s.t. $\mathcal{A}_1 \leq^{f}_{Sch} \mathcal{A}_2$. If $\mathcal{B}$ is partially compatible with $\mathcal{A}_1$ and $\mathcal{A}_2$ then $\mathcal{B}||\mathcal{A}_1 \leq^{f}_{Sch} \mathcal{B}||\mathcal{A}_2$.*

Proof. If $\mathcal{E}$ is an environment for both $\mathcal{B}||\mathcal{A}_1$ and $\mathcal{B}||\mathcal{A}_2$, then $\mathcal{E}' = \mathcal{E}||\mathcal{B}$ is an environment for both $\mathcal{A}_1$ and $\mathcal{A}_2$. By associativity of parallel composition, we have for every $i \in \{1, 2\}$, $(\mathcal{E}||\mathcal{B})||\mathcal{A}_i = \mathcal{E}||(\mathcal{B}||\mathcal{A}_i)$. Since $\mathcal{A}_1 \leq^{f}_{SSchema}$

$\mathcal{A}_2$, for any scheduler $\sigma \in Sch((\mathcal{E}||\mathcal{B})||\mathcal{A}_1)$, it exists a corresponding scheduler $\sigma' \in Sch((\mathcal{E}||\mathcal{B})||\mathcal{A}_2)$, s. t. $f\text{-}dist_{(\mathcal{E}||\mathcal{B}),\mathcal{A}_1}(\epsilon_\sigma) = f\text{-}dist_{(\mathcal{E}||\mathcal{B}),\mathcal{A}_2}(\epsilon_{\sigma'})$. Thus, by stability by composition, for any scheduler $\sigma \in Sch(\mathcal{E}||(\mathcal{B}||\mathcal{A}_1))$, it exists a corresponding schedule $\sigma' \in Sch(\mathcal{E}||(\mathcal{B}||\mathcal{A}_2))$, s. t. $f\text{-}dist_{(\mathcal{E},(\mathcal{B}||\mathcal{A}_1))}(\epsilon_\sigma) = f\text{-}dist_{(\mathcal{E},(\mathcal{B}||\mathcal{A}_2))}(\epsilon_{\sigma'})$, that is $\mathcal{A}_1||\mathcal{B} \leq^f_{Sch} \mathcal{A}_2||\mathcal{B}$. □

Now we introduce the insight function $print_{(\mathcal{E},\mathcal{A})}$ that we will use for monotonicity of implementation w.r.t. PSIOA creation.

Definition 4.21 ($print_{(\mathcal{E},\mathcal{A})}$). Let $\mathcal{A}$ be a PSIOA and $\mathcal{E} \in env(\mathcal{A})$ We note

$$print_{(\mathcal{E},\mathcal{A})} : \begin{cases} Frags(\mathcal{E}||\mathcal{A}) & \rightarrow & Frags(\mathcal{E}) \times trace(\mathcal{A}) \\ \alpha & \mapsto & (\alpha \upharpoonright \mathcal{E}, trace(\alpha)) \end{cases}$$

We note $Prints(\mathcal{E},\mathcal{A}) \triangleq range(print_{(\mathcal{E},\mathcal{A})})$ and $\forall((e,\beta),(e',\beta')) \in Prints(\mathcal{E},\mathcal{A})^2$, $(e,\beta) \leq (e',\beta')$ iff both $e \leq e'$ and $\beta \leq \beta'$. For every $\zeta \in Prints(\mathcal{E},\mathcal{A})$, we note $C_\zeta = \{\zeta' \in Prints_{(\mathcal{E},\mathcal{A})} | \zeta \leq \zeta'\}$ called a *cone of prints*. We note $\mathcal{F}_{Prints(\mathcal{E},\mathcal{A})}$ the σ-field generated by the set of cones of prints.

Lemma 4.22 (print is an insight function stable by composition). *Let $\mathcal{A}$ be a PSIOA and $\mathcal{E} \in env(\mathcal{A})$. $print_{(\mathcal{E},\mathcal{A})}$ is an insight function stable by composition. a measurable function from $\mathcal{F}_{Execs(\mathcal{E}||\mathcal{A})}$ to $\mathcal{F}_{Prints(\mathcal{E},\mathcal{A})}$.*

Proof. (1) (measurability) We need to show that $\forall G \in \mathcal{F}_{Prints(\mathcal{E},\mathcal{A})}, print^{-1}_{(\mathcal{E},\mathcal{A})}(G) \in \mathcal{F}_{Frags(\mathcal{E}||\mathcal{A})}$.
We note $f_1 : \alpha \in Frags(\mathcal{E}||\mathcal{A}) \mapsto \alpha \upharpoonright \mathcal{E}$ and $f_2 : \alpha \in Frags(\mathcal{E}||\mathcal{A}) \mapsto trace(\alpha)$. We can already remark that (*) $\forall i \in \{1,2\}, \forall(\alpha,\alpha') \in Frags^2(\mathcal{A}), \alpha \leq \alpha' \implies f_i(\alpha) \leq f_i(\alpha')$ and (**) $\forall i \in \{1,2\}, \forall(y_i,\alpha') \in range(f_i) \times Frags(\mathcal{A}), y_i \leq f_i(\alpha') \implies \exists(\alpha,\alpha'') \in Frags^2(\mathcal{A}), \alpha' = \alpha^\frown\alpha''$ and $f_i(\alpha) = y_i$.
Let $G \in \mathcal{F}_{Prints(\mathcal{E},\mathcal{A})}$. By construction, it exists $\zeta = (e,\beta) \in Prints(\mathcal{E},\mathcal{A})$ s. t. $G = C_\zeta$. Let $F \triangleq print^{-1}_{(\mathcal{E},\mathcal{A})}(G) = \{\alpha' \in Frags(\mathcal{A}) | \zeta \leq print_{(\mathcal{E},\mathcal{A})}(\alpha')\} = F_1 \cap F_2$ with $F_1 = \{\alpha'_1 \in Frags(\mathcal{A}) | e \leq f_1(\alpha')\}$ and $F_2 = \{\alpha'_2 \in Frags(\mathcal{A}) | \beta \leq f_2(\alpha'_2)\}$. By (*) and (**), $F_1 = \bigcup_{\alpha_1 \in Frags(\mathcal{A}), \alpha_1 \upharpoonright \mathcal{E} = e} C_{\alpha_1}$ and $F_2 = \bigcup_{\alpha_2 \in Frags(\mathcal{A}), trace(\alpha_2) = \beta}$. By closeness of σ-field under countable union, $F_1, F_2 \in \mathcal{F}_{Frags(\mathcal{E}||\mathcal{A})}$ and by closeness of σ-field under intersection $F \in \mathcal{F}_{Frags(\mathcal{E}||\mathcal{A})}$ which ends the proof.

(2) (stability by composition) Let $(\mathcal{A}_1,\mathcal{A}_2,\mathcal{B},\mathcal{E})$ be a quadruplet of PSIOA, s. t. $\mathcal{B}$ is compatible with $\mathcal{A}_1$ and $\mathcal{A}_2$, $\mathcal{E} \in env(\mathcal{B}||\mathcal{A}_1) \cap env(\mathcal{B}||\mathcal{A}_2)$. Let $(\alpha,\pi) \in Execs_{\mathcal{E}||\mathcal{B}||\mathcal{A}_1} \times Execs_{\mathcal{E}||\mathcal{B}||\mathcal{A}_2}$, clearly $\alpha \upharpoonright (\mathcal{E}||\mathcal{B}) = \pi \upharpoonright (\mathcal{E}||\mathcal{B}) \implies \alpha \upharpoonright \mathcal{E} = \pi \upharpoonright \mathcal{E}$, while the trace component stay the same. Thus, for every $(C_1,C_2) \in \mathcal{F}_{Execs(\mathcal{E}||\mathcal{B}||\mathcal{A}_1)} \times \mathcal{F}_{Execs(\mathcal{E}||\mathcal{B}||\mathcal{A}_2)}$, $print_{(\mathcal{E}||\mathcal{B},\mathcal{A}_1)}(C_1) = print_{(\mathcal{E}||\mathcal{B},\mathcal{A}_2)}(C_2) \implies f_{(\mathcal{E},\mathcal{B}||\mathcal{A}_1)}(C_1) = f_{(\mathcal{E},\mathcal{B}||\mathcal{A}_2)}(C_2)$.

□

Thus, given an environment $\mathcal{E}$ of $\mathcal{A}$ probability measure μ on $\mathcal{F}_{Execs(\mathcal{E}||\mathcal{A})}$, and a scheduler σ of $(\mathcal{E}||\mathcal{A})$ we define $pdist_{(\mathcal{E},\mathcal{A})}(\sigma,\mu) \triangleq print\text{-}dist_{(\mathcal{E},\mathcal{A})}(\sigma,\mu)$, to be the image measure of $\epsilon_{\sigma,\mu}$ under $print_{(\mathcal{E},\mathcal{A})}$. We note $pdist_{(\mathcal{E},\mathcal{A})}(\sigma)$ for $pdist_{(\mathcal{E},\mathcal{A})}(\sigma,\delta_{start(\mathcal{E},\mathcal{A})})$.

This choice that slightly differs from $tdist_{(\mathcal{E},\mathcal{A})}(\sigma,\mu) = trace\text{-}dist_{(\mathcal{E},\mathcal{A})}(\sigma,\mu)$ used in [4], is motivated by the achievement of monotonicity of *print*-implementation w.r.t. PSIOA creation.

4.7 Hiding operator

We anticipate the definition of configuration automata by introducing the classic hiding operator. This operator "hide" the output actions transforming them into internal actions

Definition 4.23 (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 4.24 (hiding on PSIOA). Let $\mathcal{A} = (Q_{\mathcal{A}}, \bar{q}_{\mathcal{A}}, sig(\mathcal{A}), D_{\mathcal{A}})$ be a PSIOA. Let h a function mapping each state $q \in Q$ to a set of output actions. We note $hide(\mathcal{A}, h)$ the PSIOA $(Q, \bar{q}, sig'(\mathcal{A}), D)$, where $sig'(\mathcal{A}) : q \in Q \mapsto hide(sig(\mathcal{A})(q), h(q))$.

Lemma 4.25 (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 1. *We can restrict hiding operation to set of actions included 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).*

4.8 State renaming operator

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

Definition 4.26. (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 $ren : Q_{\mathcal{A}} \to Q_{\mathcal{A}'}$ be a bijective mapping. Then $ren(\mathcal{A})$ (we abuse the notation) is the automaton given by:

- $start(ren(\mathcal{A})) = ren(start(\mathcal{A}))$
- $states(ren(\mathcal{A})) = ren(states(\mathcal{A}))$
- $\forall q_{\mathcal{A}'} \in states(ren(\mathcal{A})), sig(ren(\mathcal{A}))(q_{\mathcal{A}'}) = sig(\mathcal{A})(ren^{-1}(q_{\mathcal{A}'}))$
- $\forall q_{\mathcal{A}'} \in states(ren(\mathcal{A})), \forall a \in sig(ren(\mathcal{A}))(q_{\mathcal{A}'})$, 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 4.27. (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 4.28. *Let $\mathcal{A}$ and $\mathcal{A}'$ be two PSIOA s. t. $\mathcal{A}' = ren(\mathcal{A})$ with $ren : states(\mathcal{A}) \to states(\mathcal{A}')$ being a bijective map. 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 $\eta_{(\mathcal{A}', ren(q^j), a^{j+1})}(ren(q^{j+1})) = \eta_{(\mathcal{A}, q^j, a^{j+1})}(q^{j+1}) > 0$. □

5 PROBABILISTIC CONFIGURATION AUTOMATA

We combine the notion of configuration of [1] with the probabilistic setting of [8]. A configuration is a set of automata attached with their current states. This will be a very useful tool to define dynamicity by mapping the state of an automaton of a certain "layer" to a configuration of automata of lower layer, where the set of automata in the configuration can dynamically change from on state of the automaton of the upper level to another one.

5.1 configuration

Definition 5.1 (Configuration). A configuration is a pair $(\mathbf{A}, \mathbf{S})$ where

- $\mathbf{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ is a finite set of PSIOA identifiers 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, there is a subtlety, since it captures a set of some automata ($\mathbf{A}$) in their current state ($\mathbf{S}$), but the set of automata of the systems will not be fixed in the time.

Definition 5.2 (Compatible configuration). A configuration $(\mathbf{A}, \mathbf{S})$, with $\mathbf{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$, is compatible iff the set $\mathbf{A}$ is compatible at state $(\mathbf{S}(\mathcal{A}_1), ..., \mathbf{S}(\mathcal{A}_n))$ as per definition 4.6

Definition 5.3 (Intrinsic attributes of a configuration). Let $C = (\mathbf{A}, \mathbf{S})$ be a compatible 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 actions of the configuration,
- $in(C) = (\bigcup_{\mathcal{A} \in \mathbf{A}} in(\mathcal{A})(\mathbf{S}(\mathcal{A}))) - out(C)$ represents the input actions of the configuration,
- $int(C) = \bigcup_{\mathcal{A} \in \mathbf{A}} int(\mathcal{A})(\mathbf{S}(\mathcal{A}))$ represents the internal actions of the configuration,
- $ext(C) = in(C) \cup out(C)$ represents the external actions of the configuration,
- $sig(C) = (in(C), out(C), int(C))$ is called the intrinsic signature of the configuration,
- $US(C) = (\mathbf{S}(\mathcal{A}_1), ..., \mathbf{S}(\mathcal{A}_n))$ represents the states of the set of 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 be used later to capture the idea of destruction of an automaton.

Definition 5.4 (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 $C = reduce(C)$.

5.2 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. These distributions belong to the measurable set $(Q_{conf}, Disc(Q_{conf}))$ where Q_{conf} denotes the (countable) set of configurations.

Lemma 5.5. *The set Q_{conf} of configurations is countable.*

Proof. (1) $\{\mathbf{A} \in \mathcal{P}(Autids) | \mathbf{A}$ is finite $\}$ is countable, (2) $\forall \mathcal{A} \in Autids, states(\mathcal{A})$ is countable by definition 4.1 of PSIOA and (3) the cartesian product of countable sets is a countable set. □

Definition 5.6 (Preserving distribution). A *preserving distribution* $\eta_p \in Disc(Q_{conf})$ s. t. it exists a set of automata $\mathbf{A}$, called *automata support of* η_p, such that $\forall (\mathbf{A}', \mathbf{S}') \in supp(\eta_p), \mathbf{A} = \mathbf{A}'$.

We define a companion distribution as the natural distribution of the corresponding set 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 5.7 (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 automata 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 $\mathbf{S}'' : \mathcal{A}_i \in \mathbf{A} \mapsto q''_i \in Q_{\mathcal{A}_i}$,

if for every configuration C'', for every $q' \in Q_{\mathbf{A}}$, $US(C'') = q' \implies \eta_{(\mathbf{A},q,a)}(q') = \eta_p(C'')$.

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. So η_p can have several companion distributions.

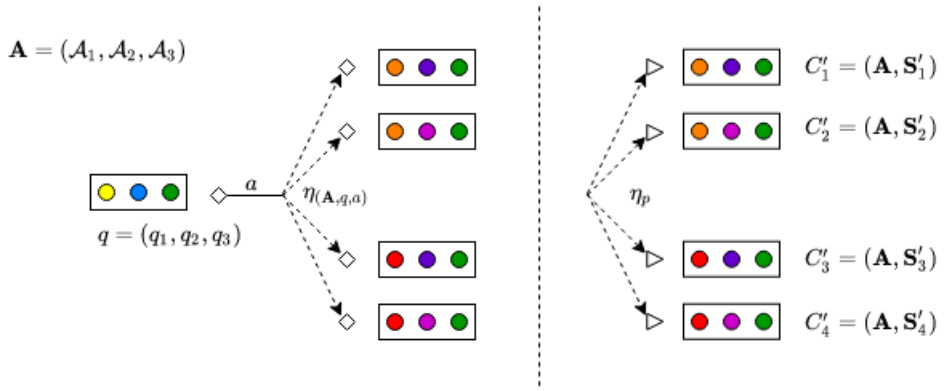


Fig. 14. A preserving distribution is matching its companion distribution.

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 5.8 (preserving transition). Let $C = (\mathbf{A}, \mathbf{S})$ be a compatible configuration, $q = US(C)$ and $\eta_p \in Disc(Q_{conf})$ be a preserving transition with $\mathbf{A}_s$ as automata support.

Then we 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 5.9 (reduced distribution). A *reduced* distribution $\eta_r \in Disc(Q_{conf})$ 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 set of new automata that are created.

Definition 5.10 (Generation of reduced distribution). Let $\eta_p \in Disc(Q_{conf})$ be a preserving distribution with $\mathbf{A}$ as automata support. Let $\varphi \subset Autids$, φ is finite. 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)
  $\forall (\mathbf{A}'', \mathbf{S}'') \in supp(\eta_{nr}), \mathbf{A}'' = \mathbf{A} \cup \varphi$.
- (freshly created automata start at start state) $\forall C'' = (\mathbf{A}'', \mathbf{S}'') \in supp(\eta_{nr}), \forall \mathcal{A}_i \in \varphi \setminus \mathbf{A}, \mathbf{S}''(\mathcal{A}_i) = start(\mathcal{A}_i)$
- (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 $\forall \mathcal{A}_i \in \varphi \setminus \mathbf{A}, \mathbf{S}''(\mathcal{A}_i) = start(\mathcal{A}_i)$, $\eta_{nr}((\mathbf{A}'', \mathbf{S}'')) = \eta_p(\mathbf{A}, \mathbf{S}''\lceil \mathbf{A}))$
  where $\mathbf{S}''\lceil \mathbf{A})$ denotes the restriction of $\mathbf{S}''$ on $\mathbf{A}$)
- (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 5.11 (Intrinsic transition). Let $(\mathbf{A}, \mathbf{S})$ be arbitrary reduced compatible configuration, let $\eta \in Disc(Q_{conf})$, and let $\varphi \subseteq Autids$, $\varphi \cap \mathbf{A} = \emptyset$ and φ is finite. Then $\langle \mathbf{A}, \mathbf{S} \rangle \overset{a}{\Longrightarrow}_{\varphi} \eta$ if η is generated by η_p and φ with $(\mathbf{A}, \mathbf{S}) \overset{a}{\rightarrow} \eta_p$.

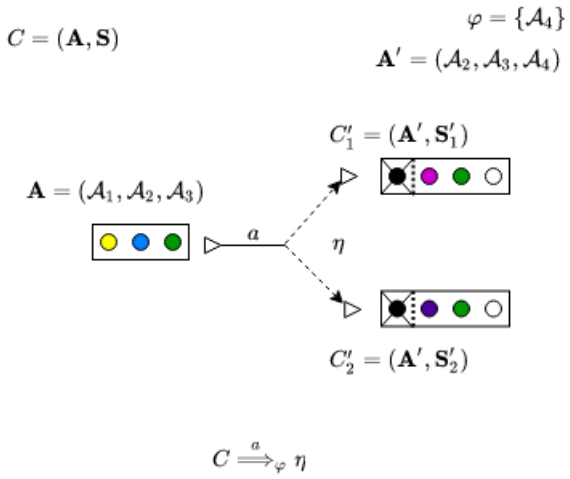


Fig. 15. An intrinsic transition where $\mathcal{A}_1$ is destroyed deterministically and $\mathcal{A}_4$ is created deterministically.

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.

5.3 Probabilistic Configuration Automata

Here we define our probabilistic configuration automata. Just before that, we introduce a notation to represents corresponding probability measures whose respective supports are linked by a bijection that preserves the measure.

Definition 5.12 ($\eta \overset{f}{\leftrightarrow} \eta'$). Let Q and Q' be two countable sets. Let $(\eta, \eta') \in Disc(Q) \times Disc(Q')$. Let $f : Q \rightarrow Q'$. We note $\eta \overset{f}{\leftrightarrow} \eta'$ if the following is verified:

- the restriction $\tilde{f}$ of f to $supp(\eta)$ is a bijection from $supp(\eta)$ to $supp(\eta')$
- $\forall q \in supp(\eta), \eta(q) = \eta'(f(q))$

Lemma 5.13. *$\eta \overset{f}{\leftrightarrow} \eta'$ and $\eta' \overset{g}{\leftrightarrow} \eta''$ implies*

- *$\eta \overset{h}{\leftrightarrow} \eta''$ where the restriction $\tilde{h}$ of h on $supp(\eta)$ verifies $\tilde{h} = \tilde{g} \circ \tilde{f}$ and*
- *$\eta' \overset{k}{\leftrightarrow} \eta$ where the restriction $\tilde{k}$ to $supp(\eta')$ verifies $\tilde{k} = \tilde{f}^{-1}$*

Proof. For the first item: The composition of two bijection is a bijection and the reverse function of a bijection is a bijection. For the second item: In the first case, $\forall q \in supp(\eta), \eta(q) = \eta'(f(q))$ with $f(q) \in supp(\eta')$ which means $\eta'(f(q)) = \eta''(g(f(q)))$. In the second case $\forall q' \in supp(\eta'), \exists! q \in supp(\eta), \eta(q) = \eta'(q' = \tilde{f}(q))$ and hence $\forall q' \in supp(\eta'), \eta'(q') = \eta(q = \tilde{f}^{-1}(q'))$. □

Now we are ready to define our probabilistic configuration automata. Such an automaton define a strong link with a dynamic configuration.

Definition 5.14 (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 $config(K)(q_K)$ is a reduced compatible configuration for all $q_K \in states(K)$.
- 3. For each $q_K \in states(K)$, a mapping $created(K)(q_K)$ with domain $sig(K)(q_K)$ and such that $\forall a \in sig(K)(q_K)$, $created(K)(q_K)(a) \subseteq Autids$ with $created(K)(q_K)(a)$ finite.
- 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. (start states preservation) 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. (top/down transition preservation) If $(q_K, a, \eta_{(K,q_K,a)}) \in dtrans(K)$ then it exists $\eta' \in Disc(Q_{conf})$ s. t. $\eta_{(K,q_K,a)} \overset{f}{\longleftrightarrow} \eta'$ with i) $f = config(K)$ and ii) $config(K)(q_K) \overset{a}{\Longrightarrow}_{\varphi} \eta'$, where $\varphi = created(K)(q_K)(a)$
- 3. (bottom/up transition preservation) If $q_K \in states(K)$ and $config(K)(q_K) \overset{a}{\Longrightarrow}_{\varphi} \eta'$ for some action a, $\varphi = created(K)(x)(a)$, and reduced compatible probabilistic measure $\eta' \in Disc(Q_{conf})$, then $(q_K, a, \eta_{(K,q_K,a)}) \in dtrans(K)$, and $\eta_{(K,q_K,a)} \overset{f}{\longleftrightarrow} \eta'$ with $f = config(K)$.
- 4. (signature preservation modulo hiding) 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) $out(K)(q_K) \cup int(X)(q_K) = out(config(K)(q_K)) \cup int(config(K)(q_K))$

This definition, proposed in a deterministic fashion in [1], captures dynamicity of the system. Each state is linked with a configuration. The set of automata of the configuration can change during an execution. A sub-automaton $\mathcal{A}$

is created from state q by the action a if $\mathcal{A} \in created(K)(q)(a)$. A sub-automaton $\mathcal{A}$ is destroyed if the non-reduced attached configuration distribution lead to a configuration where $\mathcal{A}$ is in a state $q^{\phi}_{\mathcal{A}}$ s. t. $\widehat{sig}(\mathcal{A})(q^{\phi}_{\mathcal{A}}) = \emptyset$. Then the corresponding reduced configuration will not hold $\mathcal{A}$. The last constraint 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.

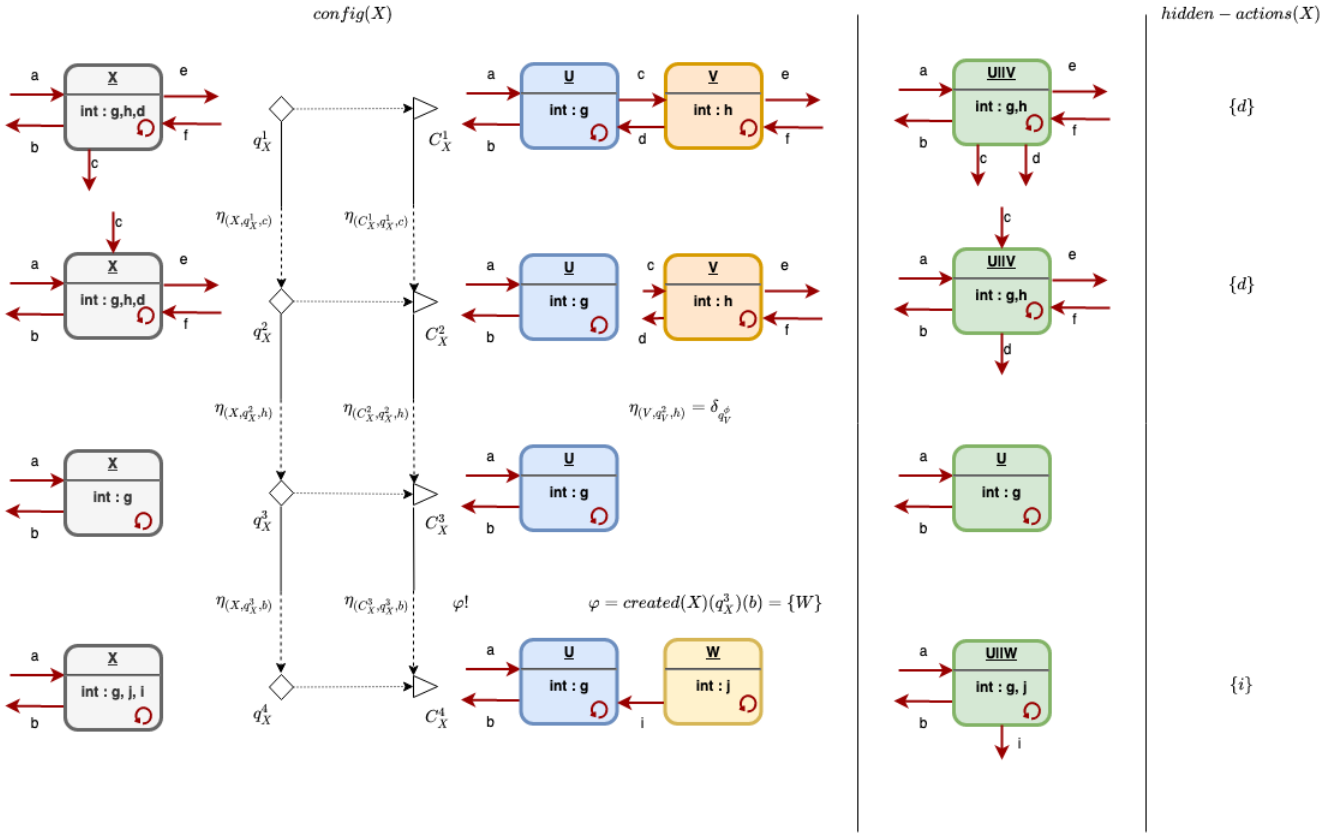


Fig. 16. A PCA life cycle.

Definition 5.15 (hiding on PCA). Let X be a PCA. Let $h : q \in states(X) \mapsto h(q) \subset out(X)(q)$ a function mapping each state $q \in states(X)$ to a set of output actions. We note $hide(X, h)$ the PCA X' that differs from X only on $sig(X')$ and $hidden\text{-}actions(X')$, where $\forall q \in states(X) = states(X')$,

- $sig(X')(q) = hide(sig(X)(q), h(q))$ and
- $hidden\text{-}actions(X')(q) = hidden\text{-}actions(X)(q) \cup h(q)$.

Additionally, we recursively define the current constitution of a PCA. To do so, we assume the existence of a subset $Autids_0 \subset Autids$ that represents the "atomic entities" of our formalism. Any automaton is constructed with automata in $Autids_0$.

Definition 5.16 (Constitution). For every $\mathcal{A} \in Autids$, we note $constitution(\mathcal{A}) : states(\mathcal{A}) \rightarrow \mathcal{P}(Autids_0)$ s. t.

- $\forall \mathcal{A} \in Autids_0, \forall q \in states(\mathcal{A}), constitution(\mathcal{A})(q) = \{\mathcal{A}\}$.
- $\forall \mathbf{A} = (\mathcal{A}_1, ..., \mathcal{A}_n) \in (Autids_0)^n, \forall q \in states(\mathcal{A})$ with $\mathcal{A} = \mathcal{A}_1||...||\mathcal{A}_n$, $constitution(\mathcal{A})(q) = \mathbf{A}$.
- the constitution of a PCA is defined recursively through its configuration, i.e. for every PCA X, $\forall q \in states(X)$, if we note $(\mathbf{A}, \mathbf{S}) = config(X)(q)$, $constitution(X)(q) = \bigcup_{\mathcal{A} \in \mathbf{A}} constitution(\mathcal{A})(\mathbf{S}(\mathcal{A}))$.

We note $UA(K) = \bigcup_{q \in states(K)}(constitution(K)(q))$.

5.4 Compatibility, composition

Again, we want to define composition operator for PCA to captures the fact the comportment of one PCA can influence the comportment of another PCA. Some syntaxic rules have to be defined first.

Compatibility and union of configuration. We extends the formalism of configuration with the union operator. We do not need a composition operator to prevent from think such an union is always compatible, which is not true.

Definition 5.17 (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 $\mathbf{A} = \mathbf{A}_1 \cup \mathbf{A}_2$ 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 $\forall i, j \in [1 : n], i \neq j, auts(C_i) \cap auts(C_j) = \emptyset$.

Lemma 5.18. *Let $C_1 = (\mathbf{A}_1, \mathbf{S}_1)$ and $C_2 = (\mathbf{A}_2, \mathbf{S}_2)$ be compatible configurations such that $\mathbf{A}_1 \cap \mathbf{A}_2 = \emptyset$. Let $C = (\mathbf{A}, \mathbf{S}) = C_1 \cup C_2$ be a compatible configuration. Then $sig(C) = sig(C_1) \times sig(C_2)$*

Proof.
- $out(C) = \bigcup_{\mathcal{A}_k \in \mathbf{A}} out(\mathcal{A}_k)(\mathbf{S}(\mathcal{A}_k)) = (\bigcup_{\mathcal{A}_i \in \mathbf{A}_1} out(\mathcal{A}_i)(\mathbf{S}(\mathcal{A}_i))) \cup (\bigcup_{\mathcal{A}_j \in \mathbf{A}_2} out(\mathcal{A}_j)(\mathbf{S}(\mathcal{A}_j))) = (\bigcup_{\mathcal{A}_i \in \mathbf{A}_1} out(\mathcal{A}_i)(\mathbf{S}_1(\mathcal{A}_i))) \cup (\bigcup_{\mathcal{A}_j \in \mathbf{A}_2} out(\mathcal{A}_j)(\mathbf{S}_2(\mathcal{A}_j))) = out(C_1) \cup out(C_2)$
- $in(C) = \bigcup_{\mathcal{A}_k \in \mathbf{A}} in(\mathcal{A}_k)(\mathbf{S}(\mathcal{A}_k)) \setminus out(C) = (\bigcup_{\mathcal{A}_i \in \mathbf{A}_1} in(\mathcal{A}_i)(\mathbf{S}(\mathcal{A}_i))) \cup (\bigcup_{\mathcal{A}_j \in \mathbf{A}_2} in(\mathcal{A}_j)(\mathbf{S}(\mathcal{A}_j))) \setminus out(C) = (\bigcup_{\mathcal{A}_i \in \mathbf{A}_1} in(\mathcal{A}_i)(\mathbf{S}_1(\mathcal{A}_i))) \cup (\bigcup_{\mathcal{A}_j \in \mathbf{A}_2} in(\mathcal{A}_j)(\mathbf{S}_2(\mathcal{A}_j))) \setminus out(C) = in(C_1) \cup in(C_2) \setminus (out(C_1) \cup out(C_2))$
- $int(C) = \bigcup_{\mathcal{A}_k \in \mathbf{A}} int(\mathcal{A}_k)(\mathbf{S}(\mathcal{A}_k)) = (\bigcup_{\mathcal{A}_i \in \mathbf{A}_1} int(\mathcal{A}_i)(\mathbf{S}(\mathcal{A}_i))) \cup (\bigcup_{\mathcal{A}_j \in \mathbf{A}_2} int(\mathcal{A}_j)(\mathbf{S}(\mathcal{A}_j))) = (\bigcup_{\mathcal{A}_i \in \mathbf{A}_1} int(\mathcal{A}_i)(\mathbf{S}_1(\mathcal{A}_i))) \cup (\bigcup_{\mathcal{A}_j \in \mathbf{A}_2} int(\mathcal{A}_j)(\mathbf{S}_2(\mathcal{A}_j))) = int(C_1) \cup int(C_2)$ Thus $(out(C), in(C), int(C)) = (out(C_1) \cup out(C_2), in(C_1) \cup in(C_2) \setminus (out(C_1) \cup out(C_2)), int(C_1) \cup int(C_2))$

□

Definition 5.19 (PCA partially-compatible at a state). Let $\mathbf{X} = \{X_1, ..., X_n\}$ be a set 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) Sub-automaton exclusivity: $\forall i, j \in [1 : n], i \neq j : auts(config(X_i)(q_{X_i})) \cap auts(config(X_j)(q_{X_j})) = \emptyset$.
(2) Compatible signatures $\{sig(X_1)(q_{X_1}), ..., sig(X_n)(q_{X_n})\}$ is a set of compatible signatures.
(3) Creation exclusivity: $\forall i, j \in [1 : n], i \neq j : \forall a \in \widehat{sig}(X_i)(q_{X_i}) \cap \widehat{sig}(X_j)(q_{X_j}) : created(X_i)(q_{X_i})(a) \cap created(X_j)(q_{X_j})(a) = \emptyset$.
(4) Constitution exclusivity: $\forall i, j \in [1 : n], i \neq j : constitution(X_i)(q_{X_i}) \cap constitution(X_j)(q_{X_j}) = \emptyset$

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 5.20 (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 $psioa(\mathbf{X})$, 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 5.21 (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.

Now, we are able to define our composition operator.

Definition 5.22 (Composition of configuration automata). Let $X_1, ..., X_n$, partially-compatible PCA. Then $X = X_1||...||X_n$ is the state machine consisting of the following components:

(1) $psioa(X) = psioa(X_1)||...||psioa(X_n)$
(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 $states(X) = states(psioa(X)), start(X) = start(psioa(X)), sig(X) = sig(psioa(X)), steps(X) = steps(psioa(X))$, and likewise for all other (sub)components and attributes of $psioa(X)$.

We want to show that the set of PCA is closed under composition. Before starting the proof, we introduce some tools.

Lemma 5.23 (Joint preserving probability distribution for union of configuration). *Let $\{C_k = (\mathbf{A}_k, \mathbf{S}_k)\}_{k \in [1:n]}$ be a finite set of compatible configurations s. t. $\forall k, \ell \in [1:n], \mathbf{A}_k \cap \mathbf{A}_\ell = \emptyset$. Let $C = (\mathbf{A}, \mathbf{S}) = \bigcup_{k \in [1:n]} C_k$ be a compatible configuration. Let $a \in \widehat{sig}(C)$. Let $(\mathcal{I}, \mathcal{J})$ be a partition of $[1:n]$ s. t. for every $i \in \mathcal{I}, a \in \widehat{sig}(C_i)$ and for every $j \in \mathcal{J}, a \in \widehat{sig}(C_j)$. For every $i \in \mathcal{I}$, let η_p^i be the unique preserving distributions that has $\eta_{(\mathbf{A}_i,q_i,a)}$ as companion distribution with $q_i = US(C_i)$. For every $j \in \mathcal{J}$, let $\eta_p^j = \delta_{C_j}$. We note η_p the unique preserving distributions that have $\eta_{(\mathbf{A},q,a)}$ as companion distribution with $q = US(C)$.*

Then, $C \xrightarrow{a} \eta_p$, s. t. for every configuration $C' = (\mathbf{A}, \mathbf{S}')$, for every (unique) finite set of configurations $\{C'_k = (\mathbf{A}_k, \mathbf{S}'_k)\}_{k \in [1:n]}$ verifying $C' = \bigcup_{k \in [1:n]} C'_k$, we have $\eta_p(C') = (\eta_p^1 \otimes ... \otimes \eta_p^n)((C'_1, ..., C'_n))$.

Proof. Since $\mathbf{A} = \bigcup_{k \in [1:n]} \mathbf{A}_k$ and $\mathbf{S}$ agrees with $\mathbf{S}_k$ on $\mathcal{A} \in \mathbf{A}_k$ for every $k \in [1:n]$, we have $\eta_{\mathbf{A},q,a} = \eta_{\mathbf{A}_1,q_1,a} \otimes ... \otimes \eta_{\mathbf{A}_n,q_n,a}$ with the convention $\eta_{\mathbf{A}_j,q_j,a} = \delta_{q_j}$ for every $j \in \mathcal{J}$. Furthermore, for every $k \in [1,n], \eta_{\mathbf{A}_k,q_k,a}$ is a companion distribution of η_p^k, that is for every C'_k, $q'_k = US(C'_k)$, $\eta_p^k(C'_k) = \eta_{\mathbf{A}_k,q_k,a}(q'_k)$. Hence for every $(C'_1, ..., C'_n)$, $(q'_1 = US(C'_1), ..., q'_n = US(C'_n)), \eta_{\mathbf{A},q,a}((q'_1, ..., q'_n)) = (\eta_{\mathbf{A}_1,q_1,a} \otimes ... \otimes \eta_{\mathbf{A}_n,q_n,a})((q'_1, ..., q'_n)) = (\eta_p^1 \otimes ... \otimes \eta_p^n((C'_1, ..., C'_n))$ (*). By definition of η_p, for every C', $(q' = US(C'))$, $\eta_{\mathbf{A},q,a}(q') = \eta_p(C')$. Since we deal with preserving distribution and $\mathbf{A} = \bigcup_{k \in [1:n]} \mathbf{A}_k$, we have each element $q' = US(C') \in Q_{\mathbf{A}}$ is of the form $(q'_1, ..., q'_n)$ with $q'_k \in Q_{\mathbf{A}_k}$ and verifies $C' = C_1 \cup ... \cup C'_n$ with $auts(C'_k) = \mathbf{A}_k$ and $US(C'_k) = q'_k$. (**) Hence we compose (*) and (**) to obtain for every configuration $C' = (\mathbf{A}, \mathbf{S}')$, for every finite set of configurations $\{C'_k = (\mathbf{A}_k, \mathbf{S}'_k)\}_{k \in [1:n]}$, s. t. $C' = \bigcup_{k \in [1:n]} C'_k$, then

$\eta_p(C') = (\eta_p^1 \otimes ... \otimes \eta_p^n)((C'_1, ..., C'_n))$. Since $a \in \widehat{sig}(C)$ and the automata support of $(\eta_p^1 \otimes ... \otimes \eta_p^n)$ is $\mathbf{A} = \bigcup_{k\in[1:n]} \mathbf{A}_k$, we can note $C \xrightarrow{a} \eta_p$

□

Definition 5.24 (merge, join). Let $\tilde{\eta} = (\eta_1, ...\eta_n) \in Disc(Q_{conf})^n$. We define

- $join(\tilde{\eta}) : (C_1, ..., C_n) \in Q_{conf}^n \mapsto (\eta_1 \otimes ... \otimes \eta_n)(C_1, ..., C_n)$ and
- $merge(\tilde{\eta}) : C \in Q_{conf} \mapsto \sum_{(C_1^k,...,C_n^k)\in Q_{conf}^n} join(\tilde{\eta})((C_1^k, ..., C_n^k)) \cdot \mathbb{1}_{(C_1^k \cup ... \cup C_n^k)=C}$

Definition 5.25 (deter-dest, base). Let $C^s = (\mathbf{A}^s, \mathbf{S}^s)$ be a configuration. For every $\mathcal{A} \in \mathbf{A}^s$, we note $q^s_{\mathcal{A}} = \mathbf{S}^s(\mathcal{A})$. Let $\varphi \in \mathcal{P}(Autids)$. We define

- $deter\text{-}dest(C^s, a) = \{\mathcal{A} \in \mathbf{A}^s | a \in \widehat{sig}(\mathcal{A})(q^s_{\mathcal{A}}) \wedge \eta_{\mathcal{A},q^s_{\mathcal{A}},a} = \delta_{q^\phi_{\mathcal{A}}}\}$.
- $base(C^s, a, \varphi) = \mathbf{A}^s \cup \varphi \setminus deter\text{-}dest(C^s, a)$.

Lemma 5.26 (Merging). *Let $\{C_1^s, ..., C_n^s\}$ be a set of compatible configuration with for every $i \in [1, n]$, $C_i^s = (\mathbf{A}_i^s, \mathbf{S}_i^s)$. Let $C^s = (\mathbf{A}^s, \mathbf{S}^s) = \bigcup_{i\in[1:n]} C_i^s$. For every $\mathcal{A} \in \mathbf{A}^s$, we note $q^s_{\mathcal{A}} = \mathbf{S}^s(\mathcal{A})$ Let $a \in \widehat{sig}(C^s)$. Let $(\mathcal{I}, \mathcal{J})$ be a partition of $[1, n]$ s. t. $\forall i \in \mathcal{I}$, $a \in \widehat{sig}(C_i^s)$ and $\forall j \in \mathcal{J}$, $a \notin \widehat{sig}(C_j^s)$. Let $(\varphi_1, ..., \varphi_n) \in \mathcal{P}(Autids)^n$, s. t. $\forall j \in J, \varphi_j = \emptyset$ and let us note $\varphi = \bigcup_{i\in\mathcal{I}} \varphi_i$. For every $i \in \mathcal{I}$, we note η_i s. t. $C_i^s \xRightarrow{a}_{\varphi_i} \eta_i$ For every $j \in \mathcal{J}$, we note η_j s. t. $\eta_j = \delta_{C_j^s}$.*

Let $\tilde{\eta} = (\eta_1, ..., \eta_n)$. Let $\eta^{join} = join(\tilde{\eta})$, $\eta^{merge} = merge(\tilde{\eta})$ and let assume, for every $C^f \in supp(\eta^{merge}))$, either (*) C^f is compatible or (**) $\forall k, \ell \in [1 : n], base(C_k^s, a, \varphi_k) \cap base(C_\ell^s, a, \varphi_\ell) = \emptyset$. Then, it means

1. *$\forall C^f \in supp(\eta^{merge})$, it exists a unique $(C_1^f, ..., C_n^f)$ s. t. 1) $C^f = C_1^f \cup ... \cup C_n^f$ and 2) $\forall k \in [1, n], C_k^f \in supp(\eta_k)$. We note $C^f.split(\tilde{\eta})$ this unique $(C_1^f, ..., C_n^f)$.*
2. *$g^{\tilde{\eta}} : C \in supp(merge(\tilde{\eta})) \mapsto C.split(\tilde{\eta}) \in supp(\eta_1) \times ... \times supp(\eta_n)$ is a bijection.*
3. *Then $merge(\tilde{\eta})(C) = join(\tilde{\eta})(C.split(\tilde{\eta}))$*
4. *$merge(\tilde{\eta}) \overset{g^{\tilde{\eta}}}{\leftrightarrow} join(\tilde{\eta})$*
5. *$C^s \xRightarrow{a}_{\varphi} merge(\tilde{\eta})$*

Proof. First we show that (*) implies (**). By contradiction. Let assume $\mathcal{A} \in base(C_k^s, a, \varphi_k) \cap base(C_\ell^s, a, \varphi_\ell)$. Since C^s is compatible, $\mathcal{A} \notin \mathbf{A}_k^s \cap \mathbf{A}_\ell^s$. By definition of the base it exists $C_k^f, C_\ell^f \in supp(\eta_k) \times supp(\eta_\ell)$, $\mathcal{A} \in auts(C_k^f) \cap auts(C_\ell^f)$ and $C_k^f \cup C_\ell^f$ is not compatible. So it exists $(C_1^f, ..., C_n^f) \in supp(\eta_1 \otimes ... \otimes \eta_n)$ s. t. $(C_1^f \cup ... \cup C_n^f)$ is not compatible.

1. The uniqueness comes from (**). Indeed, let imagine two candidates $(C_1^f, ..., C_n^f)$ and $(C_1^{f\prime}, ..., C_n^{f\prime})$ if $\mathcal{A} \in auts(C_k^f)$, then $\mathcal{A} \in auts(C_k)$ but $\mathcal{A} \notin base(C_{k'}^{s\prime})$ for any $k' \neq k$ and so $\mathcal{A} \in auts(C_k^{f\prime})$ and so $auts(C_k^f) = auts(C_i^{f\prime})$ for every $k \in [1, n]$. Now we need to have $map(C_k^f) = map(C_i^{f\prime})$ for every $k \in [1, n]$ to obtain $C^f = C_1^f \cup ... \cup C_n^f = C_1^{f'} \cup ... \cup C_n^{f'}$. The existence is by construction of $join$.
2. The bijection comes from the existence and the uniqueness of pre-image
3. By previous item 1, the sum of merge has only one element which consists of $join(\tilde{\eta})(C.split(\tilde{\eta}))$
4. By previous two items 1 and 2, that gives the definition
5. η'_k is generated by φ_k and preserving distribution $\eta_k'^p$ where $\mathbf{A}_k^p$ is the automata support of $\eta_k'^p$. By compatibility of C^s, for every $k, \ell \in [1, n], k \neq \ell$, $\mathbf{A}_k^p \cap \mathbf{A}_\ell^p = \emptyset$. Hence, we can apply lemma 5.23 and we have $C^s \xrightarrow{a} merge((\eta_1'^p, ..., \eta_n'^p))$. Moreover $merge((\eta'_1, ..., \eta'_n))$ is generated by $merge((\eta_1'^p, ..., \eta_n'^p))$ and φ since each $\eta_k'^p$ is generated by η'_k and φ_k. Then $C^s \xRightarrow{a}_{\varphi} merge((\eta'_1, ..., \eta'_n))$

□

Lemma 5.27 ($\overset{f}{\leftrightarrow}$ preservation for joint probability). *Let $\tilde{\eta}' = (\eta'_1, ..., \eta'_n)$. Let $(\eta_1, ..., \eta_n)$ and $(f_1, ..., f_n)$ s. t. $\forall i \in [1, n], \eta_i \overset{f_i}{\leftrightarrow} \eta'_i$.*

Then $\eta_1 \otimes ... \otimes \eta_n \overset{f}{\leftrightarrow} \eta'_1 \otimes ... \otimes \eta'_n$ With $f : x = (x_1, ..., x_n) \mapsto y = (f_1(x_1), ..., f_n(x_n))$.

Proof. f is still a bijection and $(\eta_1 \otimes ... \otimes \eta_n)(z_1, ..., z_n) = \eta_1(z_1) \cdot ...\eta_n(z_n) = \eta'_1(f_1(z_1)) \cdot ...\eta'_n(f_n(z_n)) = (\eta_1 \otimes ... \otimes \eta_n)(f_1(z_1), ..., f_n(z_n)) = (\eta_1 \otimes ... \otimes \eta_n)(f(z_1, ..., z_n))$ □

Now we are ready for the theorem that claims that a composition of PCA is a PCA.

Theorem 5.28 (PCA closeness under composition). *Let $X_1, ..., X_n$, be 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 5.14.

- (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 $(q, a, \eta_{X,q,a}) \in dtrans(X)$. We will establish $\exists \eta' \in Disc(Q_{conf})$ s. t. $\eta_{X,q,a} \overset{f}{\leftrightarrow} \eta'$ where $f = config(X)$ and $config(X)(q) \overset{a}{\Longrightarrow}_\varphi \eta'$ with $\varphi = created(X)(q)(a)$.
  For brevity, let $\mathcal{A}_i = psioa(X_i)$ for $i \in [1 : n]$. By definition of pca-compositon 5.22, $psioa(X) = psioa(X_1)||...||psioa(X_n) = P_1||...||P_n$. We note $\mathbf{P} = (P_1, ..., P_n)$ and then by definition of psioa-composition 4.10, $q = (q_1, ..., q_n) \in Q_{P_1} \times ... \times Q_{P_n}$, while $a \in \bigcup_{i \in [1:n]} \widehat{sig}(P_i)(q_i)$ and $\eta_{X,q,a} = \eta_{P_1,q_1,a} \otimes ... \otimes \eta_{P_n,q_n,a}$ with the convention $\eta_{P_i,q_i,a} = \delta_{q_i}$ if $a \notin \widehat{sig}(P_i)(q_i)$.
  Let $(\mathcal{I}, \mathcal{J})$ be a partition $[1 : n]$ s. t.
  For every $i \in \mathcal{I}$, $a \in \widehat{sig}(P_i)(q_i)$, and then by PCA top/down transition preservation, it exists $\eta'_i \in Disc(Q_{conf})$ s. t. $\eta_{X_i,q_i,a} = \eta_{P_i,q_i,a} \overset{f_i}{\leftrightarrow} \eta'_i$ with $f_i = config(X_i)$ and $config(X_i)(q_i) \overset{a}{\Longrightarrow}_{\varphi_i} \eta'_i$ with $\varphi_i = created(X_i)(q_i)(a)$.
  For every $j \in \mathcal{J}$, $a \notin \widehat{sig}(P_i)(q_i)$, then we note $\varphi_j = \emptyset$ and $\eta'_j = \delta_{config(X_j)(q_j)}$ that verifies $\delta_{q_j} \overset{f_j}{\leftrightarrow} \eta'_j$ with $f_j = config(X_j)$.
  We note $\tilde{\eta}' = (\eta'_1, ..., \eta'_n)$ We note $\eta' = join(\tilde{\eta}') = \eta'_1 \otimes ... \otimes \eta'_n$ and $\varphi = \bigcup_{i \in [1:n]} \varphi_i$. By definition of PCA-composition, $\varphi = created(X)(q)(a)$.
  We have $\eta_{X,q,a} \overset{f}{\leftrightarrow} \eta'$ with $f : q = (q_1, ..., q_n) \mapsto (f_1(q_1), ..., f_n(q_n))$ by lemma 5.27.
  Moreover $merge(\tilde{\eta}') \overset{g^{\tilde{\eta}}}{\leftrightarrow} join(\tilde{\eta}')$ with $g^{\tilde{\eta}} : C \in supp(merge(\tilde{\eta})) \mapsto C.split(\tilde{\eta}) \in supp(\eta_1) \times ... \times supp(\eta_n)$ by lemma 5.26, item 4.
  So $\eta_{X,q,a} \overset{h}{\leftrightarrow} merge(\tilde{\eta}')$ with $h = (g^{\tilde{\eta}})^{-1} \circ f = config(X)$.
  Moreover we have $config(X)(q) \overset{a}{\Longrightarrow}_\varphi merge(\tilde{\eta}')$ by lemma 5.26, item 5.
- (Constraint 3) Let $q \in state(X)$, $C = config(X)(q)$, $a \in \widehat{sig}(X)(q)$, $\varphi = created(X)(q)(a)$ that verify $C \overset{a}{\Longrightarrow}_\varphi \eta'$.
  We need to show that it exists $(q, a, \eta_{(X,q,a)}) \in dtrans(X)$ s. t. $\eta_{(X,q,a)} \overset{f}{\leftrightarrow} \eta'$ with $f = config(X)$.

For brevity, let $\mathcal{A}_i = psioa(X_i)$ for $i \in [1:n]$. By definition of pca-compositon 5.22 $psioa(X) = psioa(X_1)||...||psioa(X_n) = P_1||...||P_n$. We note $\mathbf{P} = (P_1, ..., P_n)$ and then by definition of psioa-composition, $q = (q_1, ..., q_n) \in Q_{P_1} \times ... \times Q_{P_n}$, while $a \in \bigcup_{i\in[1:n]} \widehat{sig}(P_i)(q_i)$.

Let $(\mathcal{I}, \mathcal{J})$ be a partition $[1:n]$ s. t.

For every $i \in \mathcal{I}$, $a \in \widehat{sig}(P_i)(q_i)$. We note $\varphi_i = created(X_i)(q_i)(a)$.For every $j \in \mathcal{J}$, $a \notin \widehat{sig}(P_i)(q_i)$, then we note $\varphi_j = \emptyset$ and $\eta'_j = \delta_{config(X_j)(q_j)}$ that verifies $\delta_{q_j} \overset{f_j}{\leftrightarrow} \eta'_j$ with $f_j = config(X_j)$.

We note $\varphi = created(X)(q)(a)$. By pca-composition definition, $\varphi = \bigcup_{k\in[1:n]} \varphi_k$. For every $k \in [1:n]$, we note $C_k = config(X_k)(q_k)$ and for every $i \in \mathcal{I}$, $\eta'_i \in Disc(Q_{conf})$ s. t. $C_i \overset{a}{\Longrightarrow}_{\varphi_i} \eta'_i$. We note $\tilde{\eta}' = (\eta'_1, ..., \eta'_n)$

By constraint 3 (bottom/up transition preservation), for every $i \in \mathcal{I}$, it exists $(q_i, a, \eta_{X_i,q_i,a}) \in dtrans(X_i)$ s. t. $\eta_{X_i,q_i,a} \overset{f_i}{\leftrightarrow} \eta'_i$ with $f_i = config(X_i)$. by lemma 5.27, $\eta_{X,q,a} = \eta_{X_1,q_1,a} \otimes ... \otimes \eta_{X_n,q_n,a} \overset{f}{\leftrightarrow} \eta'_1 \otimes ... \otimes \eta'_n = join(\tilde{\eta}')$ with the convention $\eta_{X_j,q_j,a} = \delta_{q_j}$ for $j \in \mathcal{J}$ and $f : q = (q_1, ..., q_n) \in states(X) \mapsto (f_1(q_1), ..., f_n(q_n))$.

By partially-compatiblity, for every $C^f \in supp(merge(\tilde{\eta}'))$, C^f is compatible. Hence we can apply lemma 5.26, item 5, which gives $merge(\tilde{\eta}') \overset{g}{\leftrightarrow} join(\tilde{\eta}')$ with $g : C \in supp(merge(\tilde{\eta}')) \mapsto C.split(\tilde{\eta})$.

Hence $\eta_{X,q,a} \overset{h}{\leftrightarrow} merge(\tilde{\eta}')$ with $h = g^{-1} \circ f$, that is $\eta_{X,q,a} \overset{h'}{\leftrightarrow} \eta'$ with $h' = config(X)$ and the restriction of h' on $supp(\eta_{X,q,a})$ is h.

- (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 $X_1, ..., X_n$ 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, $\forall i, j \in [1,n], i \neq j, out(config(X_i)(q_{X_i})) \cap out(config(X_j)(q_{X_j})) = int(config(X_i)(q_{X_i})) \cap \widehat{sig}(config(X_j)(q_{X_j})) = \emptyset$, which finally gives $\forall i, j \in [1,n], i \neq j, \widehat{sig}(config(X_i)(q_{X_i})) \cap h_{X_j} = \emptyset$.

  Hence, we can apply lemma 4.25 of commutativity between hiding and composition to obtain $hide(sig(config(X_1)(q_{X_1})) \times ....\times sig(config(X_n)(q_{X_n})), h_{X_1} \cup ... \cup h_{X_n}) = hide(sig(config(X_1)(q_{X_1})), h_{X_1}) \times ... \times hide(sig(config(X_n)(q_{X_n})), h_{X_n})$ where $\times$ has to be understood in the sense of definition 4.5 of signature composition.

  That is $sig(psioa(X))(q_X) = sig(psioa(X_1))(q_{X_1})) \times .... \times sig(psioa(X_n))(q_{X_n}))$, as per definition 4.5, with $sig(psioa(X))(q_X) = hide(sig(config(X)(x)), h)$. Furthermore $h \subset out(config(X)(q_X))$, since $\forall i \in [1,n]$, $h_{X_i} \subset out(config(X_i)(q_{X_i}))$. This terminates the proof.

□

6 INTRODUCTION ON PCA CORRESPONDING W.R.T. PSIOA $\mathcal{A}$, $\mathcal{B}$ TO INTRODUCE MONOTONICITY

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 $X_{\mathcal{B}}$. This definition is a key step to formally define monotonicity of a property.

If a property P is a binary relation on automata, a brave property would verify monotonicity, i. e. if 1) $(\mathcal{A}, \mathcal{B}) \in P$, and 2) $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ are PCA that differ only on the fact that $\mathcal{B}$ supplants $\mathcal{A}$ in $X_{\mathcal{B}}$, then 3) $(X_{\mathcal{A}}, X_{\mathcal{B}}) \in P$. Monotonicity of implementation w.r.t. PSIOA creation is the main contribution of the paper.

6.1 Naive correspondence between two PCA

We formalize the idea that two configurations are identical except that the automaton $\mathcal{B}$ supplants $\mathcal{A}$ but with the same external signature. The following definition comes from [1].

Definition 6.1 ($\lhd_{\mathcal{AB}}$-corresponding configurations). (see figure 30) 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 $\mathcal{A} \notin \Psi$.

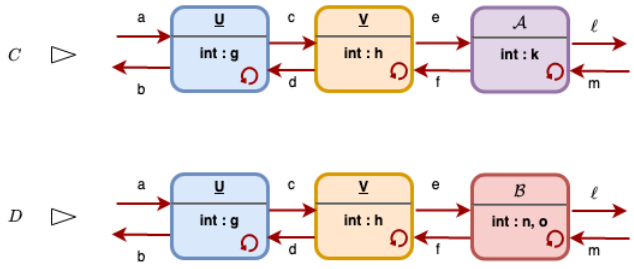


Fig. 17. $\lhd_{\mathcal{AB}}$ corresponding-configuration

Remark 2. *It is possible to have two 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 $int(\mathcal{A})(s) \neq int(\mathcal{A})(t)$.*

Now, we formalise the fact that two PCA create some PSIOA in the same manner, excepting for $\mathcal{B}$ that supplants $\mathcal{A}$. Here again, this definition comes from [1].

Definition 6.2 (Creation corresponding configuration automata). Let X, Y be configuration automata and $\mathcal{A}, \mathcal{B}$ be PSIOA. 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 $\forall a \in \widehat{sig}(X)(x) \cap \widehat{sig}(Y)(y) : created(Y)(y)(a) = created(X)(x)(a)[\mathcal{B}/\mathcal{A}]$.

In the same way than in definition 6.2, we formalise the fact that two PCA hide some output actions in the same manner. Here again, this definition is inspired by [1].

Definition 6.3 (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 $hidden\text{-}actions(Y)(y) = hidden\text{-}actions(X)(x)$.

Now we define the notion of $\mathcal{A}$-exclusive action which corresponds to an action which is is the signature of $\mathcal{A}$ only. This definition is motivated by the fact that monotonicity induces that $\mathcal{A}$-exclusive (resp. $\mathcal{B}$-exclusive) actions do not create automata. Indeed, otherwise two internal action a and a' of $\mathcal{A}$ and $\mathcal{B}$ respectively could create different automata $\mathcal{C}$ and $\mathcal{D}$ and break the correspondence.

Definition 6.4 ($\mathcal{A}$-exclusive action). Let $\mathcal{A} \in Autids$, X be a PCA. Let $q \in states(X)$, $(\mathbf{A}, \mathbf{S}) = config(X)(q)$, $act \in \widehat{sig}(X)(q)$. We say that act is $\mathcal{A}$-exclusive if for every $\mathcal{A}' \in \mathbf{A} \setminus \{\mathcal{A}\}$, $act \notin \widehat{sig}(\mathcal{A}')(\mathbf{S}(\mathcal{A}'))$ (and so $act \in \widehat{sig}(\mathcal{A})(\mathbf{S}(\mathcal{A}))$ only).

The previous definitions 6.1, 6.2, 6.3 and 6.4 allow us to define a first (naive) definition of PCA corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$.

Definition 6.5 (naively 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 naively 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. $\mathcal{A}, \mathcal{B}$
- (No creation from $\mathcal{A}$ and $\mathcal{B}$)
  - $\forall q_{X_{\mathcal{A}}} \in states(X_{\mathcal{A}})$, for every action act $\mathcal{A}$-exclusive, $created(X_{\mathcal{A}})(q_{X_{\mathcal{A}}})(act) = \emptyset$ and similarly
  - $\forall q_{X_{\mathcal{B}}} \in states(X_{\mathcal{B}})$, for every action act' $\mathcal{B}$-exclusive, $created(X_{\mathcal{B}})(q_{X_{\mathcal{B}}})(act') = \emptyset$

The last definition 6.5 of (naive) correspondence w. r. t. $\mathcal{A}$, $\mathcal{B}$ allows us to define a first (naive) definition 6.6 of monotonic relation.

Definition 6.6 (Naively monotonic relationship). Let R be a binary relation on PSIOA. We say that R is naively monotonic if for every pair of PSIOA $(\mathcal{A}, \mathcal{B}) \in R$, for every pair of PCA $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ that are naively corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$, $(psioa(X_{\mathcal{A}}), psioa(X_{\mathcal{B}})) \in R$

.

However, the relations of $print$-implementation introduced in subsection 4.6 is not monotonic without some additional technical assumptions presented in next subsection 6.2. Roughly speaking, it allows to 1) define a PCA $Y = X \setminus \{\mathcal{A}\}$ that corresponds to X "deprived" from $\mathcal{A}$ and 2) define the composition between Y and $\mathcal{A}$, 3) avoiding some ambiguities during the construction. In the first instance, the reader should skip the next subsection 6.2 on conservatism and keep in mind the intuition only. This sub-section 6.2 can be used to know the assumptions of the theorems of monotonicity and use them as black-boxes. The assumptions will be re-called during the proof.

6.2 Conservatism: the additional assumption for relevant definition of correspondence w. r. t. $\mathcal{A}, \mathcal{B}$

This subsection aims to define the notion of $\mathcal{A}$-conservative PCA.

Some definitions relative to configurations. In the remaining, it will often be useful to reason on the configurations. This is why we introduce some definitions that will be used again and again in the demonstrations.

The next definition captures the idea that two states of a certain layer represents the same situation for the bottom layer.

Definition 6.7 (configuration-equivalence between two states). Let K, K' be PCA and $(q, q') \in states(K) \times states(K')$. We say that q and q' are *config-equivalent*, noted $qR_{conf}q'$, if $config(K)(q) = config(K')(q')$. Furthermore, if

- $config(K)(q) = config(K')(q')$,
- $hidden\text{-}actions(K)(q) = hidden\text{-}actions(K')(q')$ and
- $\forall a \in \widehat{sig}(K)(q) = \widehat{sig}(K')(q')$, $created(K)(q)(a) = created(K')(q')(a)$,

we say that q and q' are *strictly-equivalent*, noted $qR_{strict}q'$.

Now, we define a special subset of PCA that do not tolerate different configuration-equivalent states.

Definition 6.8 (Configuration-conflict-free PCA). Let K be a PCA. We say K is configuration-conflict-free, if for every $q, q' \in states(K)$ s. t. $qR_{conf}q'$, then $q = q'$. The current state of a configuration-conflict-free PCA can be defined by its current attached configuration.

For some elaborate definitions, we found useful to introduce the set of potential output actions of $\mathcal{A}$ in a configuration $config(X)(q)$ coming from a state q of a PCA X:

Definition 6.9 (potential ouput). Let $\mathcal{A} \in autids$. Let X be a PCA. Let $q \in states(X)$. We note $pot\text{-}out(X)(q)(\mathcal{A})$ the set of potential output actions of $\mathcal{A}$ in $config(X)(q)$ that is

- $pot\text{-}out(X)(q)(\mathcal{A}) = \emptyset$ if $\mathcal{A} \notin auts(config(X)(q))$
- $pot\text{-}out(X)(q)(\mathcal{A}) = out(\mathcal{A})(map(config(X)(q))(\mathcal{A}))$ if $\mathcal{A} \in auts(config(X)(q))$

Here, we define a configuration C deprived from an automaton $\mathcal{A}$ in the most natural way.

Definition 6.10 ($C \setminus \{\mathcal{A}\}$ Configuration deprived from an automaton). $C = (\mathbf{A}, \mathbf{S})$. $C \setminus \{\mathcal{A}\} = (\mathbf{A}', \mathbf{S}')$ with $\mathbf{A}' = \mathbf{A} \setminus \{\mathcal{A}\}$ and $\mathbf{S}'$ the restriction of $\mathbf{S}$ on $\mathbf{A}'$

The two last definitions 6.9 and 6.10 allows us to define in compact way a new relation between states that captures the idea that two states $q \in states(X)$ and $q' \in states(Y)$ are equivalent modulo a difference uniquely due to the presence of automaton $\mathcal{A}$ in $config(X)(q)$ and $config(Y)(q')$.

Definition 6.11 ($R^{\setminus\{\mathcal{A}\}}$ relationship (equivalent if we forget $\mathcal{A}$)). Let $\mathcal{A} \in Autids$. Let $S = \{states(X) | X \text{ is a PCA }\}$ the set of states of any PCA. We defined the equivalence relation $R_{conf}^{\setminus\{\mathcal{A}\}}$ and $R_{conf}^{\setminus\{\mathcal{A}\}}$ on S defined by $\forall X, Y$ PCA, $\forall (q_X, q_Y) \in states(X) \times states(Y)$:

- $q_X R_{conf}^{\setminus\{\mathcal{A}\}} q_Y \iff config(X)(q_X) \setminus \{A\} = config(Y)(q_Y) \setminus \{A\}$
- $q_X R_{strict}^{\setminus\{\mathcal{A}\}} q_Y \iff$ the conjonction of the 3 following properties:
  - $q_X R_{conf}^{\setminus\{\mathcal{A}\}} q_Y$
  - $\forall a \in \widehat{sig}(X)(q_X) \cap \widehat{sig}(Y)(q_Y), created(Y)(q_Y)(a) \setminus \{\mathcal{A}\} = created(X)(q_X)(a) \setminus \{\mathcal{A}\}$
  - $hidden\text{-}actions(X)(q_X) \setminus pot\text{-}out(X)(q_X)(\mathcal{A}) = hidden\text{-}actions(Y)(q_Y) \setminus pot\text{-}out(Y)(q_Y)(\mathcal{A})$

Here, we recall the definition 6.4 of $\mathcal{A}$-exclusive action:

Definition 6.12 ($\mathcal{A}$-exclusive action in a PCA state (recall)). Let $\mathcal{A} \in Autids$. Let X be a PCA. Let $q_X \in states(X)$, $(\mathbf{A}, \mathbf{S}) = config(X)(q_X)$, $a \in \widehat{sig}(X)(q_X)$.

We say that a is $\mathcal{A}$-exclusive in q_X if $\forall \mathcal{B} \in \mathbf{A} \setminus \{\mathcal{A}\}$, $a \notin \widehat{sig}(\mathcal{B})(\mathbf{S}(\mathcal{B}))$ (and so $a \in \widehat{sig}(\mathcal{A})(\mathbf{S}(\mathcal{A}))$ uniquely necessarily).

$\mathcal{A}$-fair and $\mathcal{A}$-conservative: necessary assumptions to authorize the construction used in the proof. Now, we are ready to define $\mathcal{A}$-fairness and then $\mathcal{A}$-conservatism.

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 assumption will allow us to define $Y = X \setminus \{\mathcal{A}\}$ in the proof of monotonicity.

Definition 6.13 ($\mathcal{A}$-fair PCA). Let $\mathcal{A} \in Autids$. Let X be a PCA. We say that X is $\mathcal{A}$-fair if

- (configuration-conflict-free) X is configuration-conflict-free.
- (no conflict for projection) $\forall q_X, q'_X \in states(X)$, s. t. $q_X R_{conf}^{\setminus\{\mathcal{A}\}} q'_X$ then $q_X R_{strict}^{\setminus\{\mathcal{A}\}} q'_X$.
- (no exclusive creation by $\mathcal{A}$) $\forall q_X \in states(X)$, $\forall a \in \widehat{sig}(X)(q_X)$ $\mathcal{A}$-exclusive in q_X, $created(X)(q_X)(a) = \emptyset$

This definition 6.13 allows the next definition 6.14 to be well-defined. A $\mathcal{A}$-conservative PCA is a $\mathcal{A}$-fair PCA that does not hide any output action that could be an external action of $\mathcal{A}$. This assumption will allow us to define the composition between $\mathcal{A}$ and $Y = X \setminus \{\mathcal{A}\}$ in the proof of monotonicity.

Definition 6.14 ($\mathcal{A}$-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 $map(C_X)(\mathcal{A}) \triangleq q_{\mathcal{A}}$, $hidden\text{-}actions(X)(q_X) \cap \widehat{ext}(\mathcal{A})(q_{\mathcal{A}}) = \emptyset$.

6.3 Corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$

We are closed to state all the technical assumptions to achieve monotonicity of *print*-implementation w.r.t. PSIOA creation. We introduce one last assumption so-called *creation-explicitness*, used in section 11 to reduce implementation of $X_{\mathcal{B}}$ by $X_{\mathcal{A}}$ to implementation of $\mathcal{B}$ by $\mathcal{A}$.

Intuitively, a PCA is *$\mathcal{A}$-creation-explicit* if the creation of a sub-automaton $\mathcal{A}$ is equivalent to the triggering of an action in a dedicated set. This property will allow to obtain the reduction of lemma 11.23.

Definition 6.15 (creation-explicit PCA). Let $\mathcal{A}$ be a PSIOA and X be a PCA. We say that X is *$\mathcal{A}$-creation-explicit* iff: it exists a set of actions, noted $creation\text{-}actions(X)(\mathcal{A})$, s. t. $\forall q_X \in states(X)$, $\forall a \in \widehat{sig}(X)(q_X)$, if we note $\mathbf{A}_X = auts(config(X)(q_X))$ and $\varphi_X = created(X)(q_X)(a)$, then $\mathcal{A} \notin \mathbf{A}_X \wedge \mathcal{A} \in \varphi_X \iff a \in creation\text{-}actions(X)(\mathcal{A})$.

Now we can define new (non naively) correspondence w. r. t. PSIOA $\mathcal{A}$, $\mathcal{B}$ to define (non naively) monotonic relationship.

Definition 6.16 (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 1) they are naively corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$, 2) they are $\mathcal{A}$-conservative and $\mathcal{B}$-conservative respectively and 3) they are $\mathcal{A}$-creation explicit and $\mathcal{B}$-creation explicit with $creation\text{-}actions(X_{\mathcal{A}})(\mathcal{A}) = creation\text{-}actions(X_{\mathcal{B}})(\mathcal{B})$ respectively i. e. they verify:

- $X_{\mathcal{A}}$ is $\mathcal{A}$-conservative and $X_{\mathcal{B}}$ is $\mathcal{B}$-conservative
- $X_{\mathcal{A}}$ is $\mathcal{A}$-creation explicit and $X_{\mathcal{B}}$ is $\mathcal{B}$-creation explicit with $creation\text{-}actions(X_{\mathcal{A}})(\mathcal{A}) = creation\text{-}actions(X_{\mathcal{B}})(\mathcal{B})$
- $config(X_{\mathcal{A}})(\bar{q}_{X_{\mathcal{A}}}) \triangleleft_{AB} config(X_{\mathcal{B}})(\bar{q}_{X_{\mathcal{B}}})$.
- $X_{\mathcal{A}}, X_{\mathcal{B}}$ are creation-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$
- $X_{\mathcal{A}}, X_{\mathcal{B}}$ are hiding-corresponding w.r.t. $\mathcal{A}, \mathcal{B}$
- (No creation from $\mathcal{A}$ and $\mathcal{B}$)
  - $\forall q_{X_{\mathcal{A}}} \in states(X_{\mathcal{A}})$, for every action act $\mathcal{A}$-exclusive, $created(X_{\mathcal{A}})(q_{X_{\mathcal{A}}})(act) = \emptyset$ and similarly

– $\forall q_{X_{\mathcal{B}}} \in states(X_{\mathcal{B}})$, for every action act' $\mathcal{B}$-exclusive, $created(X_{\mathcal{B}})(q_{X_{\mathcal{B}}})(act') = \emptyset$

Definition 6.17 (Monotonic relationship). Let R be a binary relation on PSIOA. We say that R is monotonic if for every pair of PSIOA $(\mathcal{A}, \mathcal{B}) \in R$, for every pair of PCA $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ that are corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$, $(psioa(X_{\mathcal{A}}), psioa(X_{\mathcal{B}})) \in R$.

We would like states the monotonicy of *print*-implementation, but it holds only for a certain class of schedulers, so-called *creation-oblivious* introduced in next subsection 6.4

6.4 Creation-oblivious scheduler

Here we present a particular scheduler schema, that do not take into account previous internal actions of a particular sub-automaton to output its probability over transitions to trigger.

We start by defining *strict oblivious-schedulers* that output the same transition with the same probability for pair of execution fragments that differ only by prefixes in the same class of equivalence. This definition is inspired by the one provided in the thesis of Segala, but is more restrictive since we require a strict equality instead of a correlation (section 5.6.2 in [8]).

Definition 6.18 (oblivious scheduler). Let $\tilde{W}$ be a PCA or a PSIOA, let $\tilde{\sigma} \in schedulers(\tilde{W})$ and let $\equiv$ be an equivalence relation on $Frags^*(\tilde{W})$ verifying $\forall \tilde{\alpha_1}, \tilde{\alpha_2} \in Frags^*(\tilde{W})$ s. t. $\tilde{\alpha_1} \equiv \tilde{\alpha_2}$, $lstate(\alpha_1) = lstate(\alpha_2)$. We say that $\tilde{\sigma}$ is $(\equiv)$-strictly oblivious if $\forall \tilde{\alpha_1}, \tilde{\alpha_2}, \tilde{\alpha_3} \in Frags^*(\tilde{W})$ s. t. 1) $\alpha_1 \equiv \alpha_2$ and 2) $fstate(\tilde{\alpha_3}) = lstate(\tilde{\alpha_2}) = lstate(\tilde{\alpha_1})$, then $\tilde{\sigma}(\tilde{\alpha_1}^\frown \tilde{\alpha_3}) = \tilde{\sigma}(\tilde{\alpha_2}^\frown \tilde{\alpha_3})$.

Now we define the relation of equivalence that defines our subset of creation-oblivious schedulers. Intuitively, two executions fragments ending on $\mathcal{A}$ creation are in the same equivalence class if they differ only in terms of internal actions of $\mathcal{A}$.

Definition 6.19 ($\tilde{\alpha} \equiv^{cr}_{\mathcal{A}} \tilde{\alpha}'$). Let $\mathcal{A}$ be a PSIOA, and $\tilde{W}$ be a PCA. For every $\tilde{\alpha}, \tilde{\alpha}' \in Frags^*(\tilde{W})$, we say $\tilde{\alpha} \equiv^{cr}_{\mathcal{A}} \tilde{\alpha}'$ iff:

(1) $\tilde{\alpha}, \tilde{\alpha}'$ both ends on $\mathcal{A}$-creation.
(2) $\tilde{\alpha}$ and $\tilde{\alpha}'$ differ only in the $\mathcal{A}$-exclusive actions and the states of $\mathcal{A}$, i. e. $\mu(\tilde{\alpha}) = \mu(\tilde{\alpha}')$ where $\mu(\tilde{\alpha} = \tilde{q}^0 a^1 \tilde{q}^1 ... a^n \tilde{q}^n) \in Frags^*(\tilde{W})$ is defined as follows:
   - remove the $\mathcal{A}$-exclusive actions
   - replace each state $\tilde{q}^i$ by its configuration $Config(\tilde{W})(\tilde{q}) = (\mathbf{A}^i, \mathbf{S}^i)$
   - replace each configuration $(\mathbf{A}^i, \mathbf{S}^i)$ by $(\mathbf{A}^i, \mathbf{S}^i) \setminus \{\mathcal{A}\}$
   - replace the (non-alternating) sequences of identical configurations (due to $\mathcal{A}$-exclusiveness of removed actions) by one unique configuration.
(3) $trace(\tilde{\alpha}) = trace(\tilde{\alpha}')$,
(4) $lstate(\tilde{\alpha}) = lstate(\tilde{\alpha}')$

We can remark that the items 4 can be deduced from 1 and 2 if X is configuration-conflict-free.

Definition 6.20 (creation-oblivious scheduler). Let $\tilde{\mathcal{A}}$ be a PSIOA, $\tilde{W}$ be a PCA, $\tilde{\sigma} \in schedulers(\tilde{W})$. We say that $\tilde{\sigma}$ is *$\mathcal{A}$-creation oblivious* if it is $(\equiv^{cr}_{\mathcal{A}})$-strictly oblivious.

We say that $\tilde{\sigma}$ is *creation-oblivious* if it is $\mathcal{A}$-creation oblivious for every sub-automaton $\mathcal{A}$ of $\tilde{W}$ ($\mathcal{A} \in \bigcup_{q \in states(\tilde{W})} auts(config(\tilde{W})(q))$). We note $CrOb$ the function that maps any PCA $\tilde{W}$ to the set of creation-oblivious schedulers of $\tilde{W}$.

We have formally defined our notion of creation-oblivious scheduler. This will be a key property to ensure lemma 11.23 that allows to reduce the measure of a class of comportment as a function of measures of classes of shorter comportment where no creation of $\mathcal{A}$ or $\mathcal{B}$ occurs excepting potentially at very last action. This reduction is more or less necessary to obtain monotonicity of implementation relation:

THEOREM 6.21 ($\leq^{print}_{CrOb}$ IS MONOTONIC). *Let $\mathcal{A}, \mathcal{B} \in Autids$, $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ be PCA corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$. If $\mathcal{A} \leq^{print}_{CrOb} \mathcal{B}$, then $X_{\mathcal{A}} \leq^{print}_{CrOb} X_{\mathcal{B}}$*

The remaining sections are dedicated to the proof of this theorem 6.21. We start by defining in section 7 a morphism between executions of automata, so called *executions-matching*, that preserves structure and measure of probability under *alter ego schedulers*. Next, we define in section 8 the notion of an automaton $X_{\mathcal{A}}$ deprived from a PSIOA $\mathcal{A}$, noted $X_{\mathcal{A}} \setminus \{\mathcal{A}\}$. Furthermore, we show in section 9 that there is an executions-matching from a PCA $X_{\mathcal{A}}$ to $(X_{\mathcal{A}} \setminus \{\mathcal{A}\})||\tilde{\mathcal{A}}^{sw}$ where $\tilde{\mathcal{A}}^{sw}$ is the *simpleton wrapper* of $\mathcal{A}$, i. e. a PCA that only handle $\mathcal{A}$. The section 11 uses the morphism of section 9 to reduce the implementation of $X_{\mathcal{B}}$ by $X_{\mathcal{A}}$ to the implementation of $\mathcal{B}$ by $\mathcal{A}$ and finally obtain the monotonicity of implementation w.r.t. PSIOA creation. Finally section 12 explains why the task-scheduler introduced in [4] is not creation-oblivious.

7 EXECUTIONS-MATCHING

In this section, we introduce some tools to formalise the fact that two automata have the same comportment for the same scheduler. This section is composed by two sub-sections on PSIOA executions-matching and PCA executions-matching. Basically, an executions-matching execution from an automaton $\mathcal{A}$ to another automaton $\mathcal{B}$ is a morphism f^{ex} from $Execs(\mathcal{A})$ to $Execs(\mathcal{B})$ that is structure-preserving. In the remaining, we will often use an executions-matching to show that a pair of executions $(\alpha, \pi = f^{ex}(\alpha)) \in Execs(\mathcal{A}) \times Execs(\mathcal{B})$ have the same probability $\epsilon_{\sigma}(\alpha) = \epsilon_{\sigma'}(\pi)$ under a pair of so-called *alter-ego* schedulers $(\sigma, \sigma') \in schedulers(\mathcal{A}) \times schedulers(\mathcal{B})$ that have corresponding comportment after corresponding executions fragment $(\alpha', \pi' = f^{ex}(\alpha')) \in Frags^*(\mathcal{A}) \times Frags^*(\mathcal{B})$.

7.1 PSIOA executions-matching and semantic equivalence

This first subsection is about PSIOA executions-matching.

matching execution. An executions-matching need a states-matching (see definition 7.1) and a transitions-matching (see definition 7.3) to be defined itself.

Definition 7.1 (states-matching). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA with $Q_{\mathcal{A}} = states(\mathcal{A})$ and $Q_{\mathcal{B}} = states(\mathcal{B})$ as respective sets of states, let $Q'_{\mathcal{A}} \subset Q_{\mathcal{A}}$ and let $f : Q'_{\mathcal{A}} \to Q_{\mathcal{B}}$ be a mapping that verifies:

- Starting state preservation: If $\bar{q}_{\mathcal{A}} \in Q'_{\mathcal{A}}$ then $f(\bar{q}_{\mathcal{A}}) = \bar{q}_{\mathcal{B}}$ (with $(\bar{q}_{\mathcal{A}}, \bar{q}_{\mathcal{B}}) = (start(\mathcal{A}), start(\mathcal{B}))$)
- Signature preservation (modulo an hiding operation): $\forall (q, q') \in Q'_{\mathcal{A}} \times Q_{\mathcal{B}}$, s. t. $q' = f(q)$, $sig(\mathcal{A})(q) = hide(sig(\mathcal{B})(q'), h(q'))$ with $h(q') \subseteq out(\mathcal{B})(q')$ (resp. with $h(q') = \emptyset$, that is $sig(\mathcal{A})(q) = sig(\mathcal{B})(q')$).

then we say that f is a *weak* (resp. *strong*) *states-matching* from $\mathcal{A}$ to $\mathcal{B}$. If $Q'_{\mathcal{A}} = Q_{\mathcal{A}}$, then we say that f is a *complete* (weak or strong) states-matching from $\mathcal{A}$ to $\mathcal{B}$.

Before being able to define transitions-matching, some requirements have to be ensured. A set of transition that would ensure these requirements would be called *eligible to transitions-matching*.

Definition 7.2 (transitions set eligible to transitions matching). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA with $Q_{\mathcal{A}} = states(\mathcal{A})$ and $Q_{\mathcal{B}} = states(\mathcal{B})$ as respective sets of states, let $Q'_{\mathcal{A}} \subset Q_{\mathcal{A}}$ and let $f : Q'_{\mathcal{A}} \to Q_{\mathcal{B}}$ be a states-matching from $\mathcal{A}$ to $\mathcal{B}$. Let $D'_{\mathcal{A}} \subseteq D_{\mathcal{A}} = dtrans(\mathcal{A})$ be a subset of transition. If $D'_{\mathcal{A}}$ verifies that for every $(q, a, \eta_{(\mathcal{A},q,a)}) \in D'_{\mathcal{A}}$:

- Matched states preservation: $q \in Q'_{\mathcal{A}}$ and
- Equitable corresponding distribution: $\forall q'' \in supp(\eta_{(\mathcal{A},q,a)}), q'' \in Q'_{\mathcal{A}}$ and $\eta_{(\mathcal{A},q,a)} \xleftrightarrow{f} \eta_{(\mathcal{B},f(q),a)}$ (i. e. the restriction of f, $\tilde{f} : supp(\eta_{(\mathcal{A},q,a)}) \to supp(\eta_{(\mathcal{B},f(q),a)})$ is bijective and $\forall q'' \in supp(\eta_{(\mathcal{A},q,a)}), \eta_{(\mathcal{A},q,a)}(q'') = \eta_{(\mathcal{B},f(q),a)}(f(q''))$).

then we say that $D'_{\mathcal{A}}$ is *eligible to transitions-matching domain from* f. We omit to mention the states-matching f when this is clear in the context.

Now, we are able to define a transitions-matching, which is a property-preserving mapping from a set of transitions $D'_{\mathcal{A}} \subseteq dtrans(\mathcal{A})$ to another set of transitions $D'_{\mathcal{B}} \subseteq dtrans(\mathcal{B})$.

Definition 7.3 (transitions-matching). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA with $Q_{\mathcal{A}} = states(\mathcal{A})$ and $Q_{\mathcal{B}} = states(\mathcal{B})$ as respective sets of states, let $Q'_{\mathcal{A}} \subset Q_{\mathcal{A}}$ and let $f : Q'_{\mathcal{A}} \to Q_{\mathcal{B}}$ be a states-matching from $\mathcal{A}$ to $\mathcal{B}$. Let $D'_{\mathcal{A}} \subseteq D_{\mathcal{A}}$ be a subset of transition eligible to transitions-matching domain from f.

We define the *transitions-matching* (f, f^{tr}) *from* $\mathcal{A}$ *to* $\mathcal{B}$ *induced by the states-matching* f *and the subset of transition* $D'_{\mathcal{A}}$ s. t. $f^{tr} : D'_{\mathcal{A}} \to D_{\mathcal{B}}$ is defined by $f^{tr}((q, a, \eta_{(\mathcal{A},q,a)})) = (f(q), a, \eta_{(\mathcal{B},f(q),a)})$. If f is complete and $D'_{\mathcal{A}} = D_{\mathcal{A}}$, (f, f^{tr}) is said to be a complete transitions-matching. If f is weak (resp. strong) (f, f^{tr}) is said to be a weak (resp. strong) transitions-matching. If f is clear in the context, with a slight abuse of notation, we say that f^{tr} is a transitions-matching.

The function f^{tr} need to verify some constraints imposed by f, but if the set $D'_{\mathcal{A}}$ of concerned transitions is correctly-chosen to ensure the 2 properties of definition 7.2, then such a transitions-matching is unique.

Now, we can easily define an executions-matching with a transitions-matching, which is a property-preserving mapping from a set of execution fragments $F'_{\mathcal{A}} \subseteq Frags(\mathcal{A})$ to another set of execution fragments $F'_{\mathcal{B}} \subseteq Frags(\mathcal{B})$.

Definition 7.4 (executions-matching). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA. Let (f, f^{tr}) be a transitions-matching from $\mathcal{A}$ to $\mathcal{B}$. Let $F'_{\mathcal{A}} = \{\alpha \triangleq q^0 a^1 q^1 ... a^n q^n ... \in Frags(\mathcal{A}) | \forall i \in [0 : |\alpha| - 1], (q^i, a^{i+1}, \eta_{(\mathcal{A},q^i,a^{i+1})}) \in dom(f^{tr})\}$. Let $f^{ex} : F'_{\mathcal{A}} \to Frags(\mathcal{B})$, built from (f, f^{tr}) s. t. $\forall \alpha = q^0_{\mathcal{A}} a^1 q^1_{\mathcal{A}} ... a^n q^n_{\mathcal{A}} ... \in F'_{\mathcal{A}}$, $f^{ex}(\alpha) = f(q^0_{\mathcal{A}}) a^1 f(q^1_{\mathcal{A}}) ... a^n f(q^n_{\mathcal{A}}) ...$

We say that (f, f^{tr}, f^{ex}) is an *executions-matching* from $\mathcal{A}$ to $\mathcal{B}$. Furthermore, if (f, f^{tr}) is complete and $F'_{\mathcal{A}} = Frags(\mathcal{A})$, (f, f^{tr}, f^{ex}) is said to be a complete executions-matching. If (f, f^{tr}) is weak (resp. strong) (f, f^{tr}, f^{ex}) is said to be a weak (resp. strong) executions-matching. When (f, f^{tr}) is clear in the context, with a slight abuse of notation, we say that f^{ex} is an executions-matching.

The function f^{ex} is completely defined by (f, f^{tr}), hence we call (f, f^{tr}, f^{ex}) the executions-matching induced by the transition matching (f, f^{tr}) or the executions-matching induced by the states-matching f and the subset of transitions $dom(f^{tr})$.

The construction of f^{ex} allows us to see two executions mapped by an executions-mapping as a sequence of pairs of transitions mapped by the attached transitions-matching. This result is formalised in next lemma 7.5.

Lemma 7.5 (executions-matching seen as a sequence of transitions-matchings). *Let* $\mathcal{A}$ *and* $\mathcal{B}$ *be two PSIOA. Let* (f, f^{tr}, f^{ex}) *be an executions-matching from* $\mathcal{A}$ *to* $\mathcal{B}$. *Let* $\alpha = q^0_{\mathcal{A}} a^1 q^1_{\mathcal{A}} ... a^n q^n_{\mathcal{A}} ... \in dom(f^{ex})$ *and* $\pi = f^{ex}(\alpha) = q^0_{\mathcal{B}} a^1 q^1_{\mathcal{B}} ... a^n q^n_{\mathcal{B}} ... = f(q^0_{\mathcal{A}}) a^1 f(q^1_{\mathcal{A}}) ... a^n f(q^n_{\mathcal{A}}) ....$ *Then for every* $i \in [0 : |\alpha| - 1]$, $(q^i_{\mathcal{B}}, a^{i+1}, \eta_{(\mathcal{B},q^i_{\mathcal{B}},a^{i+1})}) = f^{tr}((q^i_{\mathcal{A}}, a^{i+1}, \eta_{(\mathcal{A},q^i_{\mathcal{A}},a^i)}))$

Proof. First, matched states preservation and action preservation are ensured by construction. By definition, for every $i \in [0 : |\alpha|-1]$, $(q^i_{\mathcal{A}}, a^{i+1}, \eta_{(\mathcal{A},q^i_{\mathcal{A}},a^{i+1})}) \in dom(f^{tr})$. We note $tr^i_{\mathcal{B}} \triangleq f^{tr}((q^i_{\mathcal{A}}, a^{i+1}, \eta_{(\mathcal{A},q^i_{\mathcal{A}},a^{i+1})}))$. By definition, $tr^i_{\mathcal{B}}$ is of the form $(f(q^i_{\mathcal{A}}), a^{i+1}, \eta)$. But a transition of this form is unique, which means $tr^i_{\mathcal{B}} = (f(q^i_{\mathcal{A}}), a^{i+1}, \eta_{(\mathcal{B},f(q^i_{\mathcal{A}}),a^{i+1})})$ which ends the proof. □

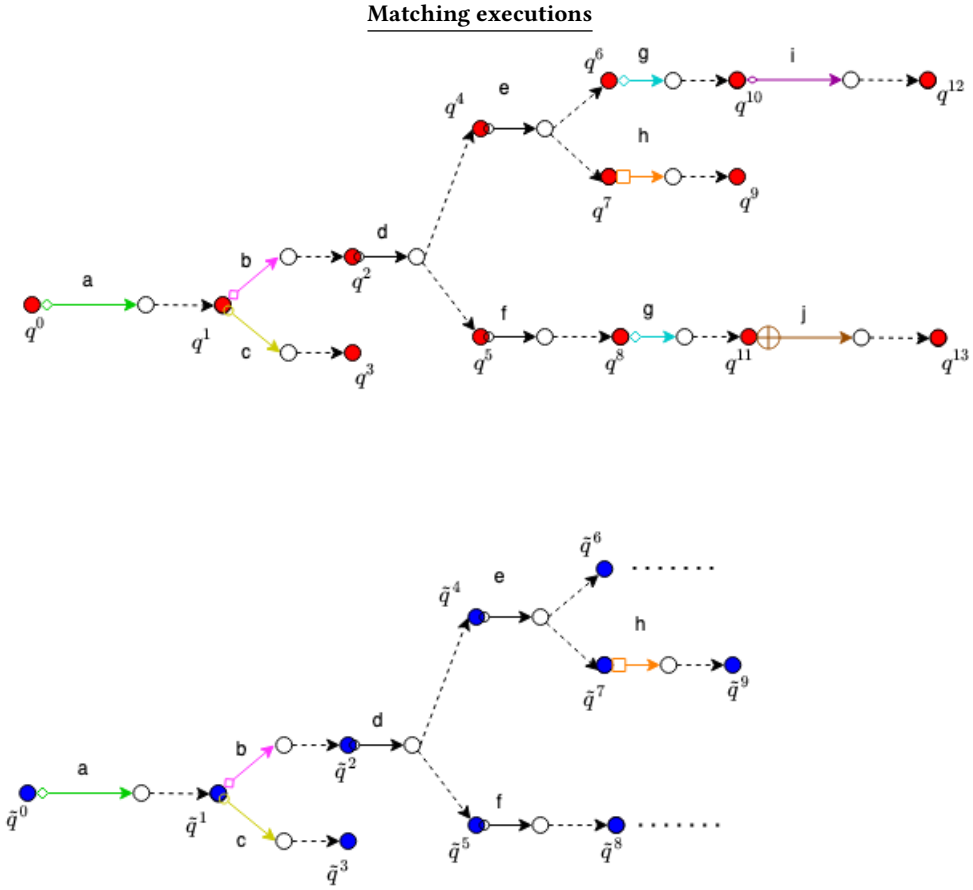


Fig. 18. Here we have $Q'_{\mathcal{A}} = \{q^0, q^1, ..., q^9\} \subsetneq Q_{\mathcal{A}}$, we define the state-matching $f : Q'_{\mathcal{A}} \to Q_{\mathcal{B}}$ s. t. $\forall k \in [1, 9], f(q^k) = \tilde{q}^k$, and $D'_{\mathcal{A}} = \{(q^0, a, \eta_{(\mathcal{A},q^0,a)}), (q^1, b, \eta_{(\mathcal{A},q^1,b)}), (q^1, c, \eta_{(\mathcal{A},q^1,c)}), (q^2, d, \eta_{(\mathcal{A},q^2,d)}), (q^4, e, \eta_{(\mathcal{A},q^4,e)}), (q^5, f, \eta_{(\mathcal{A},q^5,f)}), (q^7, h, \eta_{(\mathcal{A},q^7,h)})\}$. We can define the execution matching (f, f^{tr}, f^{ex}) induced by f and $D'_{\mathcal{A}}$.

Now we overload the definition of executions-matching to be able to state the main result of this paragraph i. e. theorem 7.9

Definition 7.6 (executions-matching overload: pre-execution-distribution). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA. Let (f, f^{tr}, f^{ex}) be an executions-matching from $\mathcal{A}$ to $\mathcal{B}$.

Let $(\mu, \mu') \in Disc(Frags(\mathcal{A})) \times Disc(Frags(\mathcal{B}))$ s. t. $\mu \overset{f^{ex}}{\leftrightarrow} \mu'$ (i.e. th restriction of f^{ex} on $supp(\tilde{\mu})$ is a bijection from $supp(\tilde{\mu})$ to $supp(\mu)$ and for every $\alpha \in supp(\mu)$, $\mu'(f^{ex}(\alpha)) = \mu(\alpha)$). Then we say that (f, f^{tr}, f^{ex}) is an *executions-matching* from $(\mathcal{A}, \mu)$ to $(\mathcal{B}, \mu')$.

In practice, we will often use executions-matching from $(\mathcal{A}, \delta_{\tilde{q}_{\mathcal{A}}})$ to $(\mathcal{B}, \delta_{\tilde{q}_{\mathcal{B}}})$.

Continued executions-matching. Motivated by PSIOA creation that would break the states-matching from a PCA $X_{\mathcal{A}}$ to the PCA $Z_{\mathcal{A}} \triangleq (X \setminus \{\mathcal{A}\})||\tilde{\mathcal{A}}^{sw}$ defined in section 9, we introduce the notion of continuation of executions-matching.

Definition 7.7 (Continued executions-matching). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA with $Q_{\mathcal{A}} = states(\mathcal{A})$ and $Q_{\mathcal{B}} = states(\mathcal{B})$ as respective sets of states and $D_{\mathcal{A}} = dtrans(\mathcal{A})$ and $D_{\mathcal{B}} = dtrans(\mathcal{B})$ as respective sets of discrete transitions. Let (f, f^{tr}, f^{ex}) be an executions-matching from $\mathcal{A}$ to $\mathcal{B}$ with $dom(f) \triangleq Q'_{\mathcal{A}} \subset Q_{\mathcal{A}}$ and $dom(f^{tr}) \triangleq D'_{\mathcal{A}} \subset D_{\mathcal{A}}$.

Let $f^{+} : Q''_{\mathcal{A}} \to Q_{\mathcal{B}}$ with $Q''_{\mathcal{A}} \subset Q_{\mathcal{A}}$. Let $D''_{\mathcal{A}} \subset dtrans(\mathcal{A})$ be a subset of transitions verifying for every $(q, a, \eta_{(\mathcal{A},q,a)}) \in D''_{\mathcal{A}} \setminus D'_{\mathcal{A}}$:

- Matched states preservation: $q \in Q'_{\mathcal{A}}$
- Extension of equitable corresponding distribution: $\forall q'' \in supp(\eta_{(\mathcal{A},q,a)}), q'' \in Q''_{\mathcal{A}}$ and $\eta_{(\mathcal{A},q,a)} \overset{f^{+}}{\longleftrightarrow} \eta_{(\mathcal{B},f(q),a)}$, i. e. the restriction of f^{+} on $supp(\eta_{(\mathcal{A},q,a)})$, $\tilde{f}^{+} : supp(\eta_{(\mathcal{A},q,a)}) \to supp(\eta_{(\mathcal{B},f(q),a)})$ is bijective and $\forall q'' \in supp(\eta_{(\mathcal{A},q,a)})$, $\eta_{(\mathcal{A},q,a)}(q'') = \eta_{(\mathcal{B},f(q),a)}(f^{+}(q''))$

We define the $(f^{+}, D''_{\mathcal{A}})$-*continuation of* f^{tr} as the function $f^{tr,+} : D'_{\mathcal{A}} \cup D''_{\mathcal{A}} \to D_{\mathcal{B}}$ s. t. $\forall (q, a, \eta_{(\mathcal{A},q,a)}) \in D'_{\mathcal{A}} \cup D''_{\mathcal{A}}, f^{tr,+}((q, a, \eta_{(\mathcal{A},q,a)})) = (f(q), a, \eta_{(\mathcal{B},f(q),a)})$.

Let $F''_{\mathcal{A}} = dom(f^{ex}) \cup \{\alpha^\frown qaq' \in Execs^{*}(\mathcal{A}) | \alpha \in dom(f^{ex}) \wedge (q, a, \eta_{(\mathcal{A},q,a))} \in D''_{\mathcal{A}}\}$. We define the $(f^{tr,+})$-*continuation of* f^{ex} as the function $f^{ex,+} : F''_{\mathcal{A}} \to Frags(\mathcal{B})$ s. t. $\forall \alpha \in dom(f^{ex}), f^{ex,+}(\alpha) = f^{ex}(\alpha)$ and $\forall \alpha' = \alpha^\frown q, a, q' \in F''_{\mathcal{A}} \setminus dom(f^{ex}), f^{ex,+}(\alpha') = f^{ex}(\alpha)^\frown f(q), a, f^{+}(q')$.

Then, we say that $((f, f^{+}), f^{tr,+}, f^{ex,+})$ is the $(f^{+}, D''_{\mathcal{A}})$-*continuation* of (f, f^{tr}, f^{ex}) which is a *continuation* of (f, f^{tr}, f^{ex}) and a *continued executions-matching from* $\mathcal{A}$ *to.* $\mathcal{B}$.

Moreover, if $(\mu, \mu') \in Disc(Frags(\mathcal{A})) \times Disc(Frags(\mathcal{B}))$ s. t. $\mu \overset{f^{ex,+}}{\longleftrightarrow} \mu'$ (i. e. the restriction $\tilde{f}^{ex,+}$ of $f^{ex,+}$ on $supp(\tilde{\mu})$ is a bijection from $supp(\tilde{\mu})$ to $supp(\mu)$ and for every $\alpha \in supp(\mu)$, $\mu'(f^{ex,+}(\alpha)) = \mu(\alpha)$), then we say that $((f, f^{+}), f^{tr,+}, f^{ex,+})$ is a *continued executions-matching* from $(\mathcal{A}, \mu)$ to $(\mathcal{B}, \mu')$.

From executions-matching to probabilistic distribution preservation. We want to states that a (potentially-continued) executions-matching preserves measure of probability of the corresponding executions.

To do so, we define alter egos schedulers to a certain executions-matching. Such pair of schedulers are very similar in the sense that their outputs depends only on the semantic structure of the input, preserved by the executions-matching.

Definition 7.8 ((f, f^{tr}, f^{ex})-alter egos schedulers). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA. Let (f, f^{tr}, f^{ex}) be an executions-matching from $\mathcal{A}$ to $\mathcal{B}$. Let $(\tilde{\sigma}, \sigma) \in schedulers(\mathcal{A}) \times schedulers(\mathcal{B})$. We say that $(\tilde{\sigma}, \sigma)$ are (f, f^{tr}, f^{ex})-*alter egos* (or f^{ex}-*alter egos*) if, and only if, for every $(\tilde{\alpha}, \alpha) \in Frags^{*}(\mathcal{A}) \times Frags^{*}(\mathcal{B})$ s. t. $\alpha = f^{ex}(\tilde{\alpha})$ (which means $\widehat{sig}(\mathcal{A})(\tilde{q}) = \widehat{sig}(\mathcal{B})(q) \triangleq sig$ with $\tilde{q} = lstate(\tilde{\alpha})$ and $q = lstate(\alpha)$ by signature preservation property of the associated states-matching), $\forall a \in sig, \tilde{\sigma}(\tilde{\alpha})((\tilde{q}, a, \eta_{(\mathcal{A},\tilde{q},a)})) = \sigma(\alpha)((q, a, \eta_{(\mathcal{B},q,a)}))$.

Let us remark that the previous definition implies that the probability of halting after corresponding executions fragments $(\tilde{\alpha}, \alpha)$ is also the same.

Now we are ready to states an intuitive result that will be often used in the remaining.

Theorem 7.9 (Executions-matching preserves general probabilistic distribution). *Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA. Let $(\tilde{\mu}, \mu) \in Disc(Frags(\mathcal{A})) \times Disc(Frags(\mathcal{B}))$. Let (f, f^{tr}, f^{ex}) be an executions-matching from $(\mathcal{A}, \tilde{\mu})$ to $(\mathcal{B}, \mu)$. Let $(\tilde{\sigma}, \sigma) \in schedulers(\mathcal{A}) \times schedulers(\mathcal{B})$, s. t. $(\tilde{\sigma}, \sigma)$ are (f, f^{tr}, f^{ex})*-alter egos. *Let $(\tilde{\alpha}, \alpha) \in Frags^{*}(\mathcal{A}) \times Frags^{*}(\mathcal{B})$ s. t. $\alpha = f^{ex}(\tilde{\alpha})$. Then $\epsilon_{\tilde{\sigma},\tilde{\mu}}(C_{\tilde{\alpha}}) = \epsilon_{\sigma,\mu}(C_{\alpha})$ and $\epsilon_{\tilde{\sigma},\tilde{\mu}}(\tilde{\alpha}) = \epsilon_{\sigma,\mu}(\alpha)$.*

Proof. First, by definition 7.6 of executions-matching, f^{ex} is a bijection from $supp(\tilde{\mu})$ to $supp(\mu)$ where $\forall \tilde{\alpha}_o \in supp(\tilde{\mu}), \mu(f^{ex}(\tilde{\alpha}_o)) = \tilde{\mu}(\tilde{\alpha}_o)$ (*). Second, by definition 4.13 of measure generated by a scheduler, $\epsilon_{\sigma,\mu}(C_{\alpha'}) = \Sigma_{\alpha_o \in supp(\mu)} \mu(\alpha_o) \cdot \epsilon_{\sigma,\alpha_o}(C_{\alpha'})$ and $\epsilon_{\tilde{\sigma},\tilde{\mu}}(C_{\tilde{\alpha}'}) = \Sigma_{\tilde{\alpha}_o \in supp(\tilde{\mu})} \tilde{\mu}(\tilde{\alpha}_o) \cdot \epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(C_{\tilde{\alpha}'})$ (**). Hence, by combining (*) and (**), we only need to show that for every $(\tilde{\alpha}_o, \alpha_o) \in supp(\tilde{\mu}) \times supp(\mu)$ with $f^{ex}(\tilde{\alpha}_o) = \alpha_o$, for every $(\tilde{\alpha}', \alpha') \in Frags^*(\mathcal{A}) \times Frags^*(\mathcal{B})$ with $f^{ex}(\tilde{\alpha}') = \alpha'$, we have $\epsilon_{\sigma,\alpha_o}(C_{\alpha'}) = \epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(C_{\tilde{\alpha}'})$ that we show by induction on the size $s = |\tilde{\alpha}| = |\alpha|$. We fix $(\tilde{\alpha}_o, \alpha_o) \in supp(\tilde{\mu}) \times supp(\mu)$ with $f^{ex}(\tilde{\alpha}_o) = \alpha_o$.

Basis: $s = 0$

Let $\tilde{\alpha}' = \tilde{q}' \in Frags^*(\mathcal{A})$, $\alpha' = q' \in Frags^*(\mathcal{B})$ with $\alpha' = f^{ex}(\tilde{\alpha}')$. We have $|\tilde{\alpha}'| = |\alpha'| = 0$. By definition 4.13 of measure generated by a scheduler,

$$\epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(C_{\tilde{\alpha}'}) =: \begin{cases} 0 & \text{if both } \tilde{\alpha}' \not\leq \tilde{\alpha}_o \text{ and } \tilde{\alpha}_o \not\leq \tilde{\alpha}' \\ 1 & \text{if } \tilde{\alpha}' \leq \tilde{\alpha}_o \\ \epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(C_{\tilde{\alpha}}) \cdot \tilde{\sigma}(\tilde{\alpha})(\eta_{(\mathcal{A},\tilde{q},a)}) \cdot \eta_{(\mathcal{A},\tilde{q},a)}(\tilde{q}') & \text{if } \tilde{\alpha}_o \leq \tilde{\alpha} \text{ and } \tilde{\alpha}' = \tilde{\alpha}^\frown \tilde{q} a \tilde{q}' \end{cases} \text{ and}$$

$$\epsilon_{\sigma,\alpha_o}(C_{\alpha'}) =: \begin{cases} 0 & \text{if both } \alpha' \not\leq \alpha_o \text{ and } \alpha_o \not\leq \alpha' \\ 1 & \text{if } \alpha' \leq \alpha_o \\ \epsilon_{\sigma,\alpha_o}(C_{\alpha}) \cdot \sigma(\alpha)(\eta_{(\mathcal{B},q,a)}) \cdot \eta_{(\mathcal{B},q,a)}(q') & \text{if } \alpha_o \leq \alpha \text{ and } \alpha' = \alpha^\frown q a q' \end{cases}$$

Since $|\tilde{\alpha}'| = |\alpha'| = 0$ the third case is never met. The second case can be written: $\tilde{\alpha}' \leq \tilde{\alpha}_o$ (resp. $\alpha' \leq \alpha_o$) iff $fstate(\tilde{\alpha}_o) = \tilde{q}'$ (resp. $fstate(\alpha_o) = q'$). Hence, for every $(\tilde{\alpha}_o, \alpha_o)$ s. t. $f^{ex}(\tilde{\alpha}_o) = \alpha_o$, $\epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(C_{\tilde{\alpha}'}) = \epsilon_{\sigma,\alpha_o}(C'_\alpha)$ which ends the basis.

Induction: We assume the result to be true up to size s and we show it implies the result is true for size $s+1$. Let $(\tilde{\alpha}', \tilde{\alpha}, \alpha', \alpha) \in Frags^*(\mathcal{A})^2 \times Frags^*(\mathcal{B})^2$ with $\tilde{\alpha}' = \tilde{\alpha}^\frown \tilde{q} a \tilde{q}'$ and $\alpha' = \alpha^\frown q a q'$ s. t. $\alpha' = f^{ex}(\tilde{\alpha}')$ with $|\tilde{\alpha}'| = |\alpha'| = s+1$. We want to show that $\epsilon_{\tilde{\sigma},\tilde{\mu}}(C_{\tilde{\alpha}'}) = \epsilon_{\sigma,\mu}(C_{\alpha'})$. By definition 4.13 of measure generated by a scheduler,

$$\epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(C_{\tilde{\alpha}'}) =: \begin{cases} 0 & \text{if both } \tilde{\alpha}' \not\leq \tilde{\alpha}_o \text{ and } \tilde{\alpha}_o \not\leq \tilde{\alpha}' \\ 1 & \text{if } \tilde{\alpha}' \leq \tilde{\alpha}_o \\ \epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(C_{\tilde{\alpha}}) \cdot \tilde{\sigma}(\tilde{\alpha})(\eta_{(\mathcal{A},\tilde{q},a)}) \cdot \eta_{(\mathcal{A},\tilde{q},a)}(\tilde{q}') & \text{if } \tilde{\alpha}_o \leq \tilde{\alpha} \text{ and } \tilde{\alpha}' = \tilde{\alpha}^\frown \tilde{q} a \tilde{q}' \end{cases} \text{ and}$$

$$\epsilon_{\sigma,\alpha_o}(C_{\alpha'}) =: \begin{cases} 0 & \text{if both } \alpha' \not\leq \alpha_o \text{ and } \alpha_o \not\leq \alpha' \\ 1 & \text{if } \alpha' \leq \alpha_o \\ \epsilon_{\sigma,\alpha_o}(C_{\alpha}) \cdot \sigma(\alpha)(\eta_{(\mathcal{B},q,a)}) \cdot \eta_{(\mathcal{B},q,a)}(q') & \text{if } \alpha_o \leq \alpha \text{ and } \alpha' = \alpha^\frown q a q' \end{cases}$$

Again, the executions-matching implies that i) both $\tilde{\alpha}' \not\leq \tilde{\alpha}_o$ and $\tilde{\alpha}_o \not\leq \tilde{\alpha}' \iff$ both $\alpha' \not\leq \alpha_o$ and $\alpha_o \not\leq \alpha'$, ii) $\tilde{\alpha} \leq \tilde{\alpha}_o \iff \alpha \leq \alpha_o$ and iii) $\tilde{\alpha}_o \leq \tilde{\alpha} \iff \alpha_o \leq \alpha$. Moreover, by induction assumption $\epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(C_{\tilde{\alpha}}) = \epsilon_{\sigma,\alpha_o}(C_\alpha)$. Hence we only need to show that $\tilde{\sigma}(\tilde{\alpha})(\eta_{(\mathcal{A},\tilde{q},a)}) \cdot \eta_{(\mathcal{A},\tilde{q},a)}(\tilde{q}') = \sigma(\alpha)(\eta_{(\mathcal{B},q,a)}) \cdot \eta_{(\mathcal{B},q,a)}(q')$ (***). By definition of alter-ego schedulers, $\tilde{\sigma}(\tilde{\alpha})(\eta_{(\mathcal{A},\tilde{q},a)}) = \sigma(\alpha)(\eta_{(\mathcal{B},q,a)})$ (j). By definition of executions-matching, $\eta_{(\mathcal{A},\tilde{q},a)}(\tilde{q}') = \eta_{(\mathcal{B},q,a)}(q')$ (jj). Thus (j) and (jj) implies (***) which allows us to terminate the induction to obtain $\epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(C_{\tilde{\alpha}'}) = \epsilon_{\sigma,\alpha_o}(C_{\alpha'})$.

Finally, let $sig = \widehat{sig}(\mathcal{A})(lstate(\tilde{\alpha}')) = \widehat{sig}(\mathcal{A})(lstate(\alpha'))$, then $\epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(\tilde{\alpha}') = \epsilon_{\tilde{\sigma},\tilde{\alpha}_o}(C_{\tilde{\alpha}'}) \cdot (1 - \Sigma_{a \in sig} \tilde{\sigma}(\tilde{\alpha}')(a)) = \epsilon_{\sigma,\alpha_o}(C_{\alpha'}) \cdot (1 - \Sigma_{a \in sig} \sigma(\alpha')(a)) = \epsilon_{\sigma,\alpha_o}(\alpha')$, which ends the proof.

□

We restate the previous theorem with continued executions-matching.

Theorem 7.10 (Continued executions-matching preserves general probabilistic distribution). *Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA. Let $(\tilde{\mu}, \mu) \in Disc(Frags(\mathcal{A})) \times Disc(Frags(\mathcal{B}))$. Let (f, f^{tr}, f^{ex}) be an executions-matching from $(\mathcal{A}, \tilde{\mu})$ to $(\mathcal{B}, \mu)$. Let $((f, f^+), f^{tr,+}, f^{ex,+})$ be a continuation of (f, f^{tr}, f^{ex}). Let $(\tilde{\sigma}, \sigma) \in schedulers(\mathcal{A}) \times schedulers(\mathcal{B})$, s. t. $(\tilde{\sigma}, \sigma)$ are (f, f^{tr}, f^{ex})*-alter egos. *Let $(\tilde{\alpha}, \alpha) \in Frags^*(\mathcal{A}) \times Frags^*(\mathcal{B})$ s. t. $\alpha = f^{ex,+}(\tilde{\alpha})$. Then $\epsilon_{\tilde{\sigma},\tilde{\mu}}(C_{\tilde{\alpha}}) = \epsilon_{\sigma,\mu}(C_\alpha)$.*

Proof. The proof is exactly the same than the one for theorem 7.9 □

Before dealing with composability of executions-matching, we prove two results about injectivity and surjectivity of executions-matching in next lemma 7.11 and 7.12.

Lemma 7.11 (Injectivity of executions-matching). *Let (f, f^{tr}, f^{ex}) be an executions-matching from $\mathcal{A}$ to $\mathcal{B}$ and $((f, f^{+}), f^{tr,+}, f^{ex,+})$ a continuation of (f, f^{tr}, f^{ex}).*

Let $\tilde{f}^{ex,+} : F''_{\mathcal{A}} \subseteq dom(f^{ex,+}) \rightarrow \tilde{F}_{\mathcal{B}} \subseteq range(f^{ex,+})$. Let $\tilde{f} : Q''_{\mathcal{A}} \subseteq dom(f) \rightarrow states(\mathcal{B})$ be the restriction of f on a set $Q''_{\mathcal{A}} \subseteq dom(f)$.

1. *If i) $\forall \alpha \in F''_{\mathcal{A}}, fstate(\alpha) \in Q''_{\mathcal{A}}$ and ii) $\tilde{f}$ is injective, then $\tilde{f}^{ex,+}$ is injective.*
2. *(Corollary) if $F''_{\mathcal{A}} \subseteq Execs(\mathcal{A})$, $f^{ex,+}$ is injective.*

Proof. (1) By induction on the size k of the prefix: Basis: By i) $fstate(\alpha), fstate(\alpha') \in Q''_{\mathcal{A}}$, by construction of $f^{ex,+}$, $f(fstate(\alpha)) = f(fstate(\alpha')) = fstate(\pi)$ and by ii) $fstate(\alpha) = fstate(\alpha')$ Induction. We assume the injectivity of $\tilde{f}^{ex,+}$ to be true for execution on size k and we show this is also true for size $k+1$. Let $\pi = s^0 b^1 s^1 ... s^k b^{k+1} s^{k+1} \in F''_{\mathcal{B}}$ Let $\alpha = q^0 a^1 q^1 ... q^k a^{k+1} q^{k+1}, \alpha' = q'^0 a'^1 q'^1 ... q'^k a'^{k+1} q'^{k+1} \in F''_{\mathcal{A}}$ s. t. $f(\alpha) = f(\alpha') = \pi$. By construction of $f^{ex,+}$, $\forall i \in [1, n]$, $b^i = a^i = a'^i$. By construction of $f^{ex,+}$, $f^{ex,+}(q'^0 a'^1 q'^1 ... q'^k) = f^{ex,+}(q^0 a^1 q^1 ... q^k) = s^0 a^1 s^1 ... s^k$. By induction assumption $q'^0 a'^1 q'^1 ... q'^k) = q^0 a^1 q^1 ... q^k$. By definition of execution, $s^{k+1} \in supp(\eta_{(\mathcal{B}, s^k, a^{k+1})})$. By equitable corresponding distribution, If $\eta_{(\mathcal{A}, q^k, a^{k+1})} \in dom(f^{tr})$, the restriction of f, $\tilde{f} : supp(\eta_{(\mathcal{A}, q^k, a^{k+1})}) \rightarrow supp(\eta_{(\mathcal{B}, s^k, a^{k+1})})$ is bijective and $\eta_{(\mathcal{A}, q^k, a^{k+1})} \in dom(f^{tr,+}) \setminus dom(f^{tr})$, the restriction of f^+, $\tilde{f^+} : supp(\eta_{(\mathcal{A}, q^k, a^{k+1})}) \rightarrow supp(\eta_{(\mathcal{B}, s^k, a^{k+1})})$ is bijective so $q^{k+1} = q'^{k+1}$ which ends the proof.

(2) We have $|start(\mathcal{A})| = 1$. Hence the restriction of f on $start(\mathcal{A})$ is necessarily injective (ii). Let $\alpha \in Execs(\mathcal{A})$. By definition of execution, $fstate(\alpha) \in start(\mathcal{A})$ (i). All the requirements of lemma 7.11, first item are met, which ends the proof.

□

Lemma 7.12 (Surjectivity property preserved by continuation). *Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA. Let (f, f^{tr}, f^{ex}) be an executions-matching from $\mathcal{A}$ to $\mathcal{B}$. Let $((f, f^{+}), f^{tr,+}, f^{ex,+})$ be the $(f^{+}, D''_{\mathcal{A}})$-continuation of (f, f^{tr}, f^{ex}) (where by definition $D''_{\mathcal{A}} \setminus dom(f^{tr})$ respect the properties of matched states preservation and extension of equitable corresponding distribution from definition 7.7). If the restriction $\tilde{f}^{ex} : E'_{\mathcal{A}} \subseteq Execs(\mathcal{A}) \rightarrow \tilde{E}_{\mathcal{B}} \subseteq Execs(\mathcal{B})$ is surjective, then $\tilde{f}^{ex,+} : E'^{+}_{\mathcal{A}} = \{\alpha' = \alpha^\frown q_{\mathcal{A}}, a, q'_{\mathcal{A}} \in Execs(\mathcal{A}) | \alpha \in E_{\mathcal{A}}, (q_{\mathcal{A}}, a, \eta_{\mathcal{A}, q_{\mathcal{A}}, a}) \in dom(f^{tr,+})\} \rightarrow \tilde{E}^{+}_{\mathcal{B}} = \{\pi' = \pi^\frown q_{\mathcal{B}}, a, q'_{\mathcal{B}} \in Execs(\mathcal{B}) | \pi \in \tilde{E}_{\mathcal{B}}, \exists \alpha \in (f^{ex})^{-1}(\pi) \cap E'_{\mathcal{A}}, q_{\mathcal{A}} = lstate(\alpha), (q_{\mathcal{A}}, a, \eta_{\mathcal{A}, q_{\mathcal{A}}, a}) \in dom(f^{tr,+})\}$ is surjective.*

Proof. Let $\pi' \in \tilde{E}_{\mathcal{B}}$. We have $\pi' = \pi^\frown q_{\mathcal{B}}, a, q'_{\mathcal{B}} \in Execs(\mathcal{B})$ s. t. $\pi \in \tilde{E}_{\mathcal{B}}$ and $\exists \alpha \in (f^{ex})^{-1}(\pi) \cap E'_{\mathcal{A}}, q_{\mathcal{A}} = lstate(\alpha)$ and $(q_{\mathcal{A}}, a, \eta_{(\mathcal{A}, q_{\mathcal{A}}, a)}) \in dom(f^{tr,+})$. By $(q_{\mathcal{A}}, a, \eta_{\mathcal{A}, q_{\mathcal{A}}, a}) \in dom(f^{tr,+})$, if i) $(q_{\mathcal{A}}, a, \eta_{\mathcal{A}, q_{\mathcal{A}}, a}) \in dom(f^{tr,+}) \setminus dom(f^{tr})$ $\eta_{\mathcal{A}, q_{\mathcal{A}}, a} \overset{f^+}{\longleftrightarrow} \eta_{\mathcal{B}, q_{\mathcal{B}}, a}$ and if ii) $(q_{\mathcal{A}}, a, \eta_{\mathcal{A}, q_{\mathcal{A}}, a}) \in dom(f^{tr})$ $\eta_{\mathcal{A}, q_{\mathcal{A}}, a} \overset{f}{\longleftrightarrow} \eta_{\mathcal{B}, q_{\mathcal{B}}, a}$. In both cases, it exists $q'_{\mathcal{A}} \in supp(\eta_{\mathcal{A}, q_{\mathcal{A}}, a})$ s. t. $f^{ex,+}(\alpha' = \alpha^\frown q_{\mathcal{A}}, a, q'_{\mathcal{A}}) = \pi'$ with $\alpha' \in E'^{+}_{\mathcal{A}}$.

□

We finish this paragraph with the concept of semantic equivalence that describes a pair of PSIOA that differ only syntactically.

Definition 7.13 (semantic equivalence). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA. We say that $\mathcal{A}$ and $\mathcal{B}$ are semantically-equivalent if it exists $f : Execs(\mathcal{A}) \rightarrow Execs(\mathcal{B})$ which is a complete bijective executions-matching from $\mathcal{A}$ to $\mathcal{B}$.

Composability of execution-matching relationship. Now we are looking for composability of executions-matching. First we define natural extension of notions presented in previous paragraph for the automaton obtained after composition with another automaton $\mathcal{E}$.

Definition 7.14 ($\mathcal{E}$-extension). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA with $Q_{\mathcal{A}} = states(\mathcal{A})$ and $Q_{\mathcal{B}} = states(\mathcal{B})$ as respective sets of states and let $\mathcal{E}$ be partially-compatible with both $\mathcal{A}$ and $\mathcal{B}$.

(1) Let $Q'_{\mathcal{A}} \subset Q_{\mathcal{A}}$. We call *$\mathcal{E}$-extension* of $Q'_{\mathcal{A}}$ the set of states $Q'_{\mathcal{A}||\mathcal{E}} = \{q \in states(\mathcal{A}||\mathcal{E}) | q \upharpoonright \mathcal{A} \in Q'_{\mathcal{A}}\}$
(2) Let $f : Q'_{\mathcal{A}} \subset Q_{\mathcal{A}} \rightarrow Q_{\mathcal{B}}$. We call $\mathcal{E}$-extension of f the function $g : Q'_{\mathcal{A}||\mathcal{E}} \rightarrow states(\mathcal{B}) \times states(\mathcal{E})$ s. t. $\forall (q_{\mathcal{A}}, q_{\mathcal{E}}) \in Q'_{\mathcal{A}||\mathcal{E}}, g((q_{\mathcal{A}}, q_{\mathcal{E}})) = (f(q_{\mathcal{A}}), q_{\mathcal{E}}))$
(3) Let $D'_{\mathcal{A}} \subset D_{\mathcal{A}}$ a subset of transitions. We call $\mathcal{E}$-extension of $D'_{\mathcal{A}}$ the set $D'_{\mathcal{A}||\mathcal{E}} = \{((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{((\mathcal{A},\mathcal{E}),(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in dtrans(\mathcal{A}||\mathcal{E}) | q_{\mathcal{A}} \in Q'_{\mathcal{A}}$ and either $(q_{\mathcal{A}}, a, \eta_{(\mathcal{A},q_{\mathcal{A}},a)}) \in D'_{\mathcal{A}}$ or the action a is not enabled in $q_{\mathcal{A}}\}$.

Now, we can start with the composability of states-matching.

Lemma 7.15 (Composability of states-matching). *Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA with $Q_{\mathcal{A}} = states(\mathcal{A})$ and $Q_{\mathcal{B}} = states(\mathcal{B})$ as respective sets of states. Let $\mathcal{E}$ be partially-compatible with $\mathcal{A}$ and $\mathcal{B}$. Let $f : Q'_{\mathcal{A}} \subset Q_{\mathcal{A}} \rightarrow Q_{\mathcal{B}}$ be a states-matching. Let g be the $\mathcal{E}$-extension of f.*

If $range(g) \subset states(\mathcal{B}||\mathcal{E})$, then g is a states-matching from $\mathcal{A}||\mathcal{E}$ to $\mathcal{B}||\mathcal{E}$.

Proof.
- Starting state preservation: if $(\bar{q}_{\mathcal{A}}, \bar{q}_{\mathcal{E}}) \in Q_{\mathcal{A}||\mathcal{E}}$ then $\bar{q}_{\mathcal{A}} \in Q'_{\mathcal{A}}$ which means $f(\bar{q}_{\mathcal{A}}) = \bar{q}_{\mathcal{B}}$, thus $g((\bar{q}_{\mathcal{A}}, \bar{q}_{\mathcal{E}})) = (\bar{q}_{\mathcal{B}}, \bar{q}_{\mathcal{E}})$.
- Signature preservation (modulo an hiding operation): $\forall ((q_{\mathcal{A}}, q_{\mathcal{E}}), (q_{\mathcal{B}}, q_{\mathcal{E}})) \in Q'_{\mathcal{A}||\mathcal{E}} \times states(\mathcal{B}||\mathcal{E})$ with $(q_{\mathcal{B}}, q_{\mathcal{E}}) = g((q_{\mathcal{A}}, q_{\mathcal{E}}))$, we have $sig(\mathcal{A})(q_{\mathcal{A}}) = sig(\mathcal{B})(f(q_{\mathcal{A}})) = hide(sig(\mathcal{B})(q_{\mathcal{B}}), h(q_{\mathcal{B}}))$ with $h(q_{\mathcal{B}}) \subseteq out(\mathcal{B})(q_{\mathcal{B}})$.
  Since $\mathcal{A}$ and $\mathcal{E}$ are partially-compatible, $sig(\mathcal{A})(q_{\mathcal{A}}) = hide(sig(\mathcal{B})(q_{\mathcal{B}}), h(q_{\mathcal{B}}))$ is compatible with $sig(\mathcal{E})(q_{\mathcal{E}})$ which means a fortiori $sig(\mathcal{B})(q_{\mathcal{B}})$ is compatible with $sig(\mathcal{E})(q_{\mathcal{E}})$.
  Namely $\forall act \in h(q_{\mathcal{B}}), act \notin in(\mathcal{E})(q_{\mathcal{E}})$. Hence $sig((\mathcal{A}, \mathcal{E}))((q_{\mathcal{A}}), q_{\mathcal{E}})) = hide(sig((\mathcal{B}, \mathcal{E}))((q_{\mathcal{B}}, q_{\mathcal{E}})), h'((q_{\mathcal{B}}, q_{\mathcal{E}}))$ with $h'((q_{\mathcal{B}}, q_{\mathcal{E}})) = h(q_{\mathcal{B}}) \subseteq out(\mathcal{B})(q_{\mathcal{B}}) \subseteq out(\mathcal{B}||)((q_{\mathcal{B}}, q_{\mathcal{E}}))$ which ends the proof.

□

The composability of states-matching is ensured under the condition $range(g) \subset states(\mathcal{B}||\mathcal{E})$ where g is the $\mathcal{E}$-extension of the original states-matching $f : Q'_{\mathcal{A}} \subseteq states(\mathcal{A}) \rightarrow states(\mathcal{B})$. In next lemma, we give a sufficient condition to ensure $range(g) \subset states(\mathcal{B}||\mathcal{E})$. This is the one that we will use in practice.

Definition 7.16 (reachable-by and states of execution (recall)). Let $\mathcal{A}$ be a PSIOA or a PCA. Let $E'_{\mathcal{A}} \subseteq Execs(\mathcal{A})$. We note $reachable - by(E_{\mathcal{A}}) = \{q \in states(\mathcal{A}) | \exists \alpha \in E'_{\mathcal{A}}, lstate(\alpha) = q\}$. Let $\alpha = q^0, a^1, q^1, ...a^n, q^n, ...$ We note $states(\alpha) = \bigcup_{i \in |\alpha|} q^i$.

Lemma 7.17 (A sufficient condition to obtain $range(g) \subset states(\mathcal{B}||\mathcal{E})$). *Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA with $Q_{\mathcal{A}} = states(\mathcal{A})$ and $Q_{\mathcal{B}} = states(\mathcal{B})$ as respective sets of states. Let $\mathcal{E}$ be partially-compatible with both $\mathcal{A}$ and $\mathcal{B}$. Let $f : Q'_{\mathcal{A}} \subset Q_{\mathcal{A}} \rightarrow Q_{\mathcal{B}}$ be a states-matching. Let $Q'_{\mathcal{A}||\mathcal{E}}$ be the $\mathcal{E}$-extension of $Q'_{\mathcal{A}}$.*

Let $Q''_{\mathcal{A}||\mathcal{E}} \subset Q'_{\mathcal{A}||\mathcal{E}}$ the set of states reachable by an execution that counts only states in $Q'_{\mathcal{A}||\mathcal{E}}$, i. e.

- $E''_{\mathcal{A}||\mathcal{E}} = \{\alpha \in Execs(\mathcal{A}||\mathcal{E}) | states(\alpha) \subseteq Q'_{\mathcal{A}||\mathcal{E}}\}$
- $Q''_{\mathcal{A}||\mathcal{E}} = reachable\text{-}by(E''_{\mathcal{A}||\mathcal{E}})$

Let f'' the restriction of f to set $Q''_{\mathcal{A}} = \{q_{\mathcal{A}} = ((q_{\mathcal{A}}, q_{\mathcal{E}}) \upharpoonright \mathcal{A}) | (q_{\mathcal{A}}, q_{\mathcal{E}}) \in Q''_{\mathcal{A}||\mathcal{E}}\}$.
Then the $\mathcal{E}$-extension of f'', noted g'' verifies $range(g'') \subset states(\mathcal{B}||\mathcal{E})$.

Proof. By induction on the minimum size of an execution $\tilde{\alpha} = q^0 a^1 ... q^n$ with $q^* = q^n, \forall i \in [0, n], q^i \in Q'_{\mathcal{A}||\mathcal{E}}$. Basis ($|\alpha| = 0 \implies \alpha = \bar{q}_{\mathcal{A}}$): we consider $q^* = \bar{q}_{\mathcal{A}}$. We have $g((\bar{q}_{\mathcal{A}}, \bar{q}_{\mathcal{E}})) = (f(\bar{q}_{\mathcal{A}}), \bar{q}_{\mathcal{E}}) = (\bar{q}_{\mathcal{B}}, \bar{q}_{\mathcal{E}}) \in states(\mathcal{B}||\mathcal{E})$.

We assume this is true for $\tilde{\alpha}$ with $lstate(\tilde{\alpha}) = q$ and we show this is also true for $\tilde{\alpha}' = \tilde{\alpha}^\frown qaq'$. By induction hypothesis $q \in states(\mathcal{B}||\mathcal{E})$. Since $q' \in states(\mathcal{A}||\mathcal{E})$, $\mathcal{A}$ and $\mathcal{E}$ are compatible at state $(q'_{\mathcal{A}}, q'_{\mathcal{E}})$, that is $sig(\mathcal{A})(q'_{\mathcal{A}})$ and $sig(\mathcal{E})(q'_{\mathcal{E}})$ are compatible, which means that a fortiori, $(sig(\mathcal{B})(f''(q'_{\mathcal{A}}))$ and $sig(\mathcal{E})(q'_{\mathcal{E}})$ are compatible and so $\mathcal{B}$ and $\mathcal{E}$ are compatible at state $(f''(q'_{\mathcal{A}}), q'_{\mathcal{E}}) = g''(q')$. Hence $g''(q')$ is a reachable compatible state of $(\mathcal{B}, \mathcal{E})$ which means this is a state of $\mathcal{B}||\mathcal{E}$.

□

Now, we can continue with the composability of transitions-matching.

Lemma 7.18 (Composability of eligibility for transitions-matching). *Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA with $Q_{\mathcal{A}} = states(\mathcal{A})$ and $Q_{\mathcal{B}} = states(\mathcal{B})$ as respective sets of states. Let $\mathcal{E}$ be partially-compatible with $\mathcal{A}$ and $\mathcal{B}$. Let $f : Q'_{\mathcal{A}} \subset Q_{\mathcal{A}} \to Q_{\mathcal{B}}$ be a states-matching and $D'_{\mathcal{A}}$ a subset of transitions eligible to transitions-matching domain from f. Let g be the $\mathcal{E}$-extension of f and $D'_{\mathcal{A}||\mathcal{E}}$ the $\mathcal{E}$-extension of $D_{\mathcal{A}}$.*
If $range(g) \subset states(\mathcal{B}||\mathcal{E})$, then $D'_{\mathcal{A}||\mathcal{E}}$ is eligible to transitions-matching domain from g.

Proof. Let $((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{((\mathcal{A},\mathcal{E}),(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in D'_{\mathcal{A}||\mathcal{E}}$.
By definition, $q_{\mathcal{A}} \in Q'_{\mathcal{A}}$ which means $(q_{\mathcal{A}}, q_{\mathcal{E}}) \in Q'_{\mathcal{A}||\mathcal{E}}$, so the matched states preservation is ensured. We still need to ensure the equitable corresponding distribution.

- Let $(q''_{\mathcal{A}}, q''_{\mathcal{E}}) \in supp(\eta_{((\mathcal{A},\mathcal{E}),(q_{\mathcal{A}},q_{\mathcal{E}}),a)})$. If $a \in \widehat{sig}(\mathcal{A})(q_{\mathcal{A}})$, then $q''_{\mathcal{A}} \in supp(\eta_{(\mathcal{A},q_{\mathcal{A}},a)})$ which means $q''_{\mathcal{A}} \in Q'_{\mathcal{A}}$ and hence $(q''_{\mathcal{A}}, q''_{\mathcal{E}}) \in Q'_{\mathcal{A}||\mathcal{E}}$. If $a \notin \widehat{sig}(\mathcal{A})$, $\eta_{(\mathcal{A},q_{\mathcal{A}},a)} = \delta_{q_{\mathcal{A}}}$, which means $q''_{\mathcal{A}} = q_{\mathcal{A}} \in Q'_{\mathcal{A}}$ and hence $(q''_{\mathcal{A}}, q''_{\mathcal{E}}) \in Q'_{\mathcal{A}||\mathcal{E}}$. Thus for every $(q''_{\mathcal{A}}, q''_{\mathcal{E}}) \in supp(\eta_{((\mathcal{A},\mathcal{E}),(q_{\mathcal{A}},q_{\mathcal{E}}),a)})$, $(q''_{\mathcal{A}}, q''_{\mathcal{E}}) \in Q'_{\mathcal{A}||\mathcal{E}}$.
- $\eta_{((\mathcal{A},\mathcal{E}),(q_{\mathcal{A}},q_{\mathcal{E}}),a)}((q''_{\mathcal{A}}, q''_{\mathcal{E}})) = \eta_{(\mathcal{A},q_{\mathcal{A}},a)} \otimes \eta_{(\mathcal{E},q_{\mathcal{E}},a)}(q''_{\mathcal{A}}, q''_{\mathcal{E}}) = \eta_{(\mathcal{A},q_{\mathcal{A}},a)}(q''_{\mathcal{A}}) \cdot \eta_{(\mathcal{E},q_{\mathcal{E}},a)}(q''_{\mathcal{E}}) = \eta_{(\mathcal{B},f(q_{\mathcal{A}}),a)}(f(q''_{\mathcal{A}})) \cdot \eta_{(\mathcal{E},q_{\mathcal{E}},a)}(q''_{\mathcal{E}}) = \eta_{(\mathcal{B},f(q_{\mathcal{A}}),a)} \otimes \eta_{(\mathcal{E},q_{\mathcal{E}},a)}(f(q''_{\mathcal{A}}), q''_{\mathcal{E}}) = \eta_{((\mathcal{B},\mathcal{E}),g(q_{\mathcal{A}},q_{\mathcal{E}}),a)}(g(q''_{\mathcal{A}}, q''_{\mathcal{E}}))$ which ends the proof of equitable corresponding distribution.

□

Definition 7.19 ($\mathcal{E}$-extension of an execution-matching). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA with $Q_{\mathcal{A}} = states(\mathcal{A})$ and $Q_{\mathcal{B}} = states(\mathcal{B})$ as respective sets of states. Let $\mathcal{E}$ be partially-compatible with both $\mathcal{A}$ and $\mathcal{B}$. Let (f, f^{tr}, f^{ex}) be an executions-matching from $\mathcal{A}$ to $\mathcal{B}$. Let g the $\mathcal{E}$-extension of f. If $range(g) \subset states(\mathcal{B}||\mathcal{E})$, then

(1) we call the $\mathcal{E}$-extension of f^{tr} the function $g^{tr} : (q, a, \eta_{(\mathcal{A}||\mathcal{E},q,a)}) \in D'_{\mathcal{A}||\mathcal{E}} \mapsto (g(q), a, \eta_{(\mathcal{B}||\mathcal{E},g(q),a)})$ where $D'_{\mathcal{A}||\mathcal{E}}$ is the $\mathcal{E}$-extension of the domain $dom(f^{tr})$ of f^{tr}.
(2) we call the $\mathcal{E}$-extension of (f, f^{tr}, f^{ex}) the matching-execution (g, g^{tr}, g^{ex}) from $\mathcal{A}||\mathcal{E}$ to $\mathcal{B}||\mathcal{E}$ induced by g and $dom(g^{tr})$.

Finally we can states the main result of this paragraph, i. e. theorem 7.20 of executions-matching composability.

Theorem 7.20 (Composability of executions-matching). *Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA. Let $\mathcal{E}$ be partially-compatible with both $\mathcal{A}$ and $\mathcal{B}$. Let (f, f^{tr}, f^{ex}) be an execution-matching from $\mathcal{A}$ to $\mathcal{B}$ where g represents the $\mathcal{E}$-extension of f. If $range(g) \subset states(\mathcal{B}||\mathcal{E})$, then the $\mathcal{E}$-extension of (f, f^{tr}, f^{ex}) is a matching-execution (g, g^{tr}, g^{ex}) from $\mathcal{A}||\mathcal{E}$ to $\mathcal{B}||\mathcal{E}$ induced by g and $dom(g^{tr})$.*

Proof. We repeated the previous definition, while an executions-matching only need a states-matching g and a set $dom(g^{tr})$ of transitions eligible to transitions-matching domain from g which is provided by construction. □

Here we give some properties preserved by $\mathcal{E}$-extension of an executions-matching.

Lemma 7.21 (Some properties preserved by $\mathcal{E}$-extension of an executions-matching). *Let $\mathcal{A}$ (resp. $\mathcal{B}$) be a PSIOA with $Q_{\mathcal{A}}$ (resp. $Q_{\mathcal{B}}$) as set of states. Let (f, f^{tr}, f^{ex}) be an execution-matching from $\mathcal{A}$ to $\mathcal{B}$.*

1. *If f is bijective and f^{-1} is complete, then for every PSIOA $\mathcal{E}$ partially-compatible with $\mathcal{A}$, $\mathcal{E}$ is partially-compatible with $\mathcal{B}$.*
2. *Let $\mathcal{E}$ partially-compatible with both $\mathcal{A}$ and $\mathcal{B}$, let g be the $\mathcal{E}$-extension of f.*
   (a) *If f is bijective and f^{-1} is complete, then $range(g) = states(\mathcal{B}||\mathcal{E})$ and so we can talk about the $\mathcal{E}$-extension of (f, f^{tr}, f^{ex})*
   (b) *If (f, f^{tr}) is a bijective complete transition-matching, (g, g^{tr}) is a bijective complete transition-matching. (And (f, f^{tr}, f^{ex}) and (g, g^{tr}, g^{ex}) are bijective complete execution-matching.)*
   (c) *If f is strong, then g is strong*
3. *Let $\mathcal{E}$ partially-compatible with both $\mathcal{A}$ and $\mathcal{B}$, let g be the $\mathcal{E}$-extension of f. Let assume $range(g) \subseteq states(\mathcal{B}||\mathcal{E})$. Let (g, g^{tr}, g^{ex}) be the $\mathcal{E}$-extension of (f, f^{tr}, f^{ex})*
   (a) *If the restriction $\tilde{f}^{ex} : E'_{\mathcal{A}} \subseteq Execs(\mathcal{A}) \to \tilde{E}_{\mathcal{B}} \subseteq Execs(\mathcal{B})$ is surjective, then $\tilde{g}^{ex} : \{\alpha \in Execs(\mathcal{A}||\mathcal{E}) | \alpha \upharpoonright \mathcal{A} \in E'_{\mathcal{A}}\} \to \{\pi \in Execs(\mathcal{B}||\mathcal{E}) | \pi \upharpoonright \mathcal{B} \in \tilde{E}_{\mathcal{B}}\}$ is surjective*
   (b) *If f is strong, g is strong.*

Proof. (1) We need to show that every pseudo-execution of $(\mathcal{B}, \mathcal{E})$ ends on a compatible state. Let $\pi = q^0 a^1 q^1 ... a^n q^n$ be a finite pseudo-execution of $(\mathcal{B}, \mathcal{E})$. We note $\alpha = (f^{-1}(q^0_{\mathcal{B}}), q^0_{\mathcal{E}}) a^1 (f^{-1}(q^1_{\mathcal{B}}), q^1_{\mathcal{E}}) ... a^n (f^{-1}(q^n_{\mathcal{B}}), q^n_{\mathcal{E}})$. The proof is in two steps. First, we show by induction that $\alpha = (f^{-1}(q^0_{\mathcal{B}}), q^0_{\mathcal{E}}) a^1 (f^{-1}(q^1_{\mathcal{B}}), q^1_{\mathcal{E}}) ... a^n (f^{-1}(q^n_{\mathcal{B}}), q^n_{\mathcal{E}})$ is an execution of $\mathcal{A}||\mathcal{E}$. Second, we deduce that it means $(f^{-1}(q^n_{\mathcal{B}}), q^n_{\mathcal{E}})$ is a compatible state of $(\mathcal{A}, \mathcal{E})$ which means that a fortiori, $(q^n_{\mathcal{B}}, q^n_{\mathcal{E}})$ is a compatible state of $(\mathcal{B}, \mathcal{E})$ which ends the proof.

- First, we show by induction that α is an execution of $\mathcal{A}||\mathcal{E}$. We have $(f^{-1}(\bar{q}_{\mathcal{B}}), \bar{q}_{\mathcal{E}}) = (\bar{q}_{\mathcal{A}}, \bar{q}_{\mathcal{E}})$ which ends the basis.
  Let assume $(f^{-1}(q^0_{\mathcal{B}}), q^0_{\mathcal{E}}) a^1 (f^{-1}(q^1_{\mathcal{B}}), q^1_{\mathcal{E}}) ... a^k (f^{-1}(q^k_{\mathcal{B}}), q^k_{\mathcal{E}})$ is an execution of $\mathcal{A}||\mathcal{E}$. Hence $(f^{-1}(q^k_{\mathcal{B}}), q^k_{\mathcal{E}})$ is a compatible state of $(\mathcal{A}, \mathcal{E})$ which means that a fortiori q^k is a compatible state of $(\mathcal{B}, \mathcal{E})$ because of signature preservation of f.
  For the same reason, $\widehat{sig}(\mathcal{B}||\mathcal{E})(q^k) = \widehat{sig}(\mathcal{A}, \mathcal{E})((f^{-1}(q^k_{\mathcal{B}}), q^k_{\mathcal{E}}))$, so $a^{k+1} \in \widehat{sig}(\mathcal{A}, \mathcal{E})((f^{-1}(q^k_{\mathcal{B}}), q^k_{\mathcal{E}}))$. Then we use the completeness of $(f^{-1}, (f^{tr})^{-1})$, to obtain the fact that either $\eta_{(\mathcal{B}, q^k_{\mathcal{B}}, a^{k+1})} \in dom((f^{tr})^{-1})$ or $a^{k+1} \notin \widehat{sig(\mathcal{B})}(q^k_{\mathcal{B}})$ (and we recall the convention that in this second case $\eta_{(\mathcal{B}, q^k_{\mathcal{B}}, a^{k+1})} = \delta_{q^k_{\mathcal{B}}}$). which means either $(f^{-1}(q^k_{\mathcal{B}}), a^{k+1}, \eta_{(\mathcal{A}, f^{-1}(q^k_{\mathcal{B}}), a^{k+1})})$ is a transition of $\mathcal{A}$ that ensures $\forall q'' \in supp(\eta_{(\mathcal{B}, q^k_{\mathcal{B}}, a^{k+1})}), f^{-1}(q'') \in supp(\eta_{(\mathcal{A}, f^{-1}(q^k_{\mathcal{B}}), a^{k+1})})$ or $a^{k+1} \notin \widehat{sig(\mathcal{A})}(f^{-1}(q^k_{\mathcal{B}}))$ (and we recall the convention that in this second case $\eta_{(\mathcal{A}, f^{-1}(q^k_{\mathcal{B}}), a^{k+1})} = \delta_{f^{-1}(q^k_{\mathcal{B}})}$). Thus for every $(q'', q''') \in supp(\eta_{(\mathcal{B}, \mathcal{E}), q^k, a^{k+1})})$, $(f^{-1}(q''), q''') = g^{-1}((q'', q''')) \in supp(\eta_{(\mathcal{A}, \mathcal{E}), g^{-1}(q^k), a^{k+1})})$ namely for $(q'', q''') = (q^{k+1}_{\mathcal{B}}, q^{k+1}_{\mathcal{E}})$. Hence, $(f^{-1}(q^{k+1}_{\mathcal{B}}), q^{k+1}_{\mathcal{E}})$ is reachable by $(\mathcal{A}, \mathcal{E})$ which means $(f^{-1}(q^0_{\mathcal{B}}), q^0_{\mathcal{E}}) a^1 (f^{-1}(q^1_{\mathcal{B}}), q^1_{\mathcal{E}}) ... a^k (f^{-1}(q^k_{\mathcal{B}}), q^k_{\mathcal{E}}) a^k (f^{-1}(q^k_{\mathcal{B}}), q^k_{\mathcal{E}}) a^{k+1} (f^{-1}(q^{k+1}_{\mathcal{B}}), q^{k+1}_{\mathcal{E}})$ is an execution of $\mathcal{A}||\mathcal{E}$. Thus by induction α is an execution of $\mathcal{A}||\mathcal{E}$.

- Since $\mathcal{A}$ and $\mathcal{E}$) are partially-compatible $(f^{-1}(q^n_{\mathcal{B}}), q^n_{\mathcal{E}})$ is a state of $\mathcal{A}||\mathcal{E}$, so $(f^{-1}(q^n_{\mathcal{B}}), q^n_{\mathcal{E}})$ is a compatible state of $(\mathcal{A}, \mathcal{E})$ which means $(q^k_{\mathcal{B}}), q^k_{\mathcal{E}})$ is a fortiori a compatible state of $(\mathcal{B}, \mathcal{E})$. Hence every reachable state of $(\mathcal{B}, \mathcal{E})$ is compatible which means $\mathcal{B}$ and $\mathcal{E}$) are partially compatible which ends the proof.

(2) (a)
- Let $(q^n_{\mathcal{B}}, q^n_{\mathcal{E}}) \in states(\mathcal{B}||\mathcal{E})$. This state is reachable, so we note $\pi = (q^0_{\mathcal{B}}, q^0_{\mathcal{E}})a^1(q^1_{\mathcal{B}}, q^1_{\mathcal{E}})...a^n(q^n_{\mathcal{B}}, q^n_{\mathcal{E}})$ the execution of $\mathcal{B}||\mathcal{E}$. Thereafter, we note $\alpha = (f^{-1}(q^0_{\mathcal{B}}), q^0_{\mathcal{E}})a^1(f^{-1}(q^1_{\mathcal{B}}), q^1_{\mathcal{E}})...a^n(f^{-1}(q^n_{\mathcal{B}}), q^n_{\mathcal{E}})$. We can show by induction that α is an execution of $\mathcal{A}||\mathcal{E}$. The proof is exactly the same than in 1. Hence α is an execution of $\mathcal{A}||\mathcal{E}$ which means $(f^{-1}(q^n_{\mathcal{B}}), q^n_{\mathcal{E}})$ is a state of $\mathcal{A}||\mathcal{E}$ and then $g((f^{-1}(q^n_{\mathcal{B}}), q^n_{\mathcal{E}})) = (q^n_{\mathcal{B}}, q^n_{\mathcal{E}})$ to finally prove that it exists q^* s. t. $g(q^*) = (q^n_{\mathcal{B}}, q^n_{\mathcal{E}})$ which means $states(\mathcal{B}||\mathcal{E}) \subseteq dom(g)$. We can reuse the proof of 1. to show that if $q \in states(\mathcal{A}||\mathcal{E})$, then $g(q) \in states(\mathcal{B}||\mathcal{E})$ which means $dom(g) \subseteq states(\mathcal{B}||\mathcal{E})$.
  Hence $dom(g) = states(\mathcal{B}||\mathcal{E})$.
- We can apply the previous lemma 7.18 to obtain the eligibility of $D_{\mathcal{A}||\mathcal{E}}$.

(b) Let assume (f, f^{tr}) are bijective. The bijectivity of g is immediate $g(.,.) = (f(.), Id(.))$. The bijectivity of g^{tr} is also immediate since $g^{tr} : \eta_{(\mathcal{A},q_{\mathcal{A}},a)} \otimes \eta_{(\mathcal{E},q_{\mathcal{E}},a)} \rightarrow f^{tr}(\eta_{(\mathcal{A},q_{\mathcal{A}},a)}) \otimes \eta_{(\mathcal{E},q_{\mathcal{E}},a)}$ with f^{tr} bijective.

(c) Immediate, since in this case $sig(\mathcal{A})(q_{\mathcal{A}}) = sig(\mathcal{B})(f(q_{\mathcal{A}}))$ implies $sig(\mathcal{A}||\mathcal{E})((q_{\mathcal{A}}, q_{\mathcal{E}})) = sig(\mathcal{B}||\mathcal{E})((f(q_{\mathcal{A}}), q_{\mathcal{E}}))$.

(3) (a) Let $\pi = ((q^0_{\mathcal{B}}, q^0_{\mathcal{E}}), a^1, (q^1_{\mathcal{B}}, q^1_{\mathcal{E}}), ..., a^n, (q^n_{\mathcal{B}}, q^n_{\mathcal{E}})) \in Execs(\mathcal{B}||\mathcal{E})$ with $\pi \upharpoonright \mathcal{B} = \hat{q}^0_{\mathcal{B}}, \hat{a}^1, \hat{q}^1_{\mathcal{B}}, ..., \hat{a}^m, \hat{q}^m_{\mathcal{B}} \in \tilde{E}_{\mathcal{B}}$, where the monotonic function $k : [0, n] \rightarrow [0, m]$, verifies $\forall i \in [0, n], k(i) \in [0, m], q^i_{\mathcal{B}} = \hat{q}^{k(i)}_{\mathcal{B}}$
By surjectivity of f^{ex} we have $\hat{\alpha} = \hat{q}^0_{\mathcal{A}}, \hat{a}^1, \hat{q}^1_{\mathcal{A}}, ..., \hat{a}^m, \hat{q}^m_{\mathcal{A}} \in E'_{\mathcal{A}}$ s. t. $f^{ex}(\hat{\alpha}) = \pi \upharpoonright \mathcal{B}$. We note $\alpha = (q^0_{\mathcal{A}}, q^0_{\mathcal{E}})a^1(q^1_{\mathcal{A}}, q^1_{\mathcal{E}})...a^n(q^n_{\mathcal{A}}, q^n_{\mathcal{E}})$ where $\forall i \in [0, n], q^i_{\mathcal{A}} = \hat{q}^{k(i)}_{\mathcal{A}}$. Hence, $\forall i \in [0, n], g((q^i_{\mathcal{A}}, q^i_{\mathcal{E}})) = (q^i_{\mathcal{B}}, q^i_{\mathcal{E}})$. Moreover, by signature preservation, $\forall i \in [0, n-1], a^{i+1} \in \widehat{sig}(\mathcal{A})(q^i_{\mathcal{A}}) \cup \widehat{sig}(\mathcal{E})(q^i_{\mathcal{E}})$. Furthermore, $\forall i \in [0, n-1]$. $(q^{i+1}_{\mathcal{A}}, q^{i+1}_{\mathcal{E}}) \in supp(\eta_{(\mathcal{A},q^i_{\mathcal{A}},a^i)} \otimes \eta_{(\mathcal{B},q^i_{\mathcal{B}},a^i)})$ since $(q^{i+1}_{\mathcal{B}}, q^{i+1}_{\mathcal{E}}) \in supp(\eta_{(\mathcal{B},q^i_{\mathcal{B}},a^i)} \otimes \eta_{(\mathcal{B},q^i_{\mathcal{B}},a^i)})$, $(q^i_{\mathcal{B}}, a^i, \eta_{(\mathcal{B},q^i_{\mathcal{B}},a^i)}) = f^{tr}(q^i_{\mathcal{A}}, a^i, \eta_{(\mathcal{A},q^i_{\mathcal{A}},a^i)})$ and $q^{i+1}_{\mathcal{B}} = f(q^{i+1}_{\mathcal{A}})$. Thus, $\alpha \in Execs(\mathcal{A}||\mathcal{E})$. Finally, by signature preservation of f, $\forall i \in [1, n] \widehat{sig}(\mathcal{A})(q_{\mathcal{A}}) = \widehat{sig}(\mathcal{B})(q_{\mathcal{B}})$, which lead us to $\alpha \upharpoonright \mathcal{A} = \hat{\alpha} \in E'_{\mathcal{A}}$. So for every $\pi \in Execs(\mathcal{B}||\mathcal{E})$ with $\pi \upharpoonright \mathcal{B} \in \tilde{E}_{\mathcal{B}}$, it exists $\alpha \in Execs(\mathcal{A}||\mathcal{E})$ with $\alpha \upharpoonright \mathcal{A} \in E'_{\mathcal{A}}$ which ends the proof.

(b) Immediate by rules of composition of signature: $\forall (q_{\mathcal{A}}, q_{\mathcal{E}}) \in states(\mathcal{A}||\mathcal{E}), \forall (q_{\mathcal{B}}, q_{\mathcal{E}}) \in states(\mathcal{B}||\mathcal{E})$ if $sig(\mathcal{A})(q_{\mathcal{A}}) = sig(\mathcal{B})(q_{\mathcal{B}})$, then $sig(\mathcal{A}||\mathcal{E})(q_{\mathcal{A}}, q_{\mathcal{E}}) = sig(\mathcal{B}||\mathcal{E}))(q_{\mathcal{B}}, q_{\mathcal{E}})$.

□

We are ready to states the composability of semantic equivalence.

Theorem 7.22 (composability of semantic equivalence). *Let $\mathcal{A}$ and $\mathcal{B}$ be PSIOA semantically-equivalent. Then for every PSIOA $\mathcal{E}$:*

- *$\mathcal{E}$ is partially-compatible with $\mathcal{A} \iff \mathcal{E}$ is partially-compatible with $\mathcal{B}$*
- *if $\mathcal{E}$ is partially-compatible with both $\mathcal{A}$ and $\mathcal{B}$, then $\mathcal{A}||\mathcal{E}$ and $\mathcal{B}||\mathcal{E}$ are semantically-equivalent PSIOA.*

Proof.
- The first item ($\mathcal{E}$ is partially-compatible with $\mathcal{A} \iff \mathcal{E}$ is partially-compatible with $\mathcal{B}$) comes from lemma 7.21, first item.
- The second item (if $\mathcal{E}$ is partially-compatible with both $\mathcal{A}$ and $\mathcal{B}$, then $\mathcal{A}||\mathcal{E}$ and $\mathcal{B}||\mathcal{E}$ are semantically-equivalent PSIOA) comes from lemma 7.21, second item.

□

A weak complete bijective transition-matching implies a weak complete bijective execution-matching which means the two automata are completely sementically equivalent modulo some hiding operation that implies that some PSIOA are partially-compatible with one of the automaton and not with the other and that the traces are not necessarily the same ones.

composition of continuation of executions-matching. Here we define $\mathcal{E}$-extension of continued executions-matching in the same way we defined $\mathcal{E}$-extension of executions-matching just before.

Definition 7.23 ($\mathcal{E}$-extension of continued executions-matching). Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA. Let $\mathcal{E}$ be partially-compatible with both $\mathcal{A}$ and $\mathcal{B}$. Let (f, f^{tr}, f^{ex}) be an executions-matching from $\mathcal{A}$ to $\mathcal{B}$. Let $((f, f^+), f^{tr,+}, f^{ex,+})$ be the $(f^+, D''_{\mathcal{A}})$-continuation of (f, f^{tr}, f^{ex}) (where by definition $D''_{\mathcal{A}} \setminus dom(f^{tr})$ respect the properties of matched states preservation and extension of equitable corresponding distribution from definition 7.7). If the respective $\mathcal{E}$-extension of f and f^+, noted g and g^+, verifie $range(g) \cup range(g^+) \subseteq (\mathcal{B}||\mathcal{E})$, we define the $\mathcal{E}$-extension of $((f, f^+), f^{tr,+}, f^{ex,+})$ as $((g, g^+), g^{tr,+}, g^{ex,+})$, where

- (g, g^{tr}, g^{ex}) is the $\mathcal{E}$-extension of (f, f^{tr}, f^e)
- $g^{tr,+} : (q, a, \eta_{(\mathcal{A}||\mathcal{E}),q,a}) \in D''_{\mathcal{A}||\mathcal{E}} \mapsto (g(q), a, \eta_{(\mathcal{A}||\mathcal{E}),g(q),a})$ where $D''_{\mathcal{A}||\mathcal{E}}$ is the $\mathcal{E}$-extension of $dom(f^{tr,+})$
- $\forall \alpha' = \alpha^\frown q, a, q'$, with $\alpha' \in dom(g^{ex})$, if $(q, a, \eta_{(\mathcal{A}||\mathcal{E}),q,a}) \in dom(g^{tr})$ $g^{ex,+}(\alpha) = g^{ex}(\alpha)$ and if $(q, a, \eta_{(\mathcal{A}||\mathcal{E}),q,a}) \in dom(g^{tr,+}) \setminus dom(g^{tr})$ $g^{ex,+}(\alpha') = g^{ex}(\alpha)^\frown g(q), a, g^+(q)$

Lemma 7.24 (Commutativity of continuation and extension). *Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA. Let $\mathcal{E}$ be partially-compatible with both $\mathcal{A}$ and $\mathcal{B}$. Let (f, f^{tr}, f^{ex}) be an executions-matching from $\mathcal{A}$ to $\mathcal{B}$. Let $((f, f^+), f^{tr,+}, f^{ex,+})$ be the $(f^+, D''_{\mathcal{A}})$-continuation of (f, f^{tr}, f^{ex}) (where by definition $D''_{\mathcal{A}}$ respect the properties of matched states preservation and extension of equitable corresponding distribution from definition 7.7). Let*

- *(g, g^{tr}, g^{ex}) be the $\mathcal{E}$-extension of (f, f^{tr}, f^e) verifying $range(g) \subseteq states(\mathcal{B}||\mathcal{E})$,*
- *$D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}}$ the $\mathcal{E}$-extension of $dom(f^{tr,+})$, i. e. $D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}} = \{((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in dtrans(\mathcal{A}||\mathcal{E}) | q_{\mathcal{A}} \in dom(f) \wedge [(q_{\mathcal{A}}, a, \eta_{(\mathcal{A},q_{\mathcal{A}},a)}) \in dom(f^{tr,+}) \vee a \notin \widehat{sig}(\mathcal{A})(q_{\mathcal{A}})]\}$.*
- *$g^+_{(c,e)}$ be the $\mathcal{E}$-extension of f^+*

Then

1. *$D''_{\mathcal{A}||\mathcal{E}} \setminus dom(g^{tr})$ verifies matched states preservation and extension of equitable corresponding distribution.*
2. *the $(g^+_{(c,e)}, (D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}}))$-continuation of (g, g^{tr}, g^{ex}), noted $((g, g^+_{(c,e)}), g^{tr,+}_{(c,e)}, g^{ex,+}_{(c,e)})$ is equal to the $\mathcal{E}$-extension of $((f, f^+), f^{tr,+}, f^{ex,+})$, noted $((g, g^+_{(e,c)}), g^{tr,+}_{(e,c)}, g^{ex,+}_{(e,c)})$.*

We show that the operation of continuation and extension are in fact commutative.

Proof. We start by showing $D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}} \setminus dom(g^{tr})$ verifies matched states preservation and extension of equitable corresponding distribution. By definition 7.7 of $\mathcal{E}$-extension, $D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}} = \{((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in dtrans(\mathcal{A}||\mathcal{E}) | q_{\mathcal{A}} \in dom(f) \wedge [(q_{\mathcal{A}}, a, \eta_{(\mathcal{A},q_{\mathcal{A}},a)}) \in dom(f^{tr,+}) \vee a \notin \widehat{sig}(\mathcal{A})(q_{\mathcal{A}})]\}$, while $dom(g^{tr}) = \{((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in dtrans(\mathcal{A}||\mathcal{E}) | q_{\mathcal{A}} \in dom(f) \wedge [(q_{\mathcal{A}}, a, \eta_{(\mathcal{A},q_{\mathcal{A}},a)}) \in dom(f^{tr}) \vee a \notin \widehat{sig}(\mathcal{A})(q_{\mathcal{A}})]\}$.

Thus $D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}} \setminus dom(g^{tr}) = \{((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in dtrans(\mathcal{A}||\mathcal{E}) | q_{\mathcal{A}} \in dom(f) \wedge [(q_{\mathcal{A}}, a, \eta_{(\mathcal{A},q_{\mathcal{A}},a)}) \in dom(f^{tr,+}) \setminus dom(f^{tr})]\}$ (*)

Let $tr = ((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E}),(q_{\mathcal{A}},q_{\mathcal{E}}),a}) \in D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}} \setminus dom(g^{tr})$, then

- Matched states preservation: By (*) $q_{\mathcal{A}} \in dom(f)$ which leads immediately to $(q_{\mathcal{A}}, q_{\mathcal{E}}) \in dom(g)$

- Extension of equitable corresponding distribution: $\forall (q''_{\mathcal{A}}, q''_{\mathcal{E}}) \in supp(\eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)}), (q''_{\mathcal{A}}, q''_{\mathcal{E}}) \in supp(\eta_{(\mathcal{A}q_{\mathcal{A}},a)} \otimes \eta_{(\mathcal{E},q_{\mathcal{E}},a)})$ with $\eta_{(\mathcal{A}q_{\mathcal{A}},a)} \in dom(f^{tr,+}) \setminus dom(f^{tr})$ by (*) which means $q''_{\mathcal{A}} \in dom(f^+)$ and $\eta_{(\mathcal{A}q_{\mathcal{A}},a)}(q''_{\mathcal{A}}) = \eta_{(\mathcal{B}f(q_{\mathcal{A}}),a)}(f^+(q''_{\mathcal{A}}))$ and so $(q''_{\mathcal{A}}, q''_{\mathcal{E}}) \in dom(g^+)$ and $\eta_{(\mathcal{A},q_{\mathcal{A}},a)} \otimes \eta_{(\mathcal{E},q_{\mathcal{E}},a)}(q''_{\mathcal{A}}, q''_{\mathcal{E}}) = \eta_{(\mathcal{A},q_{\mathcal{A}},a)}(q''_{\mathcal{A}}) \cdot \eta_{(\mathcal{E},q_{\mathcal{E}},a)}(q''_{\mathcal{E}}) = \eta_{(\mathcal{B},f(q_{\mathcal{A}}),a)}(f^+(q''_{\mathcal{A}})) \cdot \eta_{(\mathcal{E},q_{\mathcal{E}},a)}(q''_{\mathcal{E}}) = \eta_{(\mathcal{B},f(q_{\mathcal{A}}),a)} \otimes \eta_{(\mathcal{E},q_{\mathcal{E}},a)}(f^+(q''_{\mathcal{A}}), q''_{\mathcal{E}}) = \eta_{(\mathcal{B}||\mathcal{E},g(q_{\mathcal{A}},q_{\mathcal{E}}),a)}(g^+(q''_{\mathcal{A}}, q''_{\mathcal{E}}))$

We have shown that $D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}} \setminus dom(g^{tr})$ verifies matched states preservation and extension of equitable corresponding distribution.

Now, we show the second point.

- By definition 7.7 of continuation, $g^+_{(c,e)} = g^+_{(e,c)}$.
- We prove $dom(g^{tr,+}_{(c,e)}) = dom(g^{tr,+}_{(e,c)}) = D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}}$. By definition 7.7 of continuation, $dom(g^{tr,+}_{(e,c)}) = dom(g^{tr}) \cup D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}} = \{((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in dtrans(\mathcal{A}||\mathcal{E}) | q_{\mathcal{A}} \in dom(f) \wedge [(q_{\mathcal{A}}, a, \eta_{(\mathcal{A},q_{\mathcal{A}},a)}) \in dom(f^{tr}) \vee a \notin \widehat{sig}(\mathcal{A})(q_{\mathcal{A}})]\} \cup \{((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in dtrans(\mathcal{A}||\mathcal{E}) | q_{\mathcal{A}} \in dom(f) \wedge [(q_{\mathcal{A}}, a, \eta_{(\mathcal{A},q_{\mathcal{A}},a)}) \in dom(f^{tr,+}) \vee a \notin \widehat{sig}(\mathcal{A})(q_{\mathcal{A}})]\} = \{((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in dtrans(\mathcal{A}||\mathcal{E}) | q_{\mathcal{A}} \in dom(f) \wedge [(q_{\mathcal{A}}, a, \eta_{(\mathcal{A},q_{\mathcal{A}},a)}) \in dom(f^{tr,+}) \vee a \notin \widehat{sig}(\mathcal{A})(q_{\mathcal{A}})]\} = D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}}$.
  Parrallely, by definition 7.19 of $\mathcal{E}$-extension, $dom(g^{tr,+}_{(c,e)}) = \{((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in dtrans(\mathcal{A}||\mathcal{E}) | q_{\mathcal{A}} \in dom(f) \wedge [(q_{\mathcal{A}}, a, \eta_{(\mathcal{A},q_{\mathcal{A}},a)}) \in dom(f^{tr,+}) \vee a \notin \widehat{sig}(\mathcal{A})(q_{\mathcal{A}})]\} = D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}}$. Thus $dom(g^{tr,+}_{(c,e)}) = dom(g^{tr,+}_{(e,c)}) = D''^{,(c,e)}_{\mathcal{A}||\mathcal{E}}$.
- We prove $g^{tr,+}_{(c,e)} = g^{tr,+}_{(e,c)}$ Let $((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)}) \in D''_{\mathcal{A}||\mathcal{E}}$.
  By definition 7.19 of $\mathcal{E}$-extension, $g^{tr,+}_{(c,e)}(((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)})) = (g(q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},g(q_{\mathcal{A}},q_{\mathcal{E}}),a)}))$, while by definition 7.7 of continuation, $g^{tr,+}_{(e,c)}(((q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},(q_{\mathcal{A}},q_{\mathcal{E}}),a)})) = (g(q_{\mathcal{A}}, q_{\mathcal{E}}), a, \eta_{(\mathcal{A}||\mathcal{E},g(q_{\mathcal{A}},q_{\mathcal{E}}),a)}))$.
  We can remark that properties of equitable corresponding distribution are not conflicting since $dom(g^{tr,+}_{c,e}) \setminus dom(g^{tr}) = dom(g^{tr,+}_{e,c}) \setminus dom(g^{tr})$.
- $g^{e,+}_{(e,c)}$ and $g^{e,+}_{(c,e)}$ are entirely defined by $((g, g^+_{(e,c)}), (g^{tr}, g^{tr,+}_{(e,c)}))$ and $((g, g^+_{(c,e)}), (g^{tr}, g^{tr,+}_{(c,e)}))$ that are equal.

□

application for renaming and hiding. Before dealing with PCA-executions-matching, we state two intuitive theorems of executions-matching after renaming and hiding operations.

Theorem 7.25. *(strong complete bijective execution-matching after renaming) Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA and $ren : states(\mathcal{A}) \to states(\mathcal{B})$ s. t. $\mathcal{B} = ren(\mathcal{A})$. $(ren, ren^{tr}, ren^{ex})$ is a strong complete bijective execution-matching from $\mathcal{A}$ to $\mathcal{B}$ with $dom(ren^{tr}) = D_{\mathcal{A}} = dtrans(\mathcal{A})$.*

Proof. By definition *ren* ensures starting state preservation and strong signature preservation. By definition *ren* is a complete bijection, which implies matched state preservation. The equitable corresponding distribution is also ensured by definition of *ren*. Hence, all the properties are ensured □

Theorem 7.26. *(weak complete bijective executions-matching after hiding) Let $\mathcal{A}$ be a PSIOA. Let h defined on $states(\mathcal{A})$, s. t. $\forall q \in states(\mathcal{A}), h(q) \subseteq out(\mathcal{A})(q)$. Let $\mathcal{B} = hiding(\mathcal{A}, h)$. Let Id the identity function from $states(\mathcal{A})$ to $states(\mathcal{B}) = states(\mathcal{A})$. Then (Id, Id^{tr}, Id^{ex}) is a weak complete bijective execution-matching from $\mathcal{A}$ to $\mathcal{B}$.*

Proof. By definition *Id* ensures starting state preservation and weak signature preservation. By definition *Id* is a complete bijection, which implies matched state preservation. The equitable corresponding distribution is also ensured by definition of *hiding*. Hence, all the properties are ensured □

7.2 PCA-matching execution

Here we extend the notion of executions-matching to PCA. In practice, we will build executions-matchings that preserve the sequence of configurations visited by concerned executions. Hence, the definition of PCA states-matching is slightly more restrictive to capture this notion of configuration equivalence (modulo action hiding operation), while the other definitions are exactly the same ones.

matching execution.

Definition 7.27 (PCA states-matching). Let X and Y be two PCA with $Q_X = states(X)$ and $Q_Y = states(Y)$ as respective sets of states, $(\bar{q}_X, \bar{q}_Y) = (start(X), start(Y))$ and let $f : Q'_X \subset Q_X \rightarrow Q_Y$ be a mapping s. t. :

- Starting state preservation: If $\bar{q}_X \in Q'_X$, then $f(\bar{q}_X) = \bar{q}_Y$.
- Configuration preservation (modulo hiding): $\forall (q, q') \in Q'_X \times Q_Y$, s. t. $q' = f(q)$, if $auts(config(X)(q)) = (\mathcal{A}_1, ..., \mathcal{A}_n)$, then $auts(config(Y)(q')) = (\mathcal{A}'_1, ..., \mathcal{A}'_n)$ where $\forall i \in [1 : n], \mathcal{A}_i = hide(\mathcal{A}'_i, h_i)$ with h_i defined on $states(\mathcal{A}'_i)$, s. t. $h_i(q_{\mathcal{A}'_i}) \subseteq out(\mathcal{A}'_i)(q_{\mathcal{A}'_i})$ (resp. s. t. $h_i(q_{\mathcal{A}'_i}) = \emptyset$, that is $\mathcal{A}_i = \mathcal{A}'_i$)
- Hiding preservation (modulo hiding): $\forall (q, q') \in Q'_X \times Q_Y$, s. t. $q' = f(q)$, $hidden\text{-}actions(X)(q) = hidden\text{-}actions(Y)(q') \cup h^+(q')$ where h^+ defined on $states(Y)$, s. t. $h^+(q_Y) \subseteq out(Y)(q_Y)$ (resp. s. t. $h^+(q_Y) = \emptyset$, that is $hidden\text{-}actions(X)(q) = hidden\text{-}actions(Y)(q')$)
- Creation preservation $\forall (q, q') \in Q'_X \times Q_Y$, s. t. $q' = f(q)$, $\forall a \in \widehat{sig}(X)(q) = \widehat{sig}(Y)(q')$, $created(X)(q)(a) = created(Y)(q')(a)$.

then we say that f is a *weak* (resp. *strong*) *PCA states-matching* from X to Y. If $Q'_X = Q_X$, then we say that f is a *complete* (weak or strong) PCA states-matching from X to Y.

We naturally obtain that a PCA states-matching is a PSIOA states-matching:

Lemma 7.28 (A PCA states-matching is a PSIOA states-matching). *If f is a weak (resp. strong) PCA states-matching from X to Y, then f is a PSIOA states-matching from $psioa(X)$ to $psioa(Y)$ (in the sense of definition 7.1). (The converse is not necessarily true.)*

Proof. The signature preservation immediately comes from the configuration preservation and the hiding preservation. □

Now, all the definitions from definition 7.2 to definition 7.4 of previous subsections are the same that is:

Definition 7.29 (PCA transitions-matching and PCA executions-matching). Let X and Y be two PCA with $Q_X = states(X)$ and $Q_Y = states(Y)$ as respective sets of states and let $f : Q'_X \subset Q_X \rightarrow Q_Y$ be a PCA states-matching from X to Y.

- Let $D'_X \subseteq D_X = dtrans(X)$ be a subset of transitions, D'_X is *eligible to PCA transitions-matching domain* from f if it is eligible to PSIOA transitions-matching domain from f according to definition 7.2.
- Let $D'_X \subseteq D_X = dtrans(X)$ be a subset of transitions eligible to PCA transitions-matching domain from f. We define the *PCA transitions-matching* (f, f^{tr}) induced by the PCA states-matching f and the subset of transitions D'_X as the PSIOA transitions-matching induced by the PSIOA states-matching f and the subset of transitions D'_X according to definition 7.3.

- Let $f^{tr} : D'_X \subseteq D_X = dtrans(X) \rightarrow D_Y$ s. t. (f, f^{tr}) is a PCA transitions-matching, we define the PCA executions-matching (f, f^{tr}, f^{ex}) induced by (f, f^{tr}) (resp. by f and $dom(f^{tr})$) as the PSIOA executions-matching (f, f^{tr}, f^{ex}) induced by (f, f^{tr}) (resp. by f and $dom(f^{tr})$) according to definition 7.4. Furthermore, let $(\mu, \mu') \in Disc(Frags(X)) \times Disc(Frags(Y))$ s. t. for every $\alpha' \in supp(\mu), \alpha' \in dom(f^{ex})$ and $\mu(\alpha) = \mu'(f^{ex}(\alpha'))$. then we say that (f, f^{tr}, f^{ex}) is a *PCA executions-matching* from (X, μ) to (Y, μ') according to definition 7.6.
- The (f^+, D''_X)-continuation of a PCA-executions-matching (f, f^{tr}, f^{ex}) is the (f^+, D''_X)- continuation of (f, f^{tr}, f^{ex}) in the according to definition 7.7.

We restate the theorem 7.9 and 7.10 for PCA executions-matching:

Theorem 7.30 (PCA-execution-matching preserves probabilsitic distribution). *Let X and Y be two PCA $(\mu, \mu') \in Disc(Frags(X)) \times Disc(Frags(Y))$. Let (f, f^{tr}, f^{ex}) be a PCA executions-matching from (X, μ) to (Y, μ') . Let $(\tilde{\sigma}, \sigma) \in schedulers(\mathcal{A}) \times schedulers(\mathcal{B})$, s. t. $(\tilde{\sigma}, \sigma)$ are (f, f^{tr}, f^{ex})*-alter egos. *Let $(\alpha, \pi) \in dom(f^{ex}) \times Frags(Y)$.*
If $\pi = f^{ex}(\alpha)$, then $\epsilon_{\tilde{\sigma},\tilde{\mu}}(C_{\tilde{\alpha}}) = \epsilon_{\sigma,\mu}(C_\alpha)$ and $\epsilon_{\tilde{\sigma},\tilde{\mu}}(\tilde{\alpha}) = \epsilon_{\sigma,\mu}(\alpha)$.

Proof. We just re-apply the theorem 7.9, since (f, f^{tr}, f^{ex}) is a PSIOA executions-matching from $(psioa(X), \mu)$ to $(psioa(Y), \mu')$. □

Theorem 7.31 (Continued PCA executions-matching preserves general probabilistic distribution). *Let X and Y be two PCA $(\mu, \mu') \in Disc(Frags(X)) \times Disc(Frags(Y))$. Let (f, f^{tr}, f^{ex}) be a PCA executions-matching from (X, μ) to (Y, μ') . Let $((f, f^+), f^{tr,+}, f^{ex,+})$ be a continuation of (f, f^{tr}, f^{ex}). Let $(\tilde{\sigma}, \sigma) \in schedulers(\mathcal{A}) \times schedulers(\mathcal{B})$, s. t. $(\tilde{\sigma}, \sigma)$ are (f, f^{tr}, f^{ex})*-alter egos. *Let $(\alpha, \pi) \in dom(f^{ex,+}) \times Frags(Y)$.*
If $\pi = f^{ex,+}(\alpha)$, then $\epsilon_{\tilde{\sigma},\tilde{\mu}}(C_{\tilde{\alpha}}) = \epsilon_{\sigma,\mu}(C_\alpha)$.

Proof. We just re-apply the theorem, 7.10 since $((f, f^+), f^{tr,+}, f^{ex,+})$ is a continued PSIOA executions-matching from $(psioa(X), \mu)$ to $(psioa(Y), \mu')$. □

Composability of execution-matching relationship. Now we are looking for composability of PCA executions-matching. Here again the notions are the same than the ones for PSIOA excepting for states-matching and for partial-compatibility. Hence we only need to show that i) the $\mathcal{E}$-extension of a PCA states-matching is still a PCA states-matching (see lemma 7.32), ii) if $f : states(X) \rightarrow states(Y)$ is a bijective PCA states-matching and f^{-1} is complete, then for every PCA $\mathcal{E}$ partial-compatible with X, $\mathcal{E}$ is partial-compatible Y (see lemma 7.34).

Lemma 7.32 (Composability of PCA states-matching). *Let X and Y be two PCA with $Q_X = states(X)$ and $Q_Y = states(Y)$ as respective sets of states. Let $\mathcal{E}$ be partially-compatible with both X and Y. Let $f : Q'_X \subset Q_X \rightarrow Q_Y$ be a PCA states-matching. Let g be the $\mathcal{E}$-extension of f.*

If $range(g) \subset states(Y||\mathcal{E})$, then g is a PCA states-matching from $X||\mathcal{E}$ to $Y||\mathcal{E}$.

Proof.
- If $(\bar{q}_X, \bar{q}_\mathcal{E}) \in Q_{X||\mathcal{E}}$ then $\bar{q}_X \in Q'_X$ which means $f(\bar{q}_X) = \bar{q}_Y$, thus $g((\bar{q}_X, \bar{q}_\mathcal{E})) = (\bar{q}_\mathcal{E}, \bar{q}_\mathcal{E})$.
- $\forall ((q_X, q_\mathcal{E}), (q_Y, q_\mathcal{E})) \in Q'_{X||\mathcal{E}} \times states(Y||\mathcal{E})$ with $(q_Y, q_\mathcal{E}) = g((q_X, q_\mathcal{E}))$, we have
  - Configuration preservation (modulo hiding): if $auts(config(X)(q_X)) = (\mathcal{A}_1, ..., \mathcal{A}_n)$, then $auts(config(Y)(q_Y)) = (\mathcal{A}'_1, ..., \mathcal{A}'_n)$ where $\forall i \in [1 : n], \mathcal{A}_i = hide(\mathcal{A}'_i, h_i)$ with h_i defined on $states(\mathcal{A}'_i)$, s. t. $h_i(q_{\mathcal{A}'_i}) \subseteq out(\mathcal{A}'_i)(q_{\mathcal{A}'_i})$ (resp. s. t. $h_i(q_{\mathcal{A}'_i}) = \emptyset$, that is $\mathcal{A}_i = \mathcal{A}'_i$). Hence if $auts(config(X||\mathcal{E})((q_X, q_\mathcal{E})) = (\mathcal{A}_1, ..., \mathcal{A}_n, \mathcal{B}_1, ..., \mathcal{B}_m)$, then $auts(config(Y||\mathcal{E})((q_Y, q_\mathcal{E})) = (\mathcal{A}'_1, ..., \mathcal{A}'_n, \mathcal{B}_1, ..., \mathcal{B}_m)$ where $\forall i \in [1 : n], \mathcal{A}_i = hide(\mathcal{A}'_i, h_i)$ with h_i defined on $states(\mathcal{A}'_i)$, s. t. $h_i(q_{\mathcal{A}'_i}) \subseteq out(\mathcal{A}'_i)(q_{\mathcal{A}'_i})$ (resp. s. t. $h_i(q_{\mathcal{A}'_i}) = \emptyset$, that is $\mathcal{A}_i = \mathcal{A}'_i$).

- Hidding preservation (modulo hiding): $hidden\text{-}actions(X)(q_X) = hidden\text{-}actions(Y)(q_Y) \cup h^+(q_Y)$ where h^+ defined on $states(Y)$, s. t. $h^+(q_Y) \subseteq out(Y)(q_Y)$. Hence $hidden\text{-}actions(X||\mathcal{E})((q_X, q_\mathcal{E})) = hidden\text{-}actions(X)(q_X) \cup hidden\text{-}actions(\mathcal{E})(q_\mathcal{E}) = hidden\text{-}actions(Y)(q_Y) \cup hidden\text{-}actions(\mathcal{E})(q_\mathcal{E}) \cup h^+(q_Y) = hidden\text{-}actions(Y||\mathcal{E})((q_Y, q_\mathcal{E})) \cup h^{+\prime}((q_Y, q_\mathcal{E}))$ where $h^{+\prime}$ defined on $states(Y||\mathcal{E})$, s. t. $h^{+\prime}((q_Y, q_\mathcal{E})) = h^+(q_Y) \subseteq out(Y)(q_Y) \subseteq out(Y||\mathcal{E})((q_Y, q_\mathcal{E}))$.
- Creation preservation $\forall a \in \widehat{sig}(X)(q_X) = \widehat{sig}(Y)(q_Y)$, $created(X)(q_X)(a) = created(Y)(q_Y)(a)$. Hence $\forall a \in \widehat{sig}(X||\mathcal{E})((q_X, q_\mathcal{E})) = \widehat{sig}(Y||\mathcal{E})((q_Y, q_\mathcal{E}))$, either
  - $a \in \widehat{sig}(X)(q_X) = \widehat{sig}(Y)(q_Y)$ but $a \notin \widehat{sig}(\mathcal{E})(q_\mathcal{E})$ and then $created(X||\mathcal{E})((q_X, q_\mathcal{E}))(a) = created(X)(q_X)(a) = created(Y)(q_Y) = created(Y||\mathcal{E})((q_Y, q_\mathcal{E}))(a)$
  - or $a \notin \widehat{sig}(X)(q_X) = \widehat{sig}(Y)(q_Y)$ but $a \in \widehat{sig}(\mathcal{E})(q_\mathcal{E})$ and then $created(X||\mathcal{E})((q_X, q_\mathcal{E}))(a) = created(\mathcal{E})(q_\mathcal{E})(a) = created(Y||\mathcal{E})((q_Y, q_\mathcal{E}))(a)$
  - or $a \in \widehat{sig}(X)(q_X) = \widehat{sig}(Y)(q_Y)$ and $a \in \widehat{sig}(\mathcal{E})(q_\mathcal{E})$ and then $created(X||\mathcal{E})((q_X, q_\mathcal{E}))(a) = created(X)(q_X)(a) \cup created(\mathcal{E})(q_\mathcal{E})(a) = created(Y)(q_Y) \cup created(\mathcal{E})(q_\mathcal{E})(a) = created(Y||\mathcal{E})((q_Y, q_\mathcal{E}))(a)$

  Thus, $\forall a \in \widehat{sig}(X||\mathcal{E})((q_X, q_\mathcal{E})) = \widehat{sig}(Y||\mathcal{E})((q_Y, q_\mathcal{E}))$, $created(X||\mathcal{E})((q_X, q_\mathcal{E}))(a) = created(Y||\mathcal{E})((q_Y, q_\mathcal{E}))(a)$.

□

We restate the theorem 7.20 of executions-matching composability.

Theorem 7.33 (Composability of PCA matching-execution). *Let X and Y be two PCA. Let $\mathcal{E}$ be partially-compatible with both X and Y. Let (f, f^{tr}, f^{ex}) be a PCA executions-matching from X to Y. Let g be the $\mathcal{E}$-extension of f. If $range(g) \subset states(Y||\mathcal{E})$, then the $\mathcal{E}$-extension of (f, f^{tr}, f^{ex}) is a PCA executions-matching (g, g^{tr}, g^{ex}) from $X||\mathcal{E}$ to $Y||\mathcal{E}$ induced by g and $dom(g^{tr})$.*

Proof. This comes immediately from theorem 7.20. □

We extend the lemma 7.21 but we have to take a little precaution for the partial-compatibility since here the configurations have to be pairwise compatible, not only the signatures.

Lemma 7.34 (Some properties preserved by $\mathcal{E}$-extension of a PCA executions-matching). *Let X and Y be two PCA. Let (f, f^{tr}, f^{ex}) be a PCA executions-matching from X to Y.*

1. *If f is complete, then for every PSIOA $\mathcal{E}$ partially-compatible with X, $\mathcal{E}$ is partially-compatible with Y.*
2. *Let $\mathcal{E}$ partially-compatible with both X and Y, let g be the $\mathcal{E}$-extension of f.*
   (a) *If f is bijective and f^{-1} is complete, then $range(g) = states(Y||\mathcal{E})$ and so we can talk about the $\mathcal{E}$-extension of (f, f^{tr}, f^{ex})*
   (b) *If (f, f^{tr}) is a bijective complete transition-matching, (g, g^{tr}) is a bijective complete transition-matching. (And (f, f^{tr}, f^{ex}) and (g, g^{tr}, g^{ex}) are bijective complete execution-matching.)*
   (c) *If f is strong, then g is strong*

Proof. (1) We need to show that every pseudo-execution of $(Y, \mathcal{E})$ ends on a compatible state. Let $\pi = q^0 a^1 q^1 ... a^n q^n$ be a finite pseudo-execution of $(Y, \mathcal{E})$. We note $\alpha = (f^{-1}(q_Y^0), q_\mathcal{E}^0) a^1 (f^{-1}(q_Y^1), q_\mathcal{E}^1) ... a^n (f^{-1}(q_Y^n), q_\mathcal{E}^n)$. The proof is in two steps. First, we show by induction that $\alpha = (f^{-1}(q_Y^0), q_\mathcal{E}^0) a^1 (f^{-1}(q_Y^1), q_\mathcal{E}^1) ... a^n (f^{-1}(q_Y^n), q_\mathcal{E}^n)$ is an execution of $X||\mathcal{E}$. Second, we deduce that it means $(f^{-1}(q_Y^n), q_\mathcal{E}^n)$ is a compatible state of $(X, \mathcal{E})$ which means that a fortiori, $(q_Y^n, q_\mathcal{E}^n)$ is a compatible state of $(Y, \mathcal{E})$ which ends the proof.

- First, we show by induction that α is an execution of $X||\mathcal{E}$. We have $(f^{-1}(\bar{q}_Y), \bar{q}_{\mathcal{E}}) = (\bar{q}_X, \bar{q}_{\mathcal{E}})$ which ends the basis.
  Let assume $(f^{-1}(q_Y^0), q_{\mathcal{E}}^0)a^1(f^{-1}(q_Y^1), q_{\mathcal{E}}^1)...a^k(f^{-1}(q_Y^k), q_{\mathcal{E}}^k)$ is an execution of $X||\mathcal{E}$. Hence $(f^{-1}(q_Y^k), q_{\mathcal{E}}^k)$ is a compatible state of $(X, \mathcal{E})$ which means that a fortiori q^k is a compatible state of $(Y, \mathcal{E})$ because of signature preservation of f.
  For the same reason, $\widehat{sig}(Y, \mathcal{E})(q^k) = \widehat{sig}(X||\mathcal{E})((f^{-1}(q_Y^k), q_{\mathcal{E}}^k))$, so $a^{k+1} \in \widehat{sig}(X, \mathcal{E})((f^{-1}(q_Y^k), q_{\mathcal{E}}^k))$. Then we use the completeness of $(f^{-1}, (f^{tr})^{-1})$, to obtain the fact that either $\eta_{(Y,q_Y^k,a^{k+1})} \in dom((f^{tr})^{-1})$ or $a^{k+1} \notin \widehat{sig(Y)}(q_Y^k)$ (and we recall the convention that in this second case $\eta_{(Y,q_Y^k,a^{k+1})} = \delta_{q_Y^k}$). which means either $(f^{-1}(q_Y^k), a^{k+1}, \eta_{(X,f^{-1}(q_Y^k),a^{k+1})})$ is a transition of X that ensures $\forall q'' \in supp(\eta_{(Y,q_Y^k,a^{k+1})}), f^{-1}(q'') \in supp(\eta_{(X,f^{-1}(q_Y^k),a^{k+1})})$ or $a^{k+1} \notin \widehat{sig(X)}(f^{-1}(q_Y^k))$ (and we recall the convention that in this second case $\eta_{(X,f^{-1}(q_Y^k),a^{k+1})} = \delta_{f^{-1}(q_Y^k)}$). Thus for every $(q'', q''') \in supp(\eta_{(Y,\mathcal{E}),q^k,a^{k+1})})$, $(f^{-1}(q''), q''') = g^{-1}((q'', q''')) \in supp(\eta_{(X,\mathcal{E}),g^{-1}(q^k),a^{k+1})})$ namely for $(q'', q''') = (q_Y^{k+1}, q_{\mathcal{E}}^{k+1})$. Hence, $(f^{-1}(q_Y^{k+1}), q_{\mathcal{E}}^{k+1})$ is reachable by $(X, \mathcal{E})$ which means $(f^{-1}(q_Y^0), q_{\mathcal{E}}^0)a^1(f^{-1}(q_Y^1), q_{\mathcal{E}}^1)...a^k(f^{-1}(q_Y^k), q_{\mathcal{E}}^k)a^k(f^{-1}(q_Y^k), q_{\mathcal{E}}^k)a^{k+1}(f^{-1}(q_Y^{k+1}), q_{\mathcal{E}}^{k+1})$ is an execution of $X||\mathcal{E}$. Thus by induction α is an execution of $X||\mathcal{E}$.
- Since X and $\mathcal{E}$ are partially-compatible $(f^{-1}(q_Y^n), q_{\mathcal{E}}^n)$ is a state of $X||\mathcal{E}$, so $(f^{-1}(q_Y^n), q_{\mathcal{E}}^n)$ is a compatible state of $(X, \mathcal{E})$ which means $(q_Y^k, q_{\mathcal{E}}^k)$ is a fortiori a compatible state of $(Y, \mathcal{E})$. Hence every reachable state of $(Y, \mathcal{E})$ is compatible which means Y and $\mathcal{E}$ are partially compatible which ends the proof.

(2) This comes immediately from lemma 7.21 since (f, f^{tr}, f^{ex}) is a PSIOA executions-matching from $psioa(X)$ to $psioa(Y)$ by construction.

□

Finally, we restate the semantic-equivalence.

A strong complete bijective transitions-matching implies a strong complete bijective executions-matching which means the two automata are completely semantically equivalent.

Definition 7.35 (PCA semantic equivalence). Let X an Y be two PCA. We say that X and Y are semantically-equivalent if it exists a complete bijective strong PCA executions-matching from X to Y

Theorem 7.36 (composability of semantic equivalence). *Let X and Y be PCA semantically-equivalent. Then for every PSIOA $\mathcal{E}$:*

- *$\mathcal{E}$ is partially-compatible with $X \iff \mathcal{E}$ is partially-compatible with Y*
- *if $\mathcal{E}$ is an environment for both X and Y, then $X||\mathcal{E}$ and $Y||\mathcal{E}$ are PCA semantically-equivalent.*

Proof.
- The first item comes from lemma 7.34, first item
- The second item comes from lemma 7.34, second item

□

A weak complete bijective PCA transitions-matching implies a weak complete bijective PCA executions-matching which means the two automata are completely semantically equivalent modulo some hiding operation that implies that some PSIOA are partially-compatible with one of the automaton and not with the other one and that the traces are not necessarily the same ones.

8 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 $Y_{\mathcal{A}} = X_{\mathcal{A}} \setminus \{\mathcal{A}\}$. The reader can already take a look on the figures 26 and 27 to get an intuition on the desired result. 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.

We first define some notions of projection on configurations on subsection 8.1. Then we define the notion of $\mathcal{A}$-fair PCA X in subsection 8.2, which will be a sufficient condition to ensure that $Y = X \setminus \{\mathcal{A}\}$ is still a PCA, namely that it ensures the constraints of top/down and bottom/up transition preservation, which is proved in the last subsection 8.3.

8.1 Projection on Configurations

In this subsection, we want to define formally $\eta' \in Disc(Q_{conf})$ that would be the result of $\eta \in Disc(Q_{conf})$ "deprived of an automaton $\mathcal{A}$". This is achieved in definition 8.4. This definition requires particular precautions and motivate the next sequence of definitions, from definition 8.1 to 8.3

The next definition captures the idea of a state deprived of a PSIAO $\mathcal{A}$.

Definition 8.1 (State projection). Let $\mathbf{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ be a set of PSIOA partially-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).
- $q \upharpoonright \mathcal{A}_k = q_k$ if $\mathcal{A}_k \in \mathbf{A}$ only.
- $q \upharpoonright \mathbf{A}^s = q \setminus (\mathbf{A} \setminus \mathbf{A}^s)$ (recursive extension of the previous item).

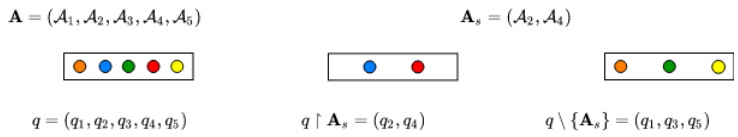


Fig. 19. State projection

The next definition captures the idea of a family transition deprived of a PSIAO $\mathcal{A}$.

Definition 8.2 (Family transition projection). (see figure 20 first for an intuition) Let $\mathbf{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ be a set of PSIOA partially-compatible at state $q = (q_1, ..., q_n) \in Q_{\mathcal{A}_1} \times ... \times Q_{\mathcal{A}_n}$. Let $\mathbf{A}^s = \{\mathcal{A}_{s^1}, ..., \mathcal{A}_{s^n}\} \subset \mathbf{A}$.

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

Then we apply this notation to preserving distributions.

Definition 8.3 (preserving distribution projection). (see figure 21) Let $\eta_p \in Disc(Q_{conf})$ be a preserving distribution. Let $\mathbf{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ its automata support (that is $\forall (\mathbf{A}', \mathbf{S}') \in supp(\eta_p), \mathbf{A}' = \mathbf{A}$). Let H be its set of companion

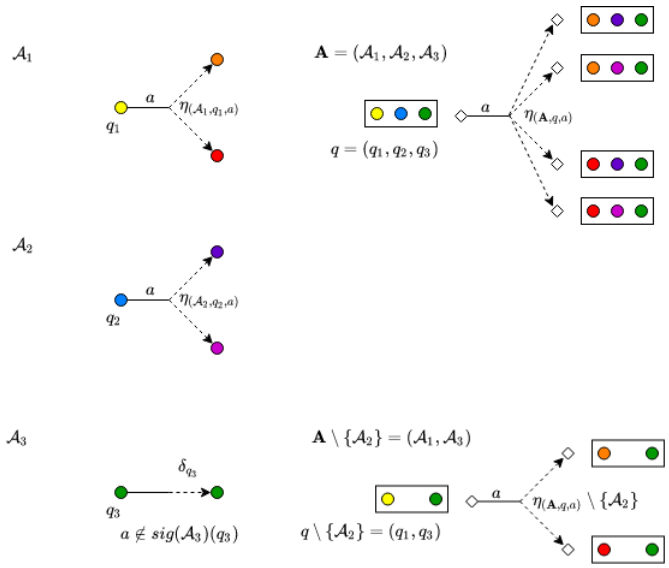


Fig. 20. Family transition projection

distributions of η_p (s. 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 automata 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 automata support and $H'' = \{\eta \upharpoonright \mathbf{A}^s | \eta \in H\}$ as companion distribution set.

Now we are able to define intrinsic transition deprived of a PSIOA $\mathcal{A}$.

Definition 8.4 (intrinsic transition projection). (see figure 22) 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. The next lemma 8.5 and 8.6 will lead to lemma 8.7. All of this 3 lemma are some versions of law of total probability. The lemma 8.7 and 8.10 (obtained via lemma 8.8), will allow the constructive definition 8.11 of PCA deprived of a (sub) PSIOA.

Lemma 8.5 (family distribution projection). *(see figure 23) Let $\mathbf{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ be a finite set of automata. Let $\mathbf{A}' = \mathbf{A} \setminus \{\mathcal{A}_k\}$. Let $\eta = \eta_1 \otimes ... \otimes \eta_n$ with $\eta_i \in Disc(Q_{\mathcal{A}_i})$ for every $i \in [1, n]$. Let $\eta' = \eta \setminus \{\mathcal{A}_k\}$.*
For every $q' \in Q_{\mathbf{A}'}$, $\eta'(q') = \Sigma_{(q \in Q_{\mathbf{A}}, q \setminus \{\mathcal{A}_k\} = q')} \eta(q)$

Proof. This comes directly from the law of total probability. We have $\forall q^\ell = (q_1, ..., q_k^\ell, ..., q_n) \in Q_{\mathbf{A}}, q' = (q_1, ..., q_{k-1}, q_{k+1}, ..., q_n) \in Q_{\mathbf{A}'}, \eta(q^\ell) = \eta'(q') \cdot \eta_k(q_k^\ell)$. Hence $\sum_{q_k^\ell \in supp(\eta_k)} \eta(q^\ell) = \sum_{q_k^\ell \in supp(\eta_k)} \eta'(q') \cdot \eta_k(q_k^\ell)$, which gives $\sum_{q \in supp(\eta), q' = q \setminus \{\mathcal{A}_k\}} \eta(q) = \eta'(q') \cdot \sum_{q_k^\ell \in supp(\eta_k)} \eta_k(q_k^\ell)$ and finally $\sum_{q \in Q_{\mathbf{A}}, q' = q \setminus \{\mathcal{A}_k\}} \eta(q) = \eta'(q')$.

□

Lemma 8.6 (preserving distribution projection). *(see figure 24) Let η_p be a preserving distribution with $\mathbf{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ as automata support . Let C_Y be a configuration $(\eta_p \setminus \{\mathcal{A}_k\})(C_Y) = \Sigma_{(C_X, C_X \setminus \{\mathcal{A}_k\} = C_Y)} \eta_p(C_X)$.*

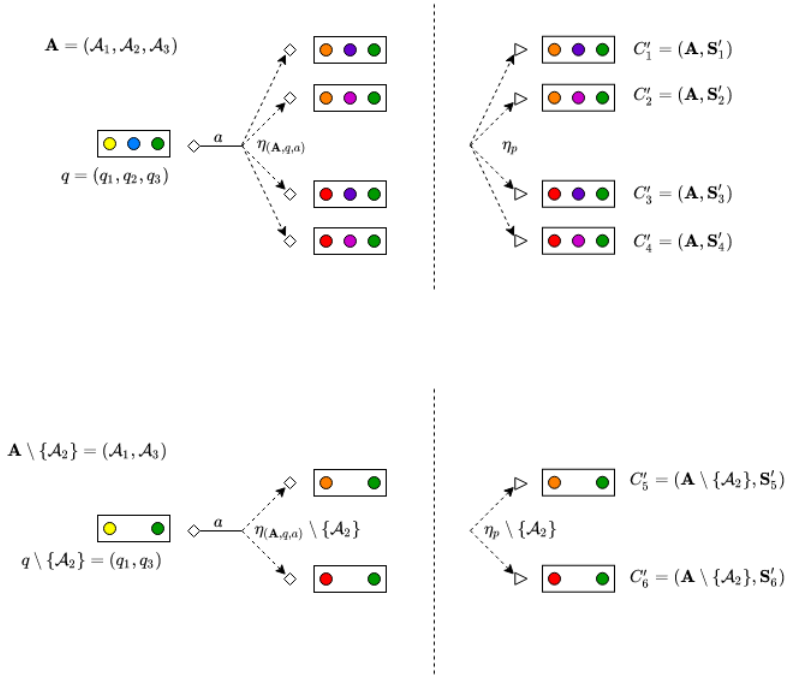


Fig. 21. Preserving distribution projection

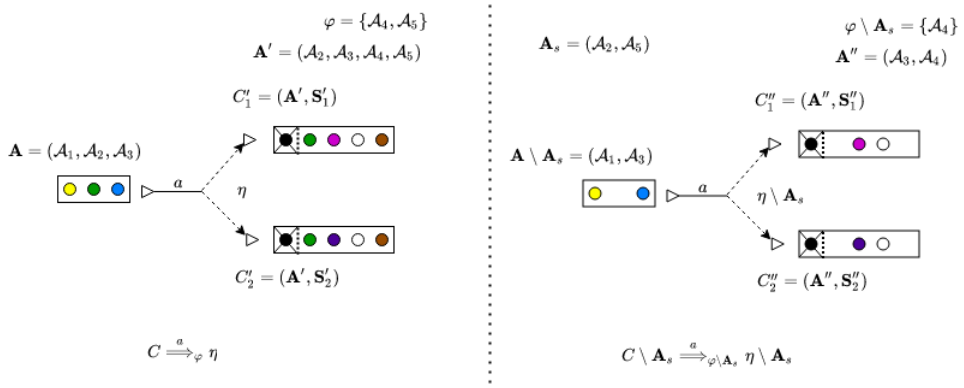


Fig. 22. intrinsic transition projection

Proof. We can apply lemma 8.5 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 8.5 every state q by the corresponding configuration. □

Lemma 8.7 (reduced distribution projection). *Let η_p be a preserving distribution with $\mathbf{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ as automata support . Let η_r be generated by φ and η_p. Let C_Y be a configuration.*

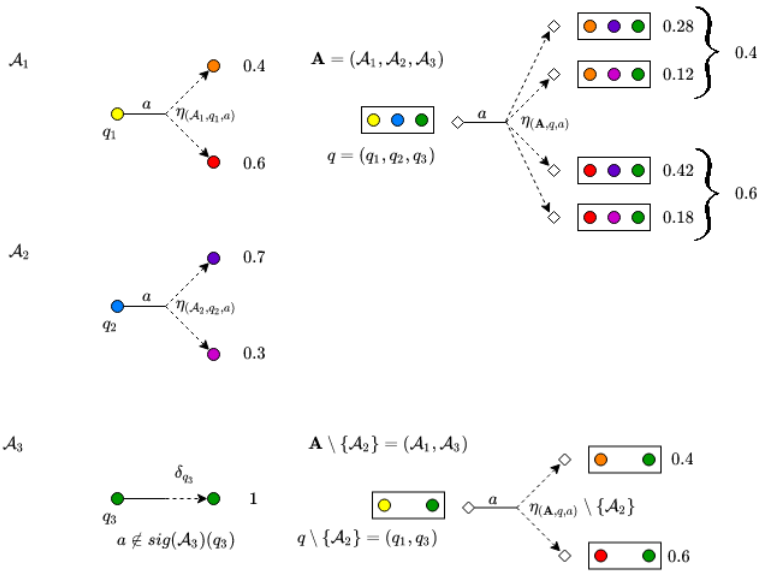


Fig. 23. total probability law for family transition projection

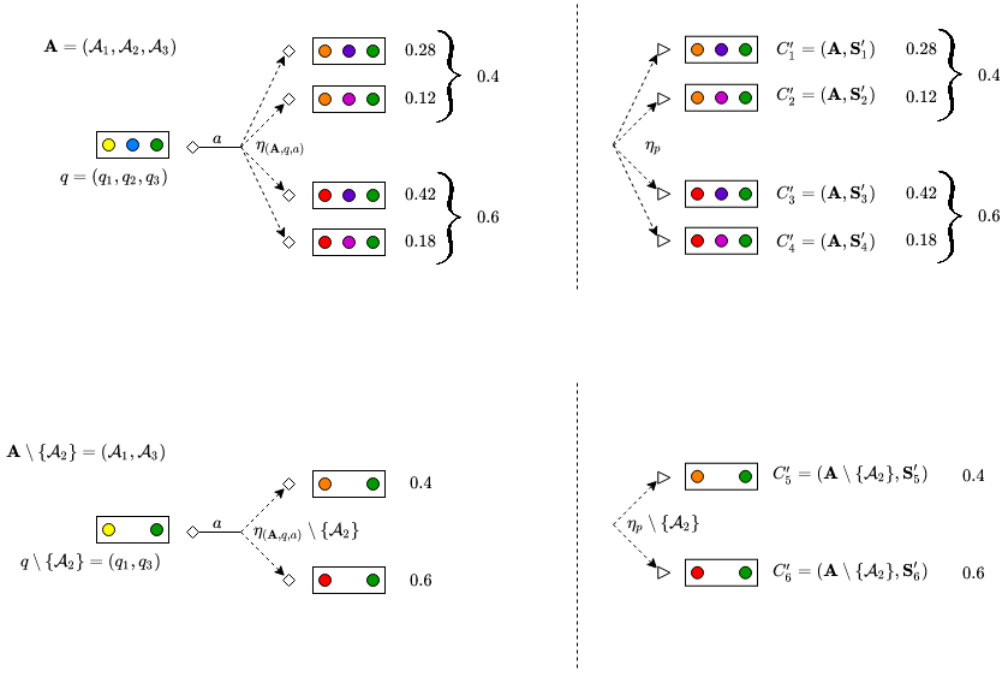


Fig. 24. total probability law for preserving configuration distribution and its companion distribution

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

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 every configuration C_Y from lemma 8.6. By definition 5.10, 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. □

The next lemma gives the intrinsic transition attached to a configuration after deprivation of a (sub) PSIOA.

Lemma 8.8 (projection on an intrinsic transition). *Let C be a configuration, P an automaton, $a \in \widehat{sig}(C \setminus P)$, $\varphi \subset Autids$ and $\eta \in Disc(Q_{conf})$, s. t.*
$C \overset{a}{\Longrightarrow}_{\varphi} \eta_r$. Then , $C \setminus \{P\} \overset{a}{\Longrightarrow}_{(\varphi \setminus \{P\})} (\eta_r \setminus \{P\})$.

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 4.10. 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, $C \setminus \{P\} \overset{a}{\rightarrow} (\eta_p \setminus \{P\})$ and then $C \setminus \{P\} \overset{a}{\Longrightarrow}_{(\varphi \setminus \{P\})} (\eta_r \setminus \{P\})$.

□

In next subsection, this lemma 8.8 will lead to lemma 8.10 which will be a key lemma to allow the constructive definition 8.11 of PCA deprived of a (sub) PSIOA.

8.2 $\mathcal{A}$-fairness assumption, motivated by our definition of PCA deprived from an internal PSIOA: $X \setminus \{\mathcal{A}\}$

Here we recall in definition 8.9 the definition 6.13 of a $\mathcal{A}$-fair PCA. Then we show lemma 8.10 (via 8.8) that will be used in complement of lemma 8.7 to enable the constructive definition of $X \setminus \{\mathcal{A}\}$.

Definition 8.9 ($\mathcal{A}$-fair PCA (recall)). Let $\mathcal{A} \in Autids$. Let X be a PCA. We say that X is $\mathcal{A}$-fair if it verifies the following constraints.

- (configuration-conflict-free) X is configuration-conflict-free, that is $\forall q_X, q'_X \in states(X)$, s. t. $q_X R_{conf} q'_X$ (i. e. $config(X)(q_X) = config(X)(q'_X)$) then $q_X = q'_X$
- (no conflict for projection) $\forall q_X, q'_X \in states(X)$, s. t. $q_X R_{conf}^{\setminus \{\mathcal{A}\}} q'_X$ then $q_X R_{strict}^{\{\mathcal{A}\}} q'_X$. That is if $config(X)(q_X) \setminus \{A\} = config(Y)(q_Y) \setminus \{A\}$, then
  - $\forall a \in \widehat{sig}(X)(q_X) \cap \widehat{sig}(Y)(q_Y)$, $created(Y)(q_Y)(a) \setminus \{\mathcal{A}\} = created(X)(q_X)(a) \setminus \{\mathcal{A}\}$
  - $hidden\text{-}actions(X)(q_X) \setminus pot\text{-}out(X)(q_X)(\mathcal{A}) = hidden\text{-}actions(Y)(q_Y) \setminus pot\text{-}out(Y)(q_Y)(\mathcal{A})$ where
    - $pot\text{-}out(X)(q)(\mathcal{A}) = \emptyset$ if $\mathcal{A} \notin auts(config(X)(q))$
    - $pot\text{-}out(X)(q)(\mathcal{A}) = out(\mathcal{A})(map(config(X)(q))(\mathcal{A}))$ if $\mathcal{A} \in auts(config(X)(q))$
- (no exclusive creation by $\mathcal{A}$) $\forall q_X \in states(X)$, $\forall a \in \widehat{sig}(X)(q_X)$ $\mathcal{A}$-exclusive in q_X, $created(X)(q_X)(a) = \emptyset$ where $\mathcal{A}$-exclusive means $\forall \mathcal{B} \in auts(config(X)(q_X))$, $a \notin \widehat{sig}(\mathcal{B})(map(config(X)(q_X))(\mathcal{B}))$.

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 will allow the definition of $X \setminus \{\mathcal{A}\}$, where X is a PCA, to be well-defined.

Now we give the second key lemma (after lemma 8.7) to allow the definition 8.11 of PCA deprived of a (sub) PSIOA. Basically, this lemma that if two states q_X and q_Y are strictly equivalent modulo the deprivation of a (sub) automaton P, noted $q_X R_{strict}^{\setminus\{P\}} q_Y$, then the intrinsic configurations issued from these states deprived of P are equal.

Lemma 8.10 (equality of intrinsic transition after deprivation of a sub-PSIOA). *Let X, Y be two PCA. Let $(q_X, q_Y) \in states(X) \times states(Y)$ s. t. $q_X R_{strict}^{\setminus\{P\}} q_Y$. Let $a \in \widehat{sig}(X)(q_X) \cap \widehat{sig}(Y)(q_Y) \setminus (pot\text{-}out(X)(q_X)(P) \cup pot\text{-}out(Y)(q_Y)(P))$. We note $C_X \triangleq config(X)(q_X), C_Y \triangleq config(Y)(q_Y), \varphi_X \triangleq created(X)(q_X)(a), \varphi_Y \triangleq created(Y)(q_Y)(a)$ and η_r^X and η_r^Y the unique reduced configuration distribution s. t. $C_X \overset{a}{\Longrightarrow}_{\varphi_X} \eta_r^X$ and $C_Y \overset{a}{\Longrightarrow}_{\varphi_Y} \eta_r^Y$. By definition of $R_{strict}^{\setminus\{P\}}$, we have $C_X \setminus \{P\} = C_Y \setminus \{P\} \triangleq C$ and $\varphi_X \setminus \{P\} = \varphi_Y \setminus \{P\} = \varphi$.*

Moreover $C \overset{a}{\Longrightarrow}_{\varphi} \eta_r$ with $\eta_r^X \setminus \{P\} = \eta_r^Y \setminus \{P\} \triangleq \eta_r$

Proof. By lemma 8.8, we have both $C \overset{a}{\Longrightarrow}_{\varphi} \eta_r^X \setminus \{P\}$ and $C \overset{a}{\Longrightarrow}_{\varphi} \eta_r^Y \setminus \{P\}$. By unicity of intrinsic transition, we have $\eta_r^X \setminus \{P\} = \eta_r^Y \setminus \{P\}$. □

Definition 8.11 ($X \setminus \{P\}$). (see figure 25 for the constructive definition and figures 26 and 27 for the desired result.) Let $P \in Autids$. Let X be a P-fair PCA, with $psioa(X) = (Q_X, \bar{q_X}, sig(X), D_X)$. We note $X \setminus \{P\}$ the automaton Y equipped with the same attributes than a PCA (psioa, config, hidden-actions, created), $\mu_s^P : Q_X \to Q_Y$ and $\mu_d^P : D_X \setminus \{\eta_{(X,q_X,a)} | a \text{ is } P\text{-exclusive in } q_X\} \to D_Y$ that respect systematically the following rules:

- P-deprivation: $\forall q_Y \in states(Y), P \notin config(Y)(q_Y), \forall a \in \widehat{sig}(Y)(q_Y)(a), P \notin created(Y)(q_Y)(a)$.
- μ_s^P-correspondence: $\forall (q_X, q_Y) \in Q_X \times Q_Y$ s. t. $\mu_s^P(q_X) = q_Y$, then $q_X R_{strict}^{\setminus\{P\}} q_Y$.
- μ_d^P-correspondence: $\forall (q_X, q_Y) \in Q_X \times Q_Y, \forall (a_X, a_Y) \in \widehat{sig}(X)(q_X) \times \widehat{sig}(Y)(q_Y)$ s. t. $\eta_{(Y,q_Y,a_Y)} = \mu_d^P(\eta_{(X,q_X,a_X)}$, then (1) $\mu_s^P(q_X) = q_Y$, (2) $a_X = a_Y$ and (3) $\forall q'_Y \in Q_Y, \eta_{(Y,q_Y,a)}(q'_Y) = \Sigma_{q'_X \in Q_X, \mu_s(q'_X) = q'_Y} \eta_{(X,q_X,a)}(q'_X)$.

and constructed (conjointly with the mapping μ_s^P and μ_d^P) as follows:

- Partitioning: We partition Q_X in equivalence classes according to the equivalence relation $R_{conf}^{\setminus\{P\}}$ that is we obtain a partition $(C_j)_{j \in J \subset \mathbb{N}}$ s. t. $\forall j \in J, \forall q_X, q'_X \in C_j, q_X R_{conf}^{\setminus\{P\}} q'_X$ and by P-fair assumption, $q_X R_{strict}^{\setminus\{P\}} q'_X$
- Q_Y, $sig(Y)$ and μ_s^P: $\forall j \in J$, we construct $q_Y^j \in Q_Y$ and conjointly extend μ_s^P s. t. $\forall q_X \in C_j$, $\mu_s^P(q_X) = q_Y^j$, verifying the P-deprivation-rule and μ_s^P-correspondence rule, that is
  - $config(Y)(q_Y^j) = config(X)(q_X) \setminus \{P\}$,
  - $hidden\text{-}actions(Y)(q_Y^j) = hidden\text{-}actions(X)(q_X) \setminus pot\text{-}out(X)(q_X)(P)$,
  - $sig(Y)(q_Y^j) = hide(sig(config(Y)(q_Y^j)), hidden\text{-}actions(Y)(q_Y^j))$
  - $\forall a \in \widehat{sig}(Y)(q_Y^j), created(Y)(q_Y^j)(a) = created(X)(q_X)(a) \setminus \{P\}$.
  - Furthermore $\bar{q}_Y = \mu_s^P(\bar{q}_X)$.
- D_Y and μ_d^P: $\forall q_Y \in Q_Y, \forall a \in \widehat{sig}(Y)(q_Y)$ (and so $\forall q_X \in (\mu_s^P)^{-1}(q_Y), a \in \widehat{sig}(X)(q_X)$) we construct $\eta_{(Y,q_Y,a)}$ and conjointly extend μ_d^P s. t. $\forall q_X \in (\mu_s^P)^{-1}(q_Y), \eta_{(Y,q_Y,a)} = \mu_d^P(\eta_{(X,q_X,a)})$, verifying the μ_d^P-correspondence rule. We show this construction is possible:
  - We note $C_Y = config(Y)(q_Y), \varphi_Y = created(Y)(q_Y)(a), \eta_Y$ the unique reduced configuration distribution so that $C_Y \overset{a}{\Longrightarrow}_{\varphi_Y} \eta_Y$. Let $(q_X^i)_{i \in I \subset \mathbb{N}} = (\mu_s^P)^{-1}(q_Y)$. For every $i \in I$, we note note $C_X^i = config(X)(q_X^i)$, $\varphi_X^i = created(X)(q_X^i)(a)$, η_X^i the unique reduced configuration distribution so that $C_X^i \overset{a}{\Longrightarrow}_{\varphi_X^i} \eta_X^i$. For every $i \in I$, we have $C_X^i \setminus \{P\} = C_Y$, $\varphi^i \setminus \{P\} = \varphi_Y$ and $\eta_X^i \setminus \{P\} = \eta_Y$ by lemma 8.10.
  - For every $q_X^i \in (\mu_s^P)^{-1}(q_Y)$, we partition $supp(\eta_{(X,q_X^i,a)})$ in equivalence classes according to the equivalence relation $R_{conf}^{\setminus\{P\}}$ that is we obtain a partition $(C'_j)_{j \in J' \subset \mathbb{N}}$ s. t. $\forall j \in J', \forall q'_X, q''_X \in C'_j, q'_X R_{conf}^{\setminus\{P\}} q''_X$ and

by P-fair assumption, $q'_X R^{\setminus\{P\}}_{strict} q''_X$. For each $j \in J'$, we extract an arbitrary $q'_X \in C'_j$ and $q'_Y = \mu_s^P(q'_X)$. We fix $\eta_{(Y,q_Y,a)}(q'_Y) = \eta_Y(config(Y)(q'_Y))$.

Now by lemma 8.7, $\eta_Y(C'_Y) = \sum_{C'_X, C'_Y = C'_X \setminus \{P\}} \eta^i_X(C'_X)$. By constraint 3 of bottom/up transition preservation, $\sum_{C'_X, C'_Y = C'_X \setminus \{P\}} \eta^i_X(C'_X) = \sum_{q'_X, C'_Y = config(X)(q'_X) \setminus \{P\}} \eta_{(X,q^i_X,a)}(q'_X)$. By construction of Q_Y under μ_s^P-correspondence, $\sum_{q'_X, C'_Y = config(X)(q'_X)\setminus\{P\}} \eta_{(X,q^i_X,a)}(q'_X) = \sum_{q'_X, q'_Y = \mu_s^P(q'_X)} \eta_{(X,q^i_X,a)}(q'_X)$. Thus, the μ_d^P-correspondence constraint, i. e. $\eta_{(Y,q_Y,a)}(q'_Y) = \sum_{q'_X, q'_Y = \mu_s^P(q'_X)} \eta_{(X,q^i_X,a)}(q'_X)$ holds for all the possible $q^i_X \in (\mu_s^P)^{-1}(q_Y)$.

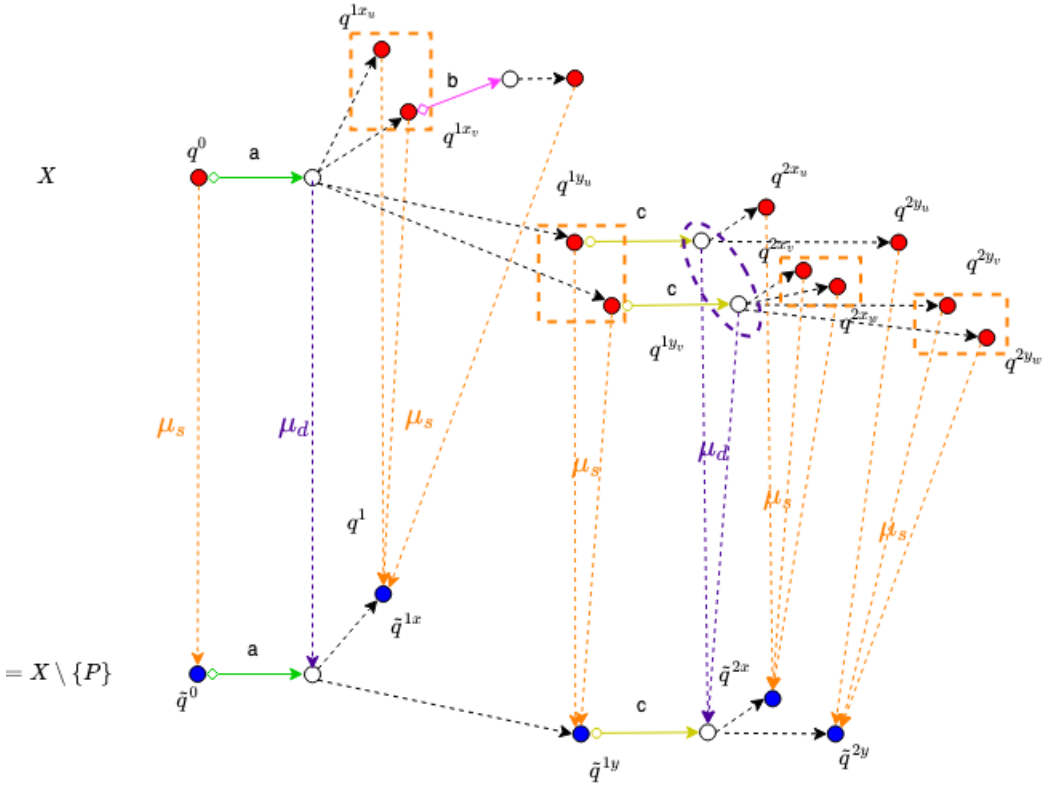


Fig. 25. constructive definition of $Y = X \setminus \{P\}$. First we construct $\tilde{q}^0$ which is the initial state of Y. Then we partition $supp(\eta_{(X,q^0,a)}) = \{q^{1x_u}, q^{1x_v}\} \cup \{q^{1y_u}, q^{1y_v}\}$ s. t. $q^{1x_u} R^{\setminus\{P\}}_{conf} q^{1x_v}$ and $q^{1y_u} R^{\setminus\{P\}}_{conf} q^{1y_v}$. Thereafter we construct $q^{\tilde{1}x} = \mu_s(q^{1x_u}) = \mu_s(q^{1x_v})$ and $q^{\tilde{1}y} = \mu_s(q^{1y_u}) = \mu_s(q^{1y_v})$. Then, $\eta_{(Y,\tilde{q^0},a)}$ is defined s. t. $\eta_{(Y,\tilde{q^0},a)}(\tilde{q}^{1x}) = \eta_{(X,q^0,a)}(q^{1x_u}) + \eta_{(X,q^0,a)}(q^{1x_v})$ and $\eta_{(Y,\tilde{q^0},a)}(\tilde{q}^{1y}) = \eta_{(X,q^0,a)}(q^{1y_u}) + \eta_{(X,q^0,a)}(q^{1y_v})$. We perform another time this procedure. by partitioning $supp(\eta_{(X,q^{1y_u},a)}) = \{q^{2x_u}\} \cup \{q^{2y_u}\}$ or $supp(\eta_{(X,q^{1y_v},a)}) = \{q^{2x_v}, q^{2x_w}\} \cup \{q^{2y_v}, q^{2y_w}\}$ arbitrarily. Indeed the obtai,ed result is the same: (i) $q^{1y_u} R^{\setminus\{P\}}_{conf} q^{1y_v}$ since they are both pre-image of $\tilde{q}^{1y}$ by μ_s, which means (ii) $q^{1y_u} R^{\setminus\{P\}}_{strict} q^{1y_v}$ since X is assumed to be P-fair. If we note $C_u = config(X)(q^{1y_u})$, $C_v = config(X)(q^{1y_v})$, $\varphi_u = created(X)(q^{1y_u})(c)$, $\varphi_v = created(X)(q^{1y_v})(c)$, $C_u \overset{c}{\Longrightarrow}_{\varphi_u} \eta_u$ and $C_v \overset{c}{\Longrightarrow}_{\varphi_v} \eta_v$ we have j) $C_u \setminus \{P\} = C_v \setminus \{P\}$, jj) $C_u \setminus \{P\} \overset{c}{\Longrightarrow}_{\varphi_u \setminus \{P\}} \eta_u \setminus \{P\}$ and jjj) $C_v \setminus \{P\} \overset{c}{\Longrightarrow}_{\varphi_v \setminus \{P\}} \eta_v \setminus \{P\}$ which implies jv) $\eta_u \setminus \{P\} = \eta_v \setminus \{P\}$.

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.

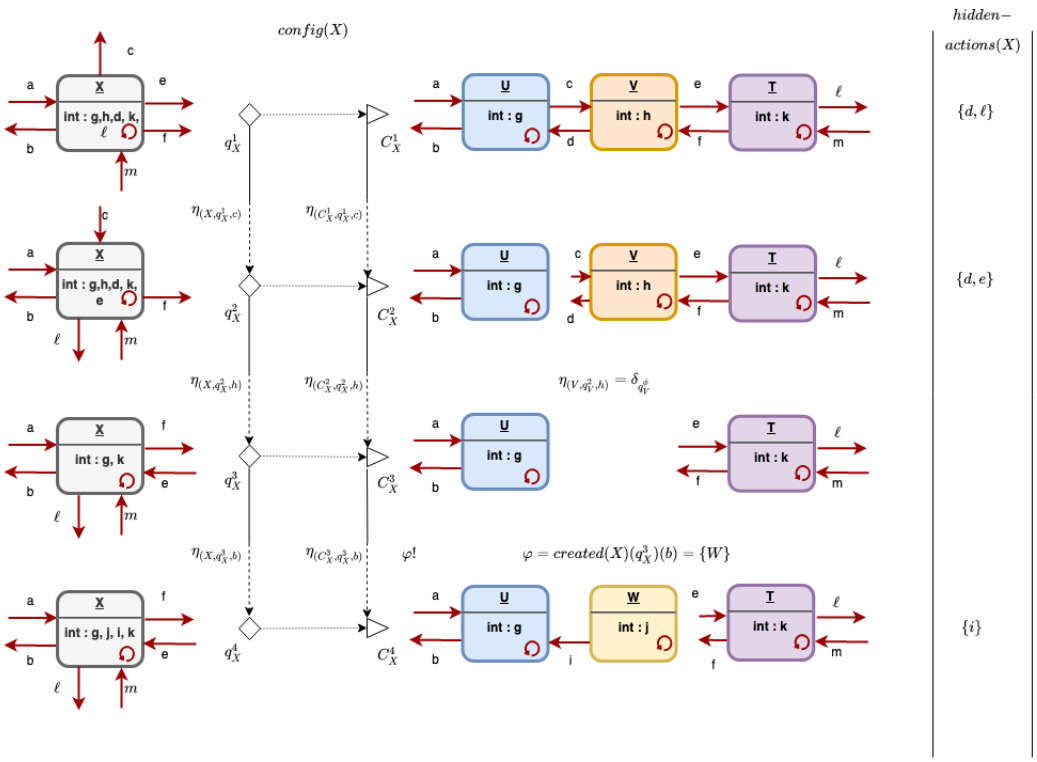


Fig. 26. Projection on PCA (part 1/2, the part 2/2 is in figure 27): the original PCA X

8.3 $Y = X \setminus \{\mathcal{A}\}$ is a PCA if X is $\mathcal{A}$-fair

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

Prepare the top/down transition preservation. We show a useful lemma to show $Y = X \setminus \{\mathcal{A}\}$ verifies the constraint 2 of top/down transition preservation.

Lemma 8.12 (corresponding transition after projection). *Let $\mathcal{A}$ be a PSIOA. Let X be a $\mathcal{A}$-fair PCA. Let $Y = X \setminus \{\mathcal{A}\}$. Let $(q_X, a, \eta_X) \in dtrans(X)$ and $a \in \widehat{sig}(config(X)(q_X) \setminus \{P\})$.*

Let $\eta'_X \in Disc(Q_{conf})$ the unique reduced configuration transition s. t. x0) $\eta_{(X,q_X,a)} \overset{f}{\leftrightarrow} \eta'_X$ with x1) $f = config(X)(q_X)$ and x2) $Config(X)(q_X) \overset{a}{\Longrightarrow}_{\varphi_X} \eta'_X$ where $\varphi_X = created(X)(q_X)(a)$.

Let $(q_Y, a, \eta_{(Y,q_Y,a)})$ with $\eta_{(Y,q_Y,a)} = \mu_d(\eta_X))$, $q_Y = \mu_s(q_X)$. Let $\eta'_Y = \eta'_X \setminus \{\mathcal{A}\}$ Then η'_Y is a reduced configuration transition that verifies y0) $\eta_{(Y,q_Y,a)} \overset{f'}{\leftrightarrow} \eta'_Y$ with y1) $f' = config(Y)(q_Y)$ and y2) $Config(Y)(q_Y) \overset{a}{\Longrightarrow}_{\varphi_Y} \eta'_Y$ where $\varphi_Y = \varphi_X \setminus \{\mathcal{A}\} = created(Y)(q_Y)(a)$.

Proof. We note $(Q_i^X)_{i\in\mathcal{I}}$ the partition of $supp(\eta_{X,q_X,a})$ s. t. $\forall i \in \mathcal{I}\ \forall q'_X, q''_X \in Q_i^X, q'_X R_{conf}^{\setminus\{\mathcal{A}\}} q''_X$. $\forall i \in \mathcal{I}$, we note $C_i^{\setminus\{\mathcal{A}\}} = config(q'_X) \setminus \{\mathcal{A}\}$ for an arbitrary element $q'_X \in Q_i^X$ and $C_i = \{C \in supp(\eta'_X) | C \setminus \mathcal{A} = C_i^{\setminus\{\mathcal{A}\}}\}$. Since x0) $\eta_{(X,q_X,a)} \overset{f}{\leftrightarrow} \eta'_X$ with x1) $f = config(X)(q_X)$, $(C_i)_{i\in\mathcal{I}}$ is a partition of $supp(\eta'_X)$.

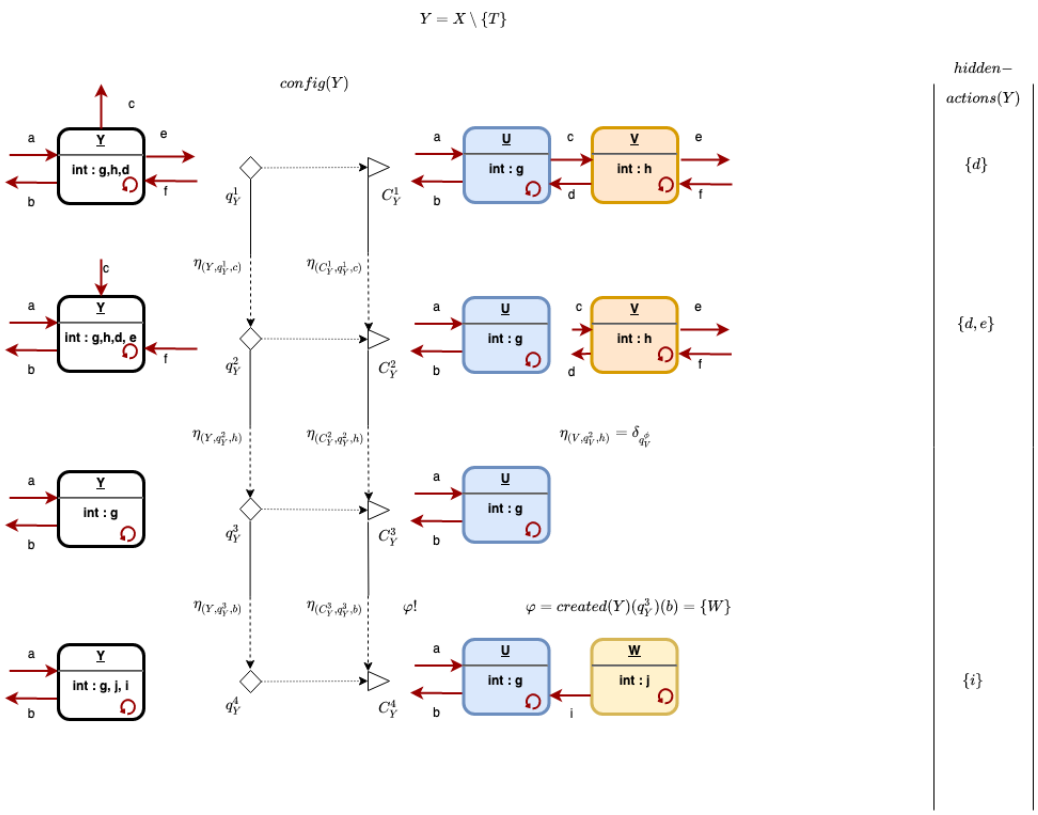


Fig. 27. Projection on PCA (part 2/2, the part 1/2 is in figure 26): the PCA $Y = X \setminus \{T\}$

For every $i \in \mathcal{I}$, we note $q_i^Y = \mu_s(q'_X)$ for an arbitrary element $q'_X \in Q_i^X$. By $\mu_s^{\mathcal{A}}$-correspondance, $config(q_i^Y) = C_i^{\setminus\{\mathcal{A}\}} = config(q'_X) \setminus \{\mathcal{A}\}$

By $\mu_d^{\mathcal{A}}$-correspondance, $\eta_{Y,q_y,a}(q'_y) = (\mu_d(\eta_{(X,q_X,a)}))(q'_Y) = \Sigma_{q'_X,\mu_s(q'_X)=q'_Y}\eta_{(X,q_X,a)}(q'_X) = \Sigma_{i\in I}\Sigma_{q'_X\in Q_i^X,\mu_s(q'_X)=q'_Y}\eta_{(X,q_X,a)}(q'_X)$. By assumption x0) and x1), $\eta_{(X,q_X,a)} \overset{f}{\leftrightarrow} \eta'_X$ with $f = config(X)$, thus $\eta_{Y,q_y,a}(q'_y) = \Sigma_{i\in\mathcal{I}}\Sigma_{q'_X\in Q_i^X,\mu_s(q'_X)=q'_Y}\eta'_X(config(X)(q'_X)) = \Sigma_{i\in\mathcal{I}}\Sigma_{C'_X\in C_i,C'_X\setminus\mathcal{A}=config(q'_Y)}\eta'_X(C'_X) = \Sigma_{C'_X,C'_X\setminus\mathcal{A}=config(q'_Y)}\eta'_X(C'_X)$

Therafter, we use the lemma 8.7 and get $\eta_{Y,q_y,a}(q'_Y) = \eta'_Y(config(Y)(q'_Y))$ with $\eta'_Y = \eta'_X \setminus \{\mathcal{A}\}$.

By definition of Y, $Config(Y)(q_Y = \mu_s(q_X)) = Config(X)(q_X)\setminus\{\mathcal{A}\}$. Then, since $a \in \widehat{sig}(config(X)(q_X)\setminus\{\mathcal{A}\})$, we can apply lemma 8.8. Thus $Config(Y)(q_Y) \overset{a}{\Longrightarrow}_{\varphi_Y} \eta'_Y$ with $\eta'_Y = \eta'_X\setminus\{\mathcal{A}\}$ and $\varphi_Y = (\varphi_X\setminus\{\mathcal{A}\})$. By $\mu_s^{\mathcal{A}}$-correspondance, $created(Y)(q_Y)(a) = created(X)(q_X)(a) \setminus \{\mathcal{A}\}$, thus $\varphi_Y = created(Y)(q_Y)(a)$.

Finally the restriction of $config(Y)$ on $supp(\eta_{Y,q_Y,a})$ is a bijection. Indeed, we note $f_1 : q_Y \mapsto Q_i^X$ s. t. $\{q_Y\} = \mu_s(Q_i^X)$, $f_2 : Q_i^X \mapsto C_i$ $f_3 : C_i \mapsto C_i^{\setminus\mathcal{A}}$. By construction, f_1 and f_3 are bijection. By bijectivity of the restriction of $config(X)$ on $supp(\eta_{X,q_X,a})$, f_2 is a bijection too. Moreover, the restriction f' of $config(Y)$ on $supp(\eta_{Y,q_Y,a})$ is $f_1 \circ f_2 \circ f_3$ and hence this is a bijection too.

□

We show a useful lemma to show $Y = X \setminus \{\mathcal{A}\}$ verifies the constraint 3 of bottom/up transition preservation.

Lemma 8.13 (existence of intrinsic transition). *Let $\mathcal{A}_k \in Autids$, let X be a $\mathcal{A}_k$-fair PCA, let $Y = X \setminus \{\mathcal{A}_k\}$ and $q_Y \in States(Y)$.*

If $\exists \eta'_Y \in Disc(Q_{conf}), a \in \widehat{sig}(Config(Y)(q_Y)), \varphi_Y = created(Y)(q_Y)(a)$ s. t. $Config(Y)(q_Y) \overset{a}{\Longrightarrow}_{\varphi_Y} \eta'_Y$ then

It exists $\exists q_X \in States(X), \mu_s(q_X) = q_Y, \eta'_X \in Disc(Q_{conf}), \eta'_Y = (\eta'_X \setminus \{\mathcal{A}_k\}), a \in \widehat{sig}(Config(X)(q_X) \setminus \{\mathcal{A}_k\}), \varphi_X = created(X)(q_X)(a)$ s. t. $Config(X)(x) \overset{a}{\Longrightarrow}_{\varphi_X} \eta'_X$.

Proof. By construction of $Y = X \setminus \{\mathcal{A}_k\}$, if $q_Y \in states(Y)$, it exists $q_X \in states(X), \mu_s(q_X) = q_Y, config(X)(q_X) \setminus \{\mathcal{A}_k\} = config(Y)(q_Y)$ and $created(X)(q_X)(a) = created(Y)(q_Y)(a) \setminus \{\mathcal{A}_k\}$. We note $C_X = (\mathbf{A}_X, \mathbf{S}_X) = config(X)(q_X)$ and $C_Y = (\mathbf{A}_Y, \mathbf{S}_Y) = config(Y)(q_Y)$ We treat two cases to show that $\eta_{(C_Y,a),p} = \eta_{(C_X,a),p} \setminus \{\mathcal{A}_k\}$:

- case 1) $\mathcal{A}_k \in \mathbf{A}_X$ with $\mathbf{S}_X(\mathcal{A}_k) = q_k$. We note $(\mathbf{A}_X = \mathcal{A}_1, ..., \mathcal{A}_k, ..., \mathcal{A}_n)$. Thus $(\mathbf{A}_Y = \mathcal{A}_1, ..., \mathcal{A}_{k-1}, \mathcal{A}_{k-1}, ..., \mathcal{A}_n)$. For every $i \in [1:n]$, we note $\eta^i = \eta_{(\mathcal{A}_i,q_i,a)}$ if $a \in \widehat{sig}(\mathcal{A}_i)(q_i)$ and $\eta^i = \delta_{q_i}$. By definition of companion distribution, $C_Y \overset{a}{\restriction} \eta_{(C_Y,a),p}$ and $C_X \overset{a}{\restriction} \eta_{(C_X,a),p}$ where $\eta_{(C_X,a),p}$ has the companion distribution $\eta^X_p = \eta^1 \otimes ... \otimes \eta^k \otimes ... \otimes \eta^n$, while $\eta_{(C_Y,a),p}$ has the companion distribution $\eta^Y_p = \eta^X_p \setminus \{\mathcal{A}_k\} = \eta^1 \otimes ... \otimes \eta^{k-1} \otimes \eta^{k+1} ... \otimes \eta^n$. Hence $\eta_{(C_Y,a),p} = \eta_{(C_X,a),p} \setminus \{\mathcal{A}_k\}$.
- case 2) $\mathcal{A}_k \notin \mathbf{A}_X$. Immediate, since we have $\mathbf{A}_X = \mathbf{A}_Y$ and $\mathcal{A}_k \notin \mathbf{A}_X$.

Let us note $\varphi_X = created(X)(q_X)(a)$ and $\varphi_Y = created(Y)(q_Y)(a) = \varphi_X \setminus \{\mathcal{A}_k\}$. We note η'_X the reduced configuration distribution generated by φ_X and $\eta_{(C_X,a),p}$ and by definition 5.11 of intrinsic transition, we have $C_X \overset{a}{\Longrightarrow}_{\varphi_X} \eta'_X$.

By definition 5.11 of intrinsic transition, $C_Y \overset{a}{\Longrightarrow}_{\varphi_Y} \eta'_Y$ with η'_Y generated by $\eta_{(C_Y,a),p} = \eta_{(C_X,a),p} \setminus \{\mathcal{A}_k\}$ and $\varphi_Y = \varphi_X \setminus \{\mathcal{A}_k\}$.

Thus, $\eta'_X = \eta'_Y \setminus \{\mathcal{A}_k\}$ by definition 8.4 of intrinsic transition projection, which ends the proof.

□

Now we are able to demonstrate that the PCA set is closed under deprivation.

Theorem 8.14 ($X \setminus \{P\}$ 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 construction of Y, $\bar{q}_Y = \mu^P_s(\bar{q}_X)$ and by μ_s-correspondence rule, $config(Y)(\bar{q}_Y) = config(X)(\bar{q}_X) \setminus \{P\}$. Since the constraint 1 is respected by X, it is a fortiori respected by Y.
- (Constraint 2) Let $(q_Y, a, \eta_{(Y,q_Y,a)}) \in dtrans(Y)$. By construction of Y, we know it exists $(q_X, a, \eta_{(X,q_X,a)}) \in dtrans(X)$ with $\eta_{(Y,q_Y,a)} = \mu_d(\eta_{(X,q_X,a)})$ and $q_Y = \mu_s(q_X)$. Then, because of constraint 2 ensured by X, we obtain it exists a reduced configuration distribution $\eta'_X \in Disc(Q_{conf})$ s. t. x0) $\eta_{(X,q_X,a)} \overset{f}{\leftrightarrow} \eta'_X$ with x1) $f = config(X)(q_X)$ and x2) $Config(X)(q_X) \overset{a}{\Longrightarrow}_{\varphi_X} \eta'_X$ where $\varphi_X = created(X)(q_X)(a)$. We can apply lemma 8.12 to obtain that $\eta'_Y = \eta'_X \setminus \{P\}$ is a reduced configuration transition that verifies y0) $\eta_{(Y,q_Y,a)} \overset{f'}{\leftrightarrow} \eta'_Y$ with y1) $f' = config(Y)(q_Y)$ and y2) $Config(Y)(q_Y) \overset{a}{\Longrightarrow}_{\varphi_Y} \eta'_Y$ where $\varphi_Y = \varphi_X \setminus \{P\} = created(Y)(q_Y)(a)$.
  This terminates the proof of constraint 2.
- (Constraint 3) Let $q_Y \in States(Y), \eta'_Y \in Disc(Q_{conf}), a \in \widehat{sig}(Config(Y)(q_Y)), \varphi_Y = created(Y)(y)(a)$ s. t. $Config(Y)(q_Y) \overset{a}{\Longrightarrow} \eta'_Y$
  Because of lemma 8.13, it implies it exists $q_X \in States(X), \mu_s(q_X) = q_Y$, s. t. x2) $config(X)(q_X) \overset{a}{\Longrightarrow}_{\varphi_X} \eta'_X$ with $\varphi_X = created(X)(x)(a), \eta'_Y = \eta'_X \setminus \{P\}, \varphi_Y = \varphi_X \setminus \{P\}$.

Because of constraint 3, it means $(q_X, a, \eta_{X,q_X,a}) \in dtrans(X)$ with x0) $\eta_{(X,q_X,a)} \overset{f}{\leftrightarrow} \eta'_X$ with x1) $f = config(X)(q_X)$. Since $q_Y = \mu_s(q_Y)$ and $a \in \widehat{sig}(Y)(q_Y))$, the construction of $dtrans(Y)$ implies $(q_Y, a, \eta_{(Y,q_Y,a)}) \in dtrans(Y)$ with $\eta_{(Y,q_Y,a)} = \mu_d^P(\eta_{(X,q_X,a)})$.
We can reapply lemma 8.12 to obtain that $\eta''_Y = \eta'_X \setminus \{P\}$ is a reduced configuration transition that verifies y0) $\eta_{(Y,q_Y,a)} \overset{f'}{\leftrightarrow} \eta''_Y$ with y1) $f' = config(Y)(q_Y)$ and y2) $Config(Y)(q_Y) \overset{a}{\Longrightarrow}_{\varphi_Y} \eta''_Y$ where $\varphi_Y = \varphi_X \setminus \{P\} = created(Y)(q_Y)(a)$. Finally $\eta''_Y = \eta'_Y = \eta'_X \setminus \{P\}$, which allows us to conclude that
For every $q_Y \in States(Y), a \in \widehat{sig}(Config(Y)(q_Y)), \varphi_Y = created(Y)(y)(a)$ s. t. $Config(Y)(q_Y) \overset{a}{\Longrightarrow}_{\varphi_Y} \eta'_Y$, with some reduced configuration distribution η'_Y, then $(q_Y, a, \eta_{(Y,q_Y,a)}) \in dtrans(Y)$ with $\eta_{(Y,q_Y,a)} \overset{f'}{\leftrightarrow} \eta'_Y$ where $f' = config(Y)(q_Y)$
This terminates the proof of constraint 3.

- (Constraint 4) Verified by construction (We recall that $\forall (q_Y, q_X) \in States(Y) \times States(X), q_Y = \mu_s^P(q_X), sig(Y)(q_Y) \triangleq hide(sig(config(Y)(q_Y), hidden\text{-}actions(Y)(q_Y))$ where $hidden\text{-}actions(Y)(q_Y) \triangleq hidden\text{-}actions(X)(q_X) \setminus pot\text{-}out(X)(q_X)(P)$.

□

9 RECONSTRUCTION

In the previous section, we have shown that $Y = X \setminus \mathcal{A}$ is a PCA (as long as X is $\mathcal{A}$-fair). In this section we will

(1) introduce the concept of simpleton wrapper $\tilde{\mathcal{A}}^{sw}$ that is a PCA that encapsulates $\mathcal{A}$.
(2) prove that $X \setminus \{\mathcal{A}\}$ and $\tilde{\mathcal{A}}^{sw}$ are partially-compatible (see theorem 9.13)
(3) There is a strong executions-matching from X to $(X \setminus \{\mathcal{A}\})||\tilde{\mathcal{A}}^{sw}$ in a restricted set of executions of X that do not create $\mathcal{A}$ (see theorem 9.19). Hence it is always possible to transfer a reasoning on X into a reasoning on $(X \setminus \{\mathcal{A}\})||\tilde{\mathcal{A}}^{sw}$ if no re-creation of $\mathcal{A}$ occurs.
(4) The operation of projection/deprivation and composition are commutative (see theorem 9.24).

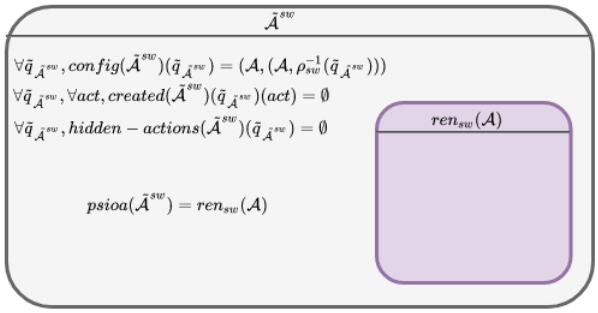
9.1 Simpleton wrapper : $\tilde{A}^{sw}$

Here we introduce simpleton wrapper $\tilde{A}^{sw}$, a PCA that only encapsulates $\tilde{A}^{sw}$

Definition 9.1 (Simpleton wrapper). (see figure 28) 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.
- $\forall \tilde{q}_{\tilde{\mathcal{A}}^{sw}} \in states(\tilde{\mathcal{A}}^{sw}), config(\tilde{\mathcal{A}}^{sw})(\tilde{q}_{\tilde{\mathcal{A}}^{sw}}) = reduced(\{\mathcal{A}\}, \mathsf{S} : \mathcal{A} \mapsto q_{\mathcal{A}} = ren_{sw}^{-1}(q_{\mathcal{A}}))$
- $\forall \tilde{q}_{\tilde{\mathcal{A}}^{sw}} \in states(\tilde{\mathcal{A}}^{sw}), \forall a \in \widehat{sig}(\tilde{\mathcal{A}}^{sw})(\tilde{q}^{sw}_{\tilde{\mathcal{A}}}), hidden\text{-}actions(\tilde{\mathcal{A}}^{sw})(\tilde{q}_{\tilde{\mathcal{A}}^{sw}}) = \emptyset$ 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.



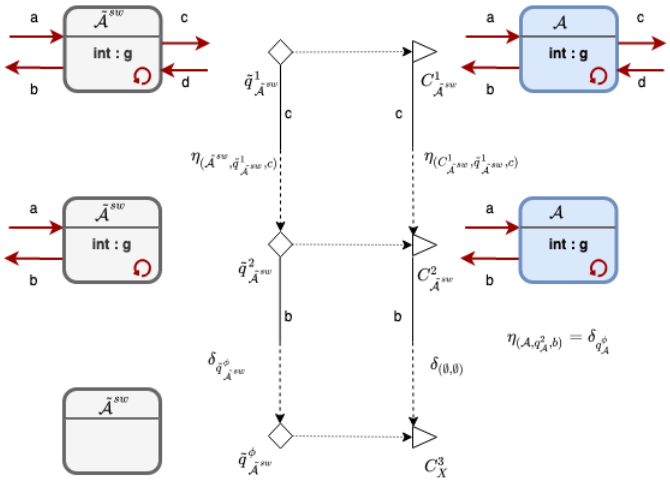


Fig. 28. Simpleton wrapper

Lemma 9.2. *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 $\mu \in Disc(Frags(\tilde{\mathcal{A}}^{sw}))$. For every schedule ρ, $apply_{\tilde{\mathcal{A}}^{sw}}(ren_{sw}(\mu), \rho)(ren_{sw}(\alpha)) = apply_{\mathcal{A}}(\mu, \rho)(\alpha)$.*

Proof. The only 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 $ren_{sw}(q)$.

Thus we can apply theorem 7.25.

□

9.2 Partial-compatibility of $(X_{\mathcal{A}} \setminus \{\mathcal{A}\})$ and $\tilde{\mathcal{A}}^{sw}$

In this subsection, 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 29).

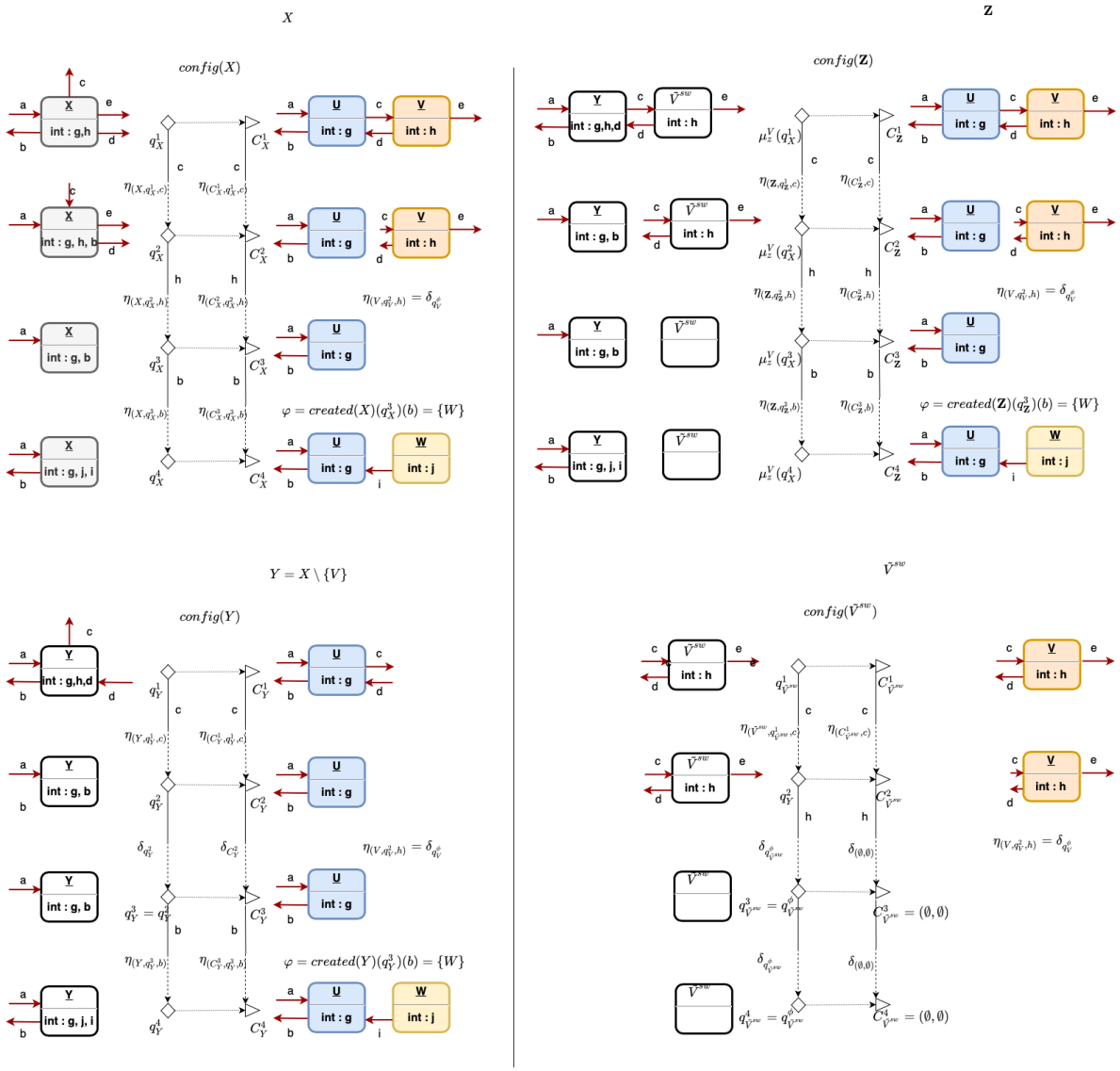


Fig. 29. Reconstruction of a PCA via $\mathbf{Z} = (X, X \setminus \{V\})$

Map X and $(X \setminus \{\mathcal{A}\}, \tilde{\mathcal{A}}^{sw})$. We first introduce two functions to map X and $(X \setminus \{\mathcal{A}\}, \tilde{\mathcal{A}}^{sw})$.

Definition 9.3 ($\mu_z^{\mathcal{A}}$ and $\mu_e^{\mathcal{A}}$: mapping of reconstruction). 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 $q_{\mathcal{A}}^{\phi} \in states(\mathcal{A})$ the (assumed) unique state s. t. $\widehat{sig}(\mathcal{A})(q_{\mathcal{A}}^{\phi}) = \emptyset$. We note:

- The function $X.\mu_z^{\mathcal{A}} : states(X) \to states(Y) \times states(\tilde{\mathcal{A}}^{sw})$ s. t. $\forall q_X \in states(X), X\mu_z^{\mathcal{A}}(q_X) = (X.\mu_s^{\mathcal{A}}(q_X), ren_{sw}(q_{\mathcal{A}}))$ with $q_{\mathcal{A}} = map(config(X)(q_X))(\mathcal{A})$ if $\mathcal{A} \in (auts(config(X)(q_X)))$ and $q_{\mathcal{A}} = q_{\mathcal{A}}^{\phi}$ otherwise.

- The function $X.\mu_e^{\mathcal{A}}$ that maps any alternating sequence $\alpha_X = q_X^0, a^1, q_X^1, a^2 ...$ of states and actions of X, to $\mu_e^{\mathcal{A}}(\alpha_X)$ the alternating sequence $\alpha_Z = X.\mu_z^{\mathcal{A}}(q_X^0), a^1, X.\mu_z^{\mathcal{A}}(q_X^1), a^2, ...$

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

Now, we recall definition 6.14 of $\mathcal{A}$-conservative PCA, an additional condition to allow the compatibility between $X \setminus \mathcal{A}$ and $\tilde{\mathcal{A}}^{sw}$.

Definition 9.4 ($\mathcal{A}$-conservative PCA (recall)). 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 $q_X \in States(X)$, $C_X = (\mathbf{A}_X, \mathbf{S}_X) = config(X)(q_X)$ s. t. $\mathcal{A} \in \mathbf{A}_X$ and $\mathbf{S}_X(\mathcal{A}) \triangleq q_{\mathcal{A}}$, $hidden\text{-}actions(X)(q_X) = hidden\text{-}actions(X)(q_X) \setminus \widehat{ext}(\mathcal{A})(q_{\mathcal{A}})$.

A $\mathcal{A}$-conservative PCA is a $\mathcal{A}$-fair PCA that does not hide any output action that could be an external action of $\mathcal{A}$.

Preservation of properties. Now we start a sequence of lemma (from lemma 9.5 to lemma 9.11) about properties preserved after reconstruction to eventually show in theorem 9.13 that $X \setminus \mathcal{A}$ and $\tilde{\mathcal{A}}^{sw}$ are partially-compatible.

The next lemma shows that reconstruction preserves signature compatibility.

Lemma 9.5 (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 $q_X \in states(X)$, $C_X = (\mathbf{A}_X, \mathbf{S}_X) = config(X)(q_X)$. Let $q_Y \in states(Y)$, $q_Y = \mu_s(q_X)$. Let $C_Y = (\mathbf{A}_Y, \mathbf{S}_Y) = config(Y)(q_Y)$.*

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 $sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_{\mathcal{A}}^{\phi}))$ are compatible and $sig(C_X) = sig(C_Y) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_{\mathcal{A}}^{\phi}))$.

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 $sig(\mathcal{A})(q_{\mathcal{A}}) = sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_{\mathcal{A}}^{\phi}))$.

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 $sig(\mathcal{A})(q_{\mathcal{A}}^{\phi}) = (\emptyset, \emptyset, \emptyset) = sig(\mathcal{A})(q_{\mathcal{A}}) = sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_{\mathcal{A}}^{\phi}))$ 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, $sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_{\mathcal{A}}^{\phi})) = sig(\mathcal{A})(q_{\mathcal{A}}^{\phi}) = (\emptyset, \emptyset, \emptyset)$. Thus $sig(C_X) = sig(C_Y) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_{\mathcal{A}}^{\phi}))$ □

The next lemma shows that reconstruction preserves signature.

Lemma 9.6 (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 $(q_Y, ren_{sw}(q_{\mathcal{A}})) = \mu_z^{\mathcal{A}}(q_X)$.*

Proof. The last lemma 9.5 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 $sig(X)(q_X) = sig(Y)(q_Y) \times sig(\mathcal{A})(q_{\mathcal{A}}) = sig(Y)(q_Y) \times sig(\tilde{\mathcal{A}}^{sw})(ren_{sw}(q_{\mathcal{A}}))$. □

The next lemma shows that reconstruction preserves partial-compatibility at any reachable state.

Lemma 9.7 (preservation of partial-compatibility at any reachable state). *Let $\mathcal{A} \in Autids$, X be a $\mathcal{A}$-conservative PCA, $Y = X \setminus \{\mathcal{A}\}$, $\mathbf{Z} = (Y, \tilde{\mathcal{A}}^{sw})$ 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. $\mu_z^{\mathcal{A}}(q_X) = q_Z$. Then $\mathbf{Z}$ is partially compatible at state q_Z (in the sense of definition 5.19).*

Proof. Since X is a $\mathcal{A}$-conservative PCA, the previous lemma 9.6 ensures that $sig(Y)(q_Y)$ 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 q_Z. □

Here, we show that reconstruction preserves probabilistic distribution of corresponding transition, as long as no creation of the concerned automaton occurs.

Lemma 9.8 (homomorphic transition without creation). *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. (i) $\mu_z(q_X) = q_Z$. Let $a \in sig(X)(x) = sig(Y)(y) \times sig(\tilde{\mathcal{A}}^{sw})(\tilde{q}_{\tilde{\mathcal{A}}^{sw}})$, verifying (ii: No creation from $\mathcal{A}$) If a is $\mathcal{A}$-exclusive in state q_X,then $created(X)(x)(a) = \emptyset$,*

- *If $\mathcal{A}$ is not created by a, i. e. if either*
  - *$\mathcal{A} \in auts(config(X)(x))$, or*
  - *$\mathcal{A} \notin auts(config(X)(x))$ and $\mathcal{A} \notin created(X)(x)(a)$ (X does not create $\mathcal{A}$ with probability 1)*

  Then $\eta_{(X,q_X,a)} \overset{\mu_z}{\leftrightarrow} \eta_{(\mathbf{Z},q_Z,a)}$
- *If $\mathcal{A}$ is created by a i. e. $\mathcal{A} \notin auts(config(X)(x))$ and $\mathcal{A} \in created(X)(x)(a)$ (X creates $\mathcal{A}$ with probability 1)*

  Then $\eta_{(X,q_X,a)} \overset{f^{\phi}}{\leftrightarrow} \eta_{(\mathbf{Z},q_Z,a)}$ where $f^{\phi} : q'_X \in supp(\eta_{(X,q_X,a)}) \mapsto (X.\mu_s^{\mathcal{A}}(q'_X), q^{\phi}_{\tilde{\mathcal{A}}^{sw}})$.

Proof. By lemma 9.6, we have $sig(X)(q_X) = sig(Y)(q_Y) \times sig(\mathcal{A})(q_{\mathcal{A}}) = sig(Y)(q_Y) \times sig(\tilde{\mathcal{A}}^{sw})(\tilde{q}_{\tilde{\mathcal{A}}^{sw}} = ren_{sw}(q_{\mathcal{A}}))$.

We note $C_X = (\mathbf{A}_X, \mathbf{S}_X) = config(X)(q_X), C_Y = (\mathbf{A}_Y, \mathbf{S}_Y) = config(Y)(q_Y), C_{\tilde{\mathcal{A}}^{sw}} = (\mathbf{A}_{\tilde{\mathcal{A}}^{sw}}, \mathbf{S}_{\tilde{\mathcal{A}}^{sw}}) = config(\tilde{\mathcal{A}}^{sw})(q_{\tilde{\mathcal{A}}^{sw}})$. By construction of μ_z, $C_X = C_Y \cup C_{\tilde{\mathcal{A}}^{sw}}$ with C_Y and $C_{\tilde{\mathcal{A}}^{sw}}$ compatible configuration (1).

We note $\varphi_X = created(X)(q_X)(a)$, $\varphi_Y = \varphi_X \setminus \{\mathcal{A}\}$, $\varphi_{\tilde{\mathcal{A}}^{sw}} = \emptyset$, $\varphi_Z = \varphi_X \cup \varphi_{\tilde{\mathcal{A}}^{sw}}$. If a is $\mathcal{A}$-exclusive in state q_X, then $\varphi_X = \varphi_Y = \emptyset$.

- If $\mathcal{A}$ is not created by a, then $\varphi_X = \varphi_Z$,
- If $\mathcal{A}$ is created by a, then $\varphi_X = \varphi_Z \cup \{\mathcal{A}\}$ and $\varphi_Z = \varphi_X \setminus \{\mathcal{A}\}$

Since X is a PCA and $(q_X, a, \eta_{(X,q_X,a)}) \in D_X$, the constraint 2 of top/down transition preservation says that it exists a unique reduced configuration distribution η'_X s. t. $\eta_{(X,q_X,a)} \overset{f^X}{\leftrightarrow} \eta'_X$ with $f^X = config(X)$ and $config(X)(q_X) \Longrightarrow_{\varphi_X} \eta'_X$ (2).

For Y (resp. $\tilde{\mathcal{A}}^{sw}$) we note $\eta_Y = \eta_{(Y,q_Y,a)}$ if $a \in \widehat{sig}(Y)(q_Y)$ and $\eta_Y = \delta_{q_Y}$ otherwise (resp. $\eta_{\tilde{\mathcal{A}}^{sw}} = \eta_{(\tilde{\mathcal{A}}^{sw},q_{\tilde{\mathcal{A}}^{sw}},a)}$ if $a \in \widehat{sig}(\tilde{\mathcal{A}}^{sw})(q_{\tilde{\mathcal{A}}^{sw}})$ and $\eta_{\tilde{\mathcal{A}}^{sw}} = \delta_{q_{\tilde{\mathcal{A}}^{sw}}}$ otherwise).

Since Y and $\tilde{\mathcal{A}}^{sw}$ are PCA, either because of the constraint 2 of top/down transition preservation or because a is not action of the signature, it exists a unique reduced configuration distribution η'_Y s. t. $\eta_Y \overset{f^Y}{\leftrightarrow} \eta'_Y$ with $f^Y = config(Y)$ and $config(Y)(q_Y) \Longrightarrow_{\varphi_Y} \eta'_Y$ (resp. $\eta'_{\tilde{\mathcal{A}}^{sw}}$ s. t. $\eta_{\tilde{\mathcal{A}}^{sw}} \overset{f^{\tilde{\mathcal{A}}^{sw}}}{\leftrightarrow} \eta'_{\tilde{\mathcal{A}}^{sw}}$ with $f^{\tilde{\mathcal{A}}^{sw}} = config(\tilde{\mathcal{A}}^{sw})$ and $config(\tilde{\mathcal{A}}^{sw})(q_{\tilde{\mathcal{A}}^{sw}}) \Longrightarrow_{\varphi_{\tilde{\mathcal{A}}^{sw}}} \eta'_{\tilde{\mathcal{A}}^{sw}}$) (3).

By construction $\forall (q'_Y, q'_{\tilde{\mathcal{A}}^{sw}}) \in states(Y) \times states(\tilde{\mathcal{A}}^{sw}), constitution(Y)(q'_Y) \cap constitution(\tilde{\mathcal{A}}^{sw})(q'_{\tilde{\mathcal{A}}^{sw}}) = \emptyset$ (and so $auts(config(Y)(q'_Y)) \cap auts(config(\tilde{\mathcal{A}}^{sw})(q'_{\tilde{\mathcal{A}}^{sw}})) = \emptyset$) which means (**) $base(C_Y, a, \varphi_Y) \cap base(C_{\tilde{\mathcal{A}}^{sw}}, a, \varphi_{\tilde{\mathcal{A}}^{sw}}) = \emptyset$.

The conjonction of (1), (2), (3) and (**) allows us to apply the lemma 5.26. This means

- by item 4 of lemma 5.26: $merge((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y)) \overset{f^s}{\leftrightarrow} join((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$ with $f^s : C'_Z \mapsto (C'_Y, C'_{\tilde{\mathcal{A}}^{sw}})$ s. t. i) $C'_Z = C'_Y \cup C'_{\tilde{\mathcal{A}}^{sw}}$, ii) $\mathcal{A} \notin C'_Y$ and iii) $\forall \mathcal{B} \neq \mathcal{A}, \mathcal{B} \notin C'_{\tilde{\mathcal{A}}^{sw}}$ (4)
- by item 5 of lemma 5.26: $C_X \overset{a}{\Longrightarrow}_{\varphi_Z} merge((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$ (5)

Furthermore $\eta_{\mathbf{Z},q_Z,a} = \eta_Y \otimes \eta_{\tilde{\mathcal{A}}^{sw}}$. So by (3), $\eta_{\mathbf{Z},q_Z,a} \overset{f^Z}{\longleftrightarrow} join((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$ (***) with $f^Z : q'_Z = (q'_Y, q'_{\tilde{\mathcal{A}}^{sw}}) \mapsto (config(Y)(q'_{\tilde{\mathcal{A}}^{sw}}), config(\tilde{\mathcal{A}}^{sw})(q'_{\tilde{\mathcal{A}}^{sw}}))$.

Now we deal have to separate the treatment of the two cases:

- If $\mathcal{A}$ is not created by a, since $\varphi_Z = \varphi_X$, because of (5) and (2), $merge((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y)) = \eta'_X$ and because of (2) $\eta_{(X,q_X,a)} \overset{f^X}{\leftrightarrow} merge((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$ (6). Because of (6) and (4), $\eta_{(X,q_X,a)} \overset{g}{\longleftrightarrow} join((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$ with $g = f^s \circ f^X$. Hence, if $\mathcal{A}$ is not created by a $\eta_{(X,q_X,a)} \overset{h}{\longleftrightarrow} \eta_{(\mathbf{Z},q_Z,a)}$ with $h = (f^Z)^{-1} \circ f^s \circ f^X = \mu_z$ which ends the proof for this case.
- If $\mathcal{A}$ is created by a, we have both
  - $C_X \overset{a}{\Longrightarrow}_{\varphi_Z} merge((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$
  - $C_X \overset{a}{\Longrightarrow}_{\varphi_Z \cup \{\mathcal{A}\}} \eta'_X$

  which means $C_X \overset{a}{\rightharpoonup} \eta'_p$ with
  - $merge((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$ generated by η'_p and φ_Z and
  - η'_X generated by η'_p and $\varphi_Z \cup \{\mathcal{A}\}$.

  Thus $\eta'_X \overset{g^\phi}{\longleftrightarrow} merge((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$ with $g^\phi : C'_X = C'_Y \cup \bar{C}_{\tilde{\mathcal{A}}^{sw}} \mapsto C'_Y$. where $\bar{C}_{\tilde{\mathcal{A}}^{sw}}(\{\mathcal{A}\}, \mathbf{S}'_{\tilde{\mathcal{A}}^{sw}} : \mathcal{A} \mapsto \bar{q}_{\tilde{\mathcal{A}}^{sw}})$.
  To summerize, we have:
  - $\eta_{(X,q_X,a)} \overset{f^X}{\longleftrightarrow} \eta'_X$
  - $\eta'_X \overset{g^\phi}{\longleftrightarrow} merge((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$
  - $merge((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y)) \overset{f^s}{\longleftrightarrow} join((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$
  - $\eta_{(\mathbf{Z},q_z,a)} \overset{f^Z}{\longleftrightarrow} join((\eta'_{\tilde{\mathcal{A}}^{sw}}, \eta'_Y))$

  Hence $\eta_{(X,q_X,a)} \overset{h}{\longleftrightarrow} \eta_{(\mathbf{Z},q_Z,a)}$ with $f^\phi = (f^Z)^{-1} \circ f^s \circ g^\phi \circ f^X$, i. e.
  $f^\phi : q'_X \in supp(\eta_{(X,q_X,a)}) \mapsto (X.\mu_s^{\mathcal{A}}(q'_X), q^\phi_{\tilde{\mathcal{A}}^{sw}})$, which ends the proof for this case.

□

The second case where $\mathcal{A}$ is created will not be used before section 11.

We take advantage of the lemma 9.11 used for theorem 9.13 to introduce the notion of twin PCA and extends directly the lemma 9.11 and theorem 9.13 to twin PCA.

Definition 9.9 ($X_{\bar{q}_X \to \bar{q}'_X}$). Let $X = (Q_X, \bar{q}_X, sig(X), D_X)$ be a PSIOA and $\bar{q}'_X \in reachable(X)$. We note $X_{\bar{q}_X \to \bar{q}'_X}$ the PSIOA $X' = (Q_X, \bar{q}'_X, sig(X), D_X)$.

Two PCA X and X' are $\mathcal{A}$-twin if they differ only by their start state where one of them corresponds to $\mathcal{A}$-creation.

Definition 9.10 ($\mathcal{A}$-twin). Let $\mathcal{A} \in Autids$. Let X, X' be PCA. We say that $X' = X_{\bar{q}_X \to \bar{q}_{X'}}$ 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. either $X' = X$ or $\mathcal{A} \in config(X')(\bar{q}_{X'})$ 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 9.11 (partial surjectivity 1). *Let $\mathcal{A} \in Autids$. Let X be a PCA $\mathcal{A}$-conservative and X' a $\mathcal{A}$-twin of X. 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 the presence of $\mathcal{A}$ in α, i. e. $\forall s \in [0, k-1]$, $q^s_{\tilde{\mathcal{A}}^{sw}} \neq ren_{sw}(q^{\phi}_{\mathcal{A}})$.

Then it exists $\tilde{\alpha} \in Execs(X')$, s. t. $X'.\mu_e^{\mathcal{A}}(\tilde{\alpha}) = \alpha$.

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

Basis: case 1) $\mathcal{A} \in config(X')(\bar{q}_{X'})$: We have $\mu_z(\bar{q}_{X'}) = (\bar{q}_{Y'}, ren_{sw}(\bar{q}_{\mathcal{A}}))$. Hence $\mu_e(\bar{q}_{X'}) = (\bar{q}_{Y'}, ren_{sw}(\bar{q}_{\mathcal{A}}))$.

case 2) $\mathcal{A} \notin config(X')(\bar{q}_{X'})$, (necessarily $X = X'$): $\mu_z(\bar{q}_{X'}) = (\bar{q}_{Y'}, ren_{sw}(q^{\phi}_{\mathcal{A}}))$. Hence $\mu_e(\bar{q}_{X'}) = (\bar{q}_{Y'}, ren_{sw}(q^{\phi}_{\mathcal{A}}))$.

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 $a^{s+1} \in sig(X)(\tilde{q}^s)$. Thereafter, by construction of $X \setminus \{\mathcal{A}\}$ it exists $\tilde{q}^{s+1}$ s. t. $q^{s+1} = \mu_z^{\mathcal{A}}(\tilde{q}^{s+1})$. Finally, since no creation of and from $\mathcal{A}$ occurs by assumption of presence of $\mathcal{A}$, we can use lemma 9.8 of homomorphic transition which give $\eta_{(X,\tilde{q}^s,a^{s+1})} \overset{\mu_z}{\leftrightarrow} \eta_{(\mathbf{Z},q^s,a^{s+1})}$ which means $\tilde{q}^{s+1} \in supp(\eta_{(X,\tilde{q}^s,a^{s+1})})$ which ends the induction and so the proof. □

Before using lemma 9.11 and 9.7 to demonstrate theorem 9.13 of partial compatibility after reconstruction, we take the opportunity to extend lemma 9.11:

Lemma 9.12 (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 9.11. 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. □

Now we can use lemma 9.11 and 9.7 to demonstrate theorem 9.13 of partial compatibility after reconstruction.

Theorem 9.13 (Partial-compatibility after resconstruction). *Let $\mathcal{A} \in Autids$. Let X be a PCA $\mathcal{A}$-conservative s. t. $\forall q_X \in states(X)$, for every action a $\mathcal{A}$-exclusive in q_X, $created(X)(q_X)(a) = \emptyset$. Let X' ba a $\mathcal{A}$-twin of X and $Y' = X' \setminus \{\mathcal{A}\}$. Then Y' and $\tilde{\mathcal{A}}^{sw}$ are partially-compatible.*

Proof. Let $\mathbf{Z'} = (Y', \tilde{\mathcal{A}}^{sw})$. Let α be a pseudo-execution of $\mathbf{Z'}$ with Let $lstate(\alpha) = q_{\mathbf{Z}} = (q_{Y'}, q_{\tilde{\mathcal{A}}^{sw}})$. 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 (*) $\mathcal{A}$ cannot be re-created after destruction by neither Y or $\tilde{\mathcal{A}}^{sw}$ and (**) $\forall q_X \in states(X)$, for every action a $\mathcal{A}$-exclusive in q_X, $created(X)(q_X)(a) = \emptyset$ we can use the previous lemma 9.11 to show it exists $\tilde{\alpha} \in Execs(X')$, s. t. $\mu_e(\tilde{\alpha}) = \alpha$. Thus, $lstate(\alpha) = \mu_z(lstate(\tilde{\alpha}))$ which means $\mathbf{Z'}$ is partially-compatible at $lstate(\alpha)$ by lemma 9.7. 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.

9.3 Execution-matching from X to $X \setminus \{\mathcal{A}\}||\tilde{\mathcal{A}}^{sw}$

In this subsection, we show in theorem 9.19 that $X.\mu_e^{\mathcal{A}}$ is a (incomplete) PCA executions-matching from X to $(X \setminus \{\mathcal{A}\})||\tilde{\mathcal{A}}^{sw}$ in a restricted set of executions of X that do not create $\mathcal{A}$.

We start by defining the restricted set of executions of X that do not create $\mathcal{A}$ with definitions 9.14 and 9.15.

Definition 9.14 (execution without creation). Let $\mathcal{A}$ be a PSIOA. Let X be a PCA , we note $execs\text{-}without\text{-}creation(X)(\mathcal{A})$ the set of executions of X without creation of $\mathcal{A}$, i. e. $execs\text{-}without\text{-}creation(X)(\mathcal{A}) = \{\alpha = q^0a^1q^1...a^kq^k \in Execs(X) | \forall i \in [0, |\alpha|], \mathcal{A} \notin auts(config(X)(q^i)) \implies \mathcal{A} \notin auts(config(X)(q^{i+1}))\}$.

Definition 9.15 (reachable-by). Let X be a PSIOA or a PCA. Let $Execs'_X \subseteq Execs(X)$. We note $reachable\text{-}by(Execs'_X)$ the set of states of X reachable by an execution of $Execs'_X$, i. e. $reachable\text{-}by(Execs'_X) = \{q \in states(X) | \exists \alpha \in Execs'_X, lstate(\alpha) = q\}$

The next 2 lemma show that reconstruction preserves configuration and signature. They will be sufficient to show that the restriction of $\mu_e^{\mathcal{A}}$ on $reachable\text{-}by(execs\text{-}without\text{-}creation(X)(\mathcal{A}))$ is a PCA executions-matching.

Lemma 9.16 (μ_z configuration preservation). *Let $\mathcal{A} \in Autids$. Let X be a $\mathcal{A}$-conservatiee PCA, $Y = X \setminus \mathcal{A}$, $Z = Y||\mathcal{A}^{sw}$. Let $q_X \in states(X), q_Z = (q_Y, q_{\tilde{\mathcal{A}}^{sw}}) \in states(Z)$ s. t. $\mu_z(q_X) = q_Z$. Then $config(X)(q_X) = config(Z)(q_Z)$.*

Proof. By definition of composition of PCA, $config(Z)(q_Z) = config(Y)(q_Y) \cup config(\tilde{\mathcal{A}}^{sw})(q_{\tilde{\mathcal{A}}^{sw}})$. (*)
Also, by $\mu_z^{\mathcal{A}}$-correspondence, $config(X)(q_X) \setminus \mathcal{A} = config(Y)(q_Y)$ (**).
We deal with the two cases $\widehat{sig}(\tilde{A}^{sw})(q_{\tilde{\mathcal{A}}^{sw}}) = \emptyset$ or $\widehat{sig}(\tilde{A}^{sw})(q_{\tilde{\mathcal{A}}^{sw}}) \neq \emptyset$

- If $\widehat{sig}(\tilde{A}^{sw})(q_{\tilde{\mathcal{A}}^{sw}}) = \emptyset$, then $\mathcal{A} \notin aut(config(X)(q_X))$ which means, that $config(X)(q_X) = config(X)(q_X) \setminus \mathcal{A}$ (1). Furthermore, $config(\tilde{\mathcal{A}}^{sw})(q_{\tilde{\mathcal{A}}^{sw}}) = (\emptyset, \emptyset)$ (2) .Because of (**) and (1), $config(X)(q_X) = config(Y)(q_Y)$ and because of (*) and (2), $config(X)(q_X) = config(Z)(q_Z)$.
- If $\widehat{sig}(\tilde{A}^{sw})(q_{\tilde{\mathcal{A}}^{sw}}) \neq \emptyset$, then $\mathcal{A} \in aut(config(X)(q_X))$. We note $C_{\mathcal{A}} = config(\tilde{\mathcal{A}}^{sw})(q_{\tilde{\mathcal{A}}^{sw}}) = (\{\mathcal{A}\}, \mathbf{S} : \mathcal{A} \mapsto map(config(X)(q_X))(\mathcal{A}))$. By (*), $config(Z)(q_Z) = config(Y)(q_Y) \cup C_{\mathcal{A}}$ and by (**) $config(Y)(q_Y) \cup C_{\mathcal{A}} = config(X)(q_X) \setminus \mathcal{A} \cup C_{\mathcal{A}} = config(X)(q_X)$. Hence, $config(X)(q_X) = config(Z)(q_Z)$

Thus in all cases, $config(X)(q_X) = config(Z)(q_Z)$ which ends the proof.

□

Lemma 9.17 (μ_z signature-preservation). *Let $\mathcal{A} \in Autids$. Let X be a $\mathcal{A}$-conservatiee PCA, $Y = X \setminus \mathcal{A}$, $Z = Y||\mathcal{A}^{sw}$. Let $q_X \in states(X), q_Z = (q_Y, q_{\tilde{\mathcal{A}}^{sw}}) \in states(Z)$ s. t. $\mu_z(q_X) = q_Z$. Then $sig(X)(q_X) = sig(Z)(q_Z)$.*

Proof. By lemma 9.6 of preservation of signature $sig(X)(q_X) = sig(Y)(q_Y) \times sig(\tilde{\mathcal{A}}^{sw})(q_{\tilde{\mathcal{A}}^{sw}})$. By definition of composition of PCA, $sig(Z)(q_Z) = sig(Y)(q_Y) \times sig(\tilde{\mathcal{A}}^{sw})(q_{\tilde{\mathcal{A}}^{sw}})$ which ends the proof. □

Now we can states our strong PCA executions-matching:

Definition 9.18. Let $\mathcal{A}$ be a PSIOA. Let X be a $\mathcal{A}$-conservative PCA. Let $Y = X \setminus \{\mathcal{A}\}$ and $Z = Y||\tilde{\mathcal{A}}^{sw}$.
We define $(X.\tilde{\mu}_z^{\mathcal{A}}, X.\tilde{\mu}_{tr}^{\mathcal{A}}, X.\tilde{\mu}_e^{\mathcal{A}})$ (noted $(\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_{tr}^{\mathcal{A}}, \tilde{\mu}_e^{\mathcal{A}})$ when it is clear in the context) as follows:

- $\tilde{\mu}_z^{\mathcal{A}}$ the restriction of $\mu_z^{\mathcal{A}}$ on $reachable\text{-}by(execs\text{-}without\text{-}creation(X)(\mathcal{A}))$.
- $f^{tr} : (q_X, a, \eta_{(X,q_X,a)}) \in D'_X \mapsto (\tilde{\mu}_z^{\mathcal{A}}(q_X), a, \eta_{(Z,\tilde{\mu}_z^{\mathcal{A}}(q_X),a)})$ where $D'_X = \{(q_X, a, \eta_{(X,q_X,a)}) \in D_X | q_X \in reachable\text{-}by(execs\text{-}without\text{-}creation(X)(\mathcal{A})), (\mathcal{A} \notin auts(config(X)(q_X) \implies \mathcal{A} \notin created(X)(q_X)(a))\}$.
- $\tilde{\mu}_e^{\mathcal{A}}$ the restriction of $\mu_e^{\mathcal{A}}$ on $execs\text{-}without\text{-}creation(X)(\mathcal{A})$.

Theorem 9.19 (execution-matching after reconstruction). *Let $\mathcal{A}$ be a PSIOA. Let X be a $\mathcal{A}$-conservative PCA. Let $Y = X \setminus \{\mathcal{A}\}$. The triplet $(\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_{tr}^{\mathcal{A}}, \tilde{\mu}_e^{\mathcal{A}})$ is a strong PCA executions-matching from X to $Y||\tilde{\mathcal{A}}^{sw}$ if $\mathcal{A} \in auts(config(X_{\mathcal{A}})(start(X_{\mathcal{A}})))$ and from X to $Y||\tilde{\mathcal{A}}^{sw}_{\bar{q}_{\tilde{\mathcal{A}}^{sw}} \to q^{\phi}_{\tilde{\mathcal{A}}^{sw}}}$ otherwise.*

Proof. We note $Z = Y||\tilde{\mathcal{A}}^{sw}$ and $Z^{\phi} = Y||\tilde{\mathcal{A}}^{sw}_{\bar{q}_{\tilde{\mathcal{A}}^{sw}} \to q^{\phi}_{\tilde{\mathcal{A}}^{sw}}}$

- $\tilde{\mu}_z^{\mathcal{A}}$ is a strong PCA-state-matching since
  - starting state preservation is ensured by construction:
    - $\mathcal{A} \in auts(config(X_{\mathcal{A}})(start(X_{\mathcal{A}}))) : \tilde{\mu}_z^{\mathcal{A}}(\bar{q}_X) = \bar{q}_Z$
    - $\mathcal{A} \notin auts(config(X_{\mathcal{A}})(start(X_{\mathcal{A}}))) : \tilde{\mu}_z^{\mathcal{A}}(\bar{q}_X) = \bar{q}_{Z^{\phi}}$
  - signature preservation is ensured $\forall (q_X, q_Z) \in states(X) \times states(Z)$ s. t. $q_Z = \tilde{\mu}_z^{\mathcal{A}}(q_X)$, $sig(X)(q_X) = sig(Z)(q_Z)$ by lemma 9.17 of signature preservation of μ_z.
- $D'_X \triangleq dom(\tilde{\mu}_{tr}^{\mathcal{A}})$ is eligible to PCA transition-matching (and thus $(\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_{tr}^{\mathcal{A}})$ is a strong PCA-transition-matching) since
  - matched state preservation is ensured: $\forall \eta_{(X,q_X,a)} \in D'_X$, $q_X \in dom(\tilde{\mu}_z^{\mathcal{A}})$ by construction of D'_X
  - equitable corresponding distribution is ensured: $\forall \eta_{(X,q_X,a)} \in D'_X, \forall q'' \in supp(\eta_{(X,q_X,a)}), \eta_{(X,q_X,a)}(q'') = \eta_{(Z,\tilde{\mu}_z^{\mathcal{A}}(q_X),a)}(\tilde{\mu}_z^{\mathcal{A}}(q''))$ by lemma 9.8 of homomorphic transition.
- $(\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_{tr}^{\mathcal{A}}, \tilde{\mu}_e^{\mathcal{A}})$ is the PCA-execution-matching induced by $(\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_{tr}^{\mathcal{A}})$. and correctly verifies:
  - For each state q in an execution in $execs\text{-}without\text{-}creation(X)(\mathcal{A})$, $q \in dom(\tilde{\mu}_z^{\mathcal{A}})$.

Then, the triplet $(\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_{tr}^{\mathcal{A}}, \tilde{\mu}_e^{\mathcal{A}})$ is a strong PCA-execution-matching from X to Z if $\mathcal{A} \in auts(config(X_{\mathcal{A}})(start(X_{\mathcal{A}})))$: $\tilde{\mu}_z^{\mathcal{A}}(\bar{q}_X) = \bar{q}_Z$ and from X to Z^{ϕ} otherwise.

□

extension and continuation of $(\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_{tr}^{\mathcal{A}}, \tilde{\mu}_e^{\mathcal{A}})$. Now, we continue the executions-matching $(\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_{tr}^{\mathcal{A}}, \tilde{\mu}_e^{\mathcal{A}})$ to deal with $\mathcal{A}$ creation at very last action.

Definition 9.20 (Preparing continuation of PCA executions-matching from X to Z). Let $\mathcal{A}$ be a PSIOA. Let X be a $\mathcal{A}$-conservative PCA. We define

- $execs\text{-}with\text{-}only\text{-}one\text{-}creation\text{-}at\text{-}last\text{-}action(X)(\mathcal{A}) = \{\alpha' = \alpha^\frown q, a, q' \in Execs(X) | \alpha \in execs\text{-}without\text{-}creation(X)(\mathcal{A}) \wedge \alpha' \notin execs\text{-}without\text{-}creation(X)(\mathcal{A})\}$.
- $\tilde{\mu}_z^{\mathcal{A},+} : q_X \in reachable\text{-}by(execs\text{-}with\text{-}only\text{-}one\text{-}creation\text{-}at\text{-}last\text{-}action(X)(\mathcal{A})) \mapsto (\tilde{\mu}_s^{\mathcal{A}}(q_{Y_{\mathcal{A}}}), q_{\mathcal{A}}^{\phi})$.
- $\tilde{\mu}_{tr}^{\mathcal{A},+} : (q_X, a, \eta_{(X,q_X,a)}) \in dom(\tilde{\mu}_{tr}^{\mathcal{A}}) \cup D_X'' \mapsto (\tilde{\mu}_z^{\mathcal{A}}(q_X), a, \eta_{(X,\tilde{\mu}_z^{\mathcal{A}}(q_X),a)})$ where
  $D_X'' = \{(q_X, a, \eta_{(X,q_X,a)}) \in dtrans(X) | q_X \in reachable\text{-}by(execs\text{-}without\text{-}creation\text{-}at\text{-}last\text{-}action(X)(\mathcal{A})) \wedge \mathcal{A} \notin auts(config(X)(q_X)) \wedge \mathcal{A} \in created(X)(q_X)(a)\}$

We show that $dom(\tilde{\mu}_{tr}^{\mathcal{A},+}) \setminus dom(\tilde{\mu}_{tr}^{\mathcal{A}})$ verifies the equitable corresponding property of definition 7.7.

Lemma 9.21 (Continuation of PCA transitions-matching from X to Z). *Let $\mathcal{A}$ be a PSIOA. Let X be a $\mathcal{A}$-conservative PCA. Let $Y = X \setminus \{\mathcal{A}\}$ and $Z = Y||\tilde{\mathcal{A}}^{sw}$.*

$\forall (q_X, a, \eta_{(X,q_X,a)}) \in dom(\tilde{\mu}_{tr}^{\mathcal{A},+}) \setminus dom(\tilde{\mu}_{tr}^{\mathcal{A}}), \forall q_X' \in supp(\eta_{(X,q_X,a)}), \eta_{(X,q_X,a)}(q_X') = \eta_{(Z,\tilde{\mu}_z^{\mathcal{A}}(q_X),a)}(\tilde{\mu}_z^{\mathcal{A},+}(q_X'))$

Proof. By configuration preservation, $Conf = config(X)(q_X) = config(Z)(\tilde{\mu}_z^{\mathcal{A}}(q_X))$. We have $Conf \xrightarrow{a} \eta_{(Conf,a),p}$. Moreover, by μ_s-correspondence rule, $\varphi_X \setminus \{\mathcal{A}\} = \varphi_Z$, with $\varphi_X = created(X)(q_X)(a)$ and $\varphi_Z = created(Z)(\tilde{\mu}_z^{\mathcal{A}}(q_X))(a)$.

Hence $Conf \overset{a}{\Longrightarrow}_{\varphi_X} \eta_X'$ with η_X' generated by φ_X and $\eta_{(Conf,a),p}$, while $Conf \overset{a}{\Longrightarrow}_{\varphi_Z} \eta_Z'$ with η_Z' generated by φ_Z and $\eta_{(Conf,a),p}$.

Since $\mathcal{A}$ is created, for every $Conf_Z' = (\mathbf{A}_Z', \mathbf{S}_Z')$ with $\mathcal{A} \notin \mathbf{A}_Z$, for every $Conf_X' = (\mathbf{A}_X', \mathbf{S}_X')$ with $\mathbf{A}_X' = \mathbf{A}_Z' \cup \{\mathcal{A}\}$ where $\mathbf{S}_X'(\mathcal{A}) = \bar{q}_{\mathcal{A}}$ and $\mathbf{S}_X'$ agrees with $\mathbf{S}_Z'$ on $\mathbf{A}_Z'$, $\eta_Z'(Conf_Z') = \eta_X'(Conf_X')$, while $\eta_X'(Conf_X'') = 0$ for every $Conf_X'' = (\mathbf{A}_X'', \mathbf{S}_X'')$ s. t either $\mathcal{A} \notin \mathbf{A}_X''$ or $\mathcal{A} \in \mathbf{A}_X''$ but $\mathbf{S}_X''(\mathcal{A}) \neq \bar{q}_{\mathcal{A}}$. So $\eta_{(Z,\tilde{\mu}_z^{\mathcal{A}}(q_X),a)}(\tilde{\mu}_z^{\mathcal{A},+}(q_X')) = \eta_Z'(config(Z)(\tilde{\mu}_z^{\mathcal{A},+}(q_X'))) = \eta_X'((config(X)(q_X'))) = \eta_{(X,q_X,a)}(q_X')$ which ends the proof.

□

Since $dom(\tilde{\mu}_{tr}^{\mathcal{A},+}) \setminus dom(\tilde{\mu}_{tr}^{\mathcal{A}})$ verifies the equitable corresponding property of definition 7.7, we can define a continuation of $(\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_{tr}^{\mathcal{A}}, \tilde{\mu}_e^{\mathcal{A}})$ that deal with $\mathcal{A}$-creation at very last action.

Definition 9.22 (Continuation of PCA executions-matching from X to Z). Let $\mathcal{A}$ be a PSIOA. Let X be a $\mathcal{A}$-conservative PCA. Let $Y = X \setminus \{\mathcal{A}\}$ and $Z = Y||\tilde{\mathcal{A}}^{sw}$. Let $D_X'' = dom(\tilde{\mu}_z^{\mathcal{A},+}) \setminus dom(\tilde{\mu}_z^{\mathcal{A}})$. Since $\forall (q_X, a, \eta_{(X,q_X,a)}) \in D_X'', \forall q_X' \in supp(\eta_{(X,q_X,a)}), \eta_{(X,q_X,a)}(q_X') = \eta_{Z,\tilde{\mu}_z^{\mathcal{A}}(q_X),a)}(\tilde{\mu}_z^{\mathcal{A},+}(q_X'))$ by previous lemma 9.21, we can define:

$((\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_z^{\mathcal{A},+}), \tilde{\mu}_{tr}^{\mathcal{A},+}, \tilde{\mu}_e^{\mathcal{A},+}))$ the $(\tilde{\mu}_z^{\mathcal{A},+}, D_X'')$-continuation of $(\tilde{\mu}_z^{\mathcal{A}}, \tilde{\mu}_{tr}^{\mathcal{A}}, \tilde{\mu}_e^{\mathcal{A}})$.

We terminate this subsection by showing the $\mathcal{E}$-extension of our continued PCA executions-matching is always well-defined.

Theorem 9.23 (extension of continued executions-matching after reconstruction). *Let $\mathcal{A}$ be a PSIOA. Let X be a $\mathcal{A}$-conservative PCA. Let $Y = X \setminus \{\mathcal{A}\}$ and $Z = Y||\tilde{\mathcal{A}}^{sw}$. Let $\tilde{\mathcal{E}}$ partially-compatible with both X and Z. The $\tilde{\mathcal{E}}$-extension of $((X.\tilde{\mu}_z^{\mathcal{A}}, X.\tilde{\mu}_z^{\mathcal{A},+}), X.\tilde{\mu}_{tr}^{\mathcal{A}}, X.\tilde{\mu}_e^{\mathcal{A}})$, noted $(((\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A}}, (\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A},+}), (\tilde{\mathcal{E}}||X).\tilde{\mu}_{tr}^{\mathcal{A}}, (\tilde{\mathcal{E}}||X).\tilde{\mu}_e^{\mathcal{A}})$, is a strong continued PCA executions-matching from $\tilde{\mathcal{E}}||X$ to $\tilde{\mathcal{E}}||Z$.*

Proof. By definition of $\tilde{\mu}_z^{\mathcal{A},+}$ and $\tilde{\mu}_z^{\mathcal{A}}$, we have

- $\tilde{E}_{\tilde{\mathcal{E}}||X} = execs\text{-}without\text{-}creation(\tilde{\mathcal{E}}||X)(\mathcal{A})$
- $\tilde{E}^+_{\tilde{\mathcal{E}}||X} = execs\text{-}with\text{-}only\text{-}one\text{-}creation\text{-}at\text{-}last\text{-}action(\tilde{\mathcal{E}}||X)(\mathcal{A})$

- $\tilde{E}_X = execs\text{-}without\text{-}creation(X)(\mathcal{A})$
- $\tilde{E}^+_X = execs\text{-}with\text{-}only\text{-}one\text{-}creation\text{-}at\text{-}last\text{-}action(X)(\mathcal{A})$
- $\tilde{Q}_{\tilde{\mathcal{E}}||X} = reachable\text{-}by(\tilde{E}_{\tilde{\mathcal{E}}||X})$
- $\tilde{Q}^+_{\tilde{\mathcal{E}}||X} = reachable\text{-}by(\tilde{E}^+_{\tilde{\mathcal{E}}||X})$
- $\tilde{Q}_X = reachable\text{-}by(\tilde{E}_X)$
- $\tilde{Q}^+_X = reachable\text{-}by(\tilde{E}^+_X)$
- $dom((\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A},+}) = \tilde{Q}^+_{\tilde{\mathcal{E}}||X}$
- $dom((\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A}}) = \tilde{Q}_{\tilde{\mathcal{E}}||X}$
- $dom(X.\tilde{\mu}_z^{\mathcal{A},+}) = \tilde{Q}^+_X$
- $dom(X.\tilde{\mu}_z^{\mathcal{A}}) = \tilde{Q}_X$

This allow us to apply lemma 7.17 of "sufficient conditions to obtain range inclusion" to both $(\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A},+}$ and $(\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A}}$ which gives $range((\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A},+}) \subseteq states(\tilde{\mathcal{E}}||Z)$ and $range((\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A}}) \subseteq states(\tilde{\mathcal{E}}||Z)$ which allows us to apply lemma 7.24.

The lemma 7.34 implies that the resulting executions-matching is a strong one.

□

9.4 Composition and projection are commutative

This section aims to show in theorem 9.24 that operation of projection/deprivation and composition are commutative.

Theorem 9.24 ($(X||\mathcal{E}) \setminus \{\mathcal{A}\}$ and $(X \setminus \{\mathcal{A}\})||\mathcal{E}$ are semantically equivalent). *Let $\mathcal{A}$ be a PSIOA. Let X be a $\mathcal{A}$-fair PCA partially-compatible with $\mathcal{E}$ that never counts $\mathcal{A}$ in its constitution with both $X, \mathcal{E}$ and $X||\mathcal{E}$ configuration-conflict-free. The PCA $(X||\mathcal{E}) \setminus \{\mathcal{A}\}$ and $(X \setminus \{\mathcal{A}\})||\mathcal{E}$ are semantically equivalent.*

Proof. We note $W = X||\mathcal{E}$, $U = (X||\mathcal{E}) \setminus \{\mathcal{A}\}$, $V = (X \setminus \{\mathcal{A}\})||\mathcal{E}$, $\mu_s^{X,\mathcal{A}} = X.\mu_s^{\mathcal{A}}$, $\mu_s^{W,\mathcal{A}} = W.\mu_s^{\mathcal{A}}$. To stay simple, we note Id the identity function on any domain, that is we note Id for both $Id_{\mathcal{E}} : q_{\mathcal{E}} \in states(\mathcal{E}) \mapsto q_{\mathcal{E}}$ and $Id_U : q_U \in states(U) \mapsto q_U$.

The plan of the proof is the following one:

- We will construct two functions, $iso_{UV} : states(U) \to states(V)$ and $iso_{VU} : states(V) \to states(U)$, s. t. $iso_{UV}(q_U)$ is the unique element of $(\mu_s^{X,\mathcal{A}}, Id)((\mu_s^{W,\mathcal{A}})^{-1}(q_U))$ and $iso_{VU}((q_Y, q_{\mathcal{E}}))$ is the unique element of $\mu_s^{W,\mathcal{A}}((\mu_s^{X,\mathcal{A}}, Id)^{-1}((q_Y, q_{\mathcal{E}})))$.
- Then we will show that iso_{UV} and iso_{VU} are two bijections s. t. $iso_{VU} = iso_{UV}^{-1}$.
- Thereafter we will show that for every $(q_U, q_V), (q'_U, q'_V) \in (states(U) \times states(V))$, s. t. $q_V = iso_{UV}(q_U)$ and $q'_V = iso_{UV}(q'_U)$, then $q_U R_{strict} q_V$, $q'_U R_{strict} q'_V$ and for every $a \in \widehat{sig}(U)(q_U) = \widehat{sig}(V)(q_V)$, $\eta_{(U,q_U,a)}(q'_U) = \eta_{(V,q_V,a)}(q'_V)$.
- Finally, it will allow us to construct a strong complete bijective execution-matching induced by iso_{UV} and D_U (the set of discrete transitions of U) in bijection with a strong complete bijective execution-matching induced by iso_{VU} and D_V (the set of discrete transitions of V) .

First, we show that for every $q_W = (q_X, q_{\mathcal{E}}) \in reachable(W) \subset states(X) \times states(\mathcal{E})$, the state $q_V \triangleq (\mu_s^{X,\mathcal{A}}, Id)(q_W) = (\mu_s^{X,\mathcal{A}}(q_X), q_{\mathcal{E}})$ is an element of $reachable(V)$ (*). We proceed by induction. Basis: $(\mu_s^{X,\mathcal{A}}(\bar{q}_X), \bar{q}_{\mathcal{E}})$ is the initial state of V. Induction: Let $q_W \triangleq (q_X, q_{\mathcal{E}}), q'_W \triangleq (q'_X, q'_{\mathcal{E}}) \in reachable(W), q_V \in reachable(V), a \in \widehat{sig}(W)(q_W)$ s. t. $q'_W \in supp(\eta_{(W,q_W,a)})$, $q_V = (\mu_s^{X,\mathcal{A}}, Id)(q_W)$, and $q'_V = (\mu_s^{X,\mathcal{A}}, Id)(q'_W)$. There is two cases:

case 1) a is $\mathcal{A}$-exclusive in q_W. In this case $q_W R^{\setminus\{\mathcal{A}\}} q'_W$, which means $q'_V = q_V$ and ends the proof

case 2) $a \in \widehat{sig}(V)(q_V) \cap \widehat{sig}(W)(q_W)$

We need to show that $q'_V \in supp(\eta_{(V,q_V,a)})$. This is easy to show. Indeed, $q'_W \in supp(\eta_{(W,q_W,a)})$ means $(q'_X, q'_{\mathcal{E}}) \in supp(\eta_{(X,q_X,a)} \otimes \eta_{(\mathcal{E},q_{\mathcal{E}},a)})$ (with the convention $\eta_{(X,q_X,a)} = \delta_{q_X}$ if $a \notin \widehat{sig}(X)(q_X)$) and $\eta_{(\mathcal{E},q_{\mathcal{E}},a)} = \delta_{q_{\mathcal{E}}}$ if $a \notin \widehat{sig}(\mathcal{E})(q_{\mathcal{E}})$)) which means $q'_X \in supp(\eta_{(X,q_X,a)})$ and $q'_{\mathcal{E}} \in supp(\eta_{(\mathcal{E},q_{\mathcal{E}},a)})$. So $\mu_s^{X,\mathcal{A}}(q'_X) \in supp(\eta_{(Y,\mu_s^{X,\mathcal{A}}(q_X),a)})$ which means $(\mu_s^{X,\mathcal{A}}(q'_X), q'_{\mathcal{E}}) \in supp(\eta_{(Y,\mu_s^{X,\mathcal{A}}(q_X),a)} \otimes \eta_{(\mathcal{E},q_{\mathcal{E}},a)})$, that is $(\mu_s^{X,\mathcal{A}}(q'_X), q'_{\mathcal{E}}) \in supp(\eta_{((Y,\mathcal{E}),(\mu_s^{X,\mathcal{A}}(q_X),q_{\mathcal{E}}),a)})\eta_{(\mathcal{E},q_{\mathcal{E}},a)})$ and thus $q'_V \in supp(\eta_{(V,q_V,a)})$ so $q'_V \in reachable(V)$.

Second, we show that for every $q_V \triangleq (q_Y, q_{\mathcal{E}}) \in reachable(V)$, it exists $q_W \triangleq (q_X, q_{\mathcal{E}}) \in reachable(W)$ s. t. $q_V = (\mu_s^{X,\mathcal{A}}, Id)(q_W)$ (**). The reasoning is the same, we proceed by induction. The basis is performed with start state correspondance as before. Induction: Let $q_V \triangleq (q_Y, q_{\mathcal{E}}), q'_V \triangleq (q'_Y, q'_{\mathcal{E}}) \in reachable(V), q_W \in reachable(W), a \in \widehat{sig}(V)(q_V) \cap \widehat{sig}(W)(q_W)$ s. t. $q'_V \in supp(\eta_{(V,q_V,a)})$ with $q_V = (\mu_s^{X,\mathcal{A}}, Id)(q_W)$.

We need to show that it exists $q'_W \in supp(\eta_{(W,q_W,a)})$ s. t. $q'_V = (\mu_s^{X,\mathcal{A}}, Id)(q'_W)$. This is easy to show because of $\mu_d^{X,\mathcal{A}}$-correspondance. For every $q'_V \triangleq (q'_Y, q_{\mathcal{E}}) \in supp(\eta_{(V,(q_Y,q_{\mathcal{E}}),a)})$, $q'_Y \in supp(\eta_{(Y,q_Y,a)})$. Because of $\mu_d^{X,\mathcal{A}}$-correspondance, it exists $q'_X \in supp(\eta_{(X,q_X,a)})$ with $q'_Y = \mu_s^{X,\mathcal{A}}(q'_X)$, thus it exists $q'_W = (q'_X, q'_{\mathcal{E}}) \in supp(\eta_{(W,(q_X,q_{\mathcal{E}}),a)})$ s. t. $q'_V = (\mu_s^{X,\mathcal{A}}(q'_X), q'_{\mathcal{E}})$ which ends the proof of this second point.

Now we can construct iso_{UV} and iso_{VU}.

- iso_{UV}: for every $q_U \in states(U)$, $(\mu_s^{W,\mathcal{A}})^{-1}(q_U) \neq \emptyset$ by construction of U and for every $q_W \triangleq (q_X, q_{\mathcal{E}}), q'_W \triangleq (q'_X, q'_{\mathcal{E}}) \in (\mu_s^{W,\mathcal{A}})^{-1}(q_U)$, $q_W R_{strict}^{\setminus\{\mathcal{A}\}} q'_W$
  [...],
  which means for every $q_W \triangleq (q_X, q_{\mathcal{E}}), q'_W \triangleq (q'_X, q'_{\mathcal{E}}) \in (\mu_s^{W,\mathcal{A}})^{-1}(q_U)$, $(\mu_s^{X,\mathcal{A}}, Id)((q_X, q_{\mathcal{E}})) = (\mu_s^{X,\mathcal{A}}, Id)((q'_X, q'_{\mathcal{E}}))$ and so $(\mu_s^{X,\mathcal{A}}, Id)((\mu_s^{W,\mathcal{A}})^{-1}(q_U)) = \{q_V\}$ where $q_V \triangleq iso_{UV}(q_U) \in states(V)$ by (*).
- iso_{VU}: for every $q_V \triangleq (q_Y, q_{\mathcal{E}}) \in states(V)$, $(\mu_s^{X,\mathcal{A}}, Id)^{-1}(q_V) \neq \emptyset$ by (**). Furthermore for every $q_W \triangleq (q_X, q_{\mathcal{E}}), q'_W \triangleq (q'_X, q_{\mathcal{E}}) \in (\mu_s^{X,\mathcal{A}}, Id)^{-1}(q_V)$, $q_X R_{strict}^{\setminus\{\mathcal{A}\}} q'_X$, which means $q_W R_{strict}^{\setminus\{\mathcal{A}\}} q'_W$ and so $\mu_s^{W,\mathcal{A}}((\mu_s^{X,\mathcal{A}}, Id)^{-1}(q_V)) = \{q_U\}$ where $q_U \triangleq iso_{VU}(q_V) \in states(U)$

Now we can show that iso_{UV} is a bijection with $iso_{VU} = iso_{UV}^{-1}$.

- surjectivity of iso_{UV}: Let $q_V = (q_Y, q_{\mathcal{E}}) \in reachable(V)$, we will show that it exists $q_U \in reachable(U)$ s. t. $iso_{UV}(q_U) = q_V$. Indeed, we already know that (*) it exists $q_W = (q_X, q_{\mathcal{E}}) \in (\mu_s^{X,\mathcal{A}}, Id)^{-1}(q_V) \cap reachable(W)$. Let $q_U = \mu_s^{W,\mathcal{A}}(q_W)$. By construction of U, we have $q_U \in reachable(U)$ and $q_W \in (\mu_s^{W,\mathcal{A}})^{-1}(q_U)$ and $(\mu_s^{X,\mathcal{A}}, Id)(q_W) = q_V$ which means $iso_{UV}(q_U) = q_V$ and ends this item.
- injectivity of iso_{UV}: Let $q_V \in reachable(V)$, Let $q_U, q'_U \in reachable(U)$ s. t. $iso_{UV}(q_U) = iso_{UV}(q'_U)$ then $q_U = q'_U$. Again for every $q_W, q'_W \in (\mu_s^{X,\mathcal{A}}, Id)^{-1}(q_V)$, $q_W R_{strict}^{\setminus\mathcal{A}} q'_W$ and so $\mu_s^{W,\mathcal{A}}(q_W) = \mu_s^{W,\mathcal{A}}(q'_W)$. But for every $q_U, q'_U \in iso_{UV}^{-1}(q_V)$, $q_U, q'_U \in \mu_s^{W,\mathcal{A}}(\mu_s^{X,\mathcal{A}}, Id)^{-1}(q_V)$ which means $q_U = q'_U$.

Let (i) $q_V = iso_{UV}(q_U)$ or (ii) $q_U = iso_{UV}(q_V)$ we will show that in both (i) and (ii) $q_V R_{strict} q_U$. By definition, $\{q_V\} = (\mu_s^{X,\mathcal{A}}, Id)(\mu_s^{W,\mathcal{A}})^{-1}(q_U))$.

In case (i) we note q_W an arbitrary element of $(\mu_s^{W,\mathcal{A}})^{-1}(q_U) \neq \emptyset$, while in case (ii) we note q_W an arbitrary element of $(\mu_s^{X,\mathcal{A}}, Id)^{-1}(q_V) \neq \emptyset$. In both cases, we have 1a) $config(W)(q_W) \setminus \{\mathcal{A}\} = config(U)(q_U)$ and 1b) $config(W)(q_W) \setminus \{\mathcal{A}\} = config(V)(q_V)$, which means 1c) $config(U)(q_U) = config(V)(q_V)$. Then we have 2a) $hidden\text{-}actions(W)(q_W) \setminus pot\text{-}out(W)(q_W)(\mathcal{A}) = hidden\text{-}actions(U)(q_U) \setminus pot\text{-}out(W)(q_W)(\mathcal{A}) = hidden\text{-}actions(U)(q_U)$ and 2b) $hidden\text{-}actions(W)(q_W) \setminus pot\text{-}out(W)(q_W)(\mathcal{A}) = hidden\text{-}actions(V)(q_V) \setminus pot\text{-}out(W)(q_W)(\mathcal{A}) = hidden\text{-}actions(V)(q_V)$,

which means 2c) $\mathit{hidden\text{-}actions}(U)(q_U) = \mathit{hidden\text{-}actions}(V)(q_V)$. Thereafter we have 3a) for every action $a \in \widehat{sig}(W)(q_W) \cap \widehat{sig}(U)(q_U), \mathit{created}(W)(q_W)(a) \setminus \{\mathcal{A}\} = \mathit{created}(U)(q_U)(a) \setminus \{\mathcal{A}\} = \mathit{created}(U)(q_U)(a)$ and 3b) for every action $a \in \widehat{sig}(W)(q_W) \cap \widehat{sig}(V)(q_V), \mathit{created}(W)(q_W)(a) \setminus \{\mathcal{A}\} = \mathit{created}(V)(q_V)(a) \setminus \{\mathcal{A}\} = \mathit{created}(V)(q_V)(a)$ which means 3c) for every action $a \in \widehat{sig}(U)(q_U) = \widehat{sig}(V)(q_V), \mathit{created}(U)(q_U)(a) = \mathit{created}(V)(q_V)(a)$. The conjonction of 3a), 3b) and 3c) lead us to $q_V R_{strict} q_U$.

Now we can show that iso_{UV} is the reverse function of iso_{VU}: Let $(q_U, q_V) \in \mathit{reachable}(U) \times \mathit{reachable}(V)$ s. t. $q_V = iso_{UV}(q_U)$. We need to show that $iso_{VU}(q_V) = q_U$. The point is that it exists a unique $q'_U \triangleq iso_{VU}(q_V)$ and we have $q_V R_{strict} q_U$ and $q_V R_{strict} q'_U$ which means $q_U R_{strict} q'_U$ and so $q_U = q'_U$ by assumption of configuration-conflict-free PCA. Hence $iso_{UV} = iso_{VU}^{-1}$.

The last point is to show that that for every $(q_U, q_V), (q'_U, q'_V) \in \mathit{reachable}(U) \times \mathit{reachable}(V)$, s. t. $q_V = iso_{UV}(q_U)$ and $q'_V = iso_{UV}(q'_U)$, then $q_U R_{strict} q_V$, $q'_U R_{strict} q'_V$ and for every $a \in \widehat{sig}(U)(q_U) = \widehat{sig}(V)(q_V)$, $\eta_{(U,q_U,a)}(q'_U) = \eta_{(V,q_V,a)}(q'_V)$.

For every $a \in \widehat{sig}(U)(q_U) = \widehat{sig}(V)(q_V)$ we have a unique η s. t. $C \xRightarrow{a}_{\varphi} \eta$ with $C = \mathit{config}(U)(q_U) = \mathit{config}(V)(q_V)$ and $\varphi = \mathit{created}(U)(q_U)(a) = \mathit{created}(V)(q_V)(a)$. Hence for every configuration $C' \in \mathit{supp}(\eta)$, it exists a unique pair $(q'_U, q'_V) \in \mathit{reachable}(U) \times \mathit{reachable}(V)$ s. t. $C' = \mathit{config}(U)(q'_U) = \mathit{config}(V)(q'_V)$. Hence $iso_{UV}(q'_U) = q'_V$ and furthermore $\eta_{(U,q_U,a)}(q'_U) = \eta_{(V,q_V,a)}(q'_V) = \eta(C)$.

Everything is ready to construct the PCA-execution-matching, which is (j) the PCA-execution-matching induced by iso_{UV} and D_U (the set of discrete transition of U) and (jj) the PCA-execution-matching induced by iso_{VU} and D_V (the set of discrete transition of V)

□

10 PCA CORRESPONDING W.R.T. PSIOA $\mathcal{A}$, $\mathcal{B}$

In the previous section we have shown that $X_{\mathcal{A}}||\mathcal{E}$ and $\tilde{\mathcal{A}}^{sw}||(X_{\mathcal{A}} \setminus \{\mathcal{A}\}||\mathcal{E})$ are linked by a strong PCA executions-matching as long as $\mathcal{A}$ is not re-created by $X_{\mathcal{A}}$. This also means that the probability distribution of $X_{\mathcal{A}}||\mathcal{E}$ is preserved by $\tilde{\mathcal{A}}^{sw}||(X \setminus \{\mathcal{A}\}||\mathcal{E})$, as long as $\mathcal{A}$ is not re-created by $X_{\mathcal{A}}$. We can have the same reasoning to obtain a strong PCA executions-matching from $X_{\mathcal{B}}||\mathcal{E}$ and $\tilde{\mathcal{B}}^{sw}||(X_{\mathcal{B}} \setminus \{\mathcal{B}\}||\mathcal{E})$.

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 $X_{\mathcal{B}}$. Hence, we recall the definitions of section 6. Then, we show that under slight assumptions, $X_{\mathcal{A}} \setminus \{\mathcal{A}\}$ and $X_{\mathcal{B}} \setminus \{\mathcal{B}\}$ are semantically equivalent (see theorem 10.13).

Combined with the result of previous section we will realise that we can obtain a strong PCA executions-matching from (*) $X_{\mathcal{A}}||\mathcal{E}$ to $\tilde{\mathcal{A}}^{sw}||(Y||\mathcal{E})$ and (**) from $X_{\mathcal{B}}||\mathcal{E}$ to $\tilde{\mathcal{B}}^{sw}||(Y||\mathcal{E})$ where Y is semantically equivalent to both $X_{\mathcal{B}} \setminus \{\mathcal{B}\}$ and $X_{\mathcal{A}} \setminus \{\mathcal{A}\}$. Hence if $\mathcal{E}' = \mathcal{E}||Y$ cannot distinguish $\tilde{\mathcal{A}}^{sw}$ from $\tilde{\mathcal{B}}^{sw}$, we will be able to show that $\mathcal{E}$ cannot distinguish $X_{\mathcal{A}}$ from $X_{\mathcal{B}}$ which will be the subject of sections 11 to finally prove the monotonicity of *print*-implementation.

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

Definition 10.1 ($\triangleleft_{\mathcal{A}\mathcal{B}}$-corresponding configurations). (see figure 30) Let $\Phi \subseteq \mathit{Autids}$, 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 $C \triangleleft_{\mathcal{A}\mathcal{B}} D$ iff (1) $\mathit{auts}(D) = \mathit{auts}(C)[\mathcal{B}/\mathcal{A}]$, (2) for every $\mathcal{A}' \notin \mathit{auts}(C) \setminus \{\mathcal{A}\} : \mathit{map}(D)(\mathcal{A}') = \mathit{map}(C)(\mathcal{A}')$, and (3) $\mathit{ext}(\mathcal{A})(s) = \mathit{ext}(\mathcal{B})(t)$ where $s = \mathit{map}(C)(\mathcal{A}), t = \mathit{map}(D)(\mathcal{B})$. That is, in $\triangleleft_{\mathcal{A}\mathcal{B}}$-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 $\mathcal{A} \notin \Psi$.

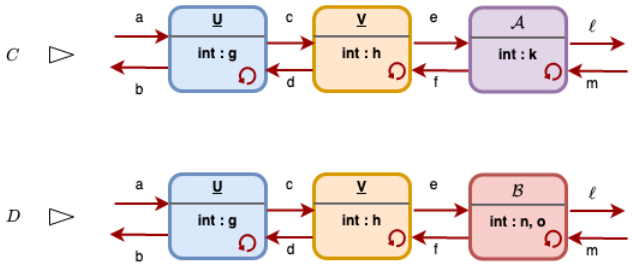


Fig. 30. $\lhd_{\mathcal{AB}}$ corresponding-configuration

Next lemma states that $\lhd_{AB}$-corresponding configurations have the same external signature, which is quite intuitive when we see the figure 30.

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. We write the proof here to be complete:

If $\mathcal{A} \notin C$ then $C = D$ by definition , and we are done. Now suppose that $\mathcal{A} \in C$, so that $C = (\mathbf{A} \cup \{\mathcal{A}\}, \mathbf{S})$ for some set $\mathbf{A}$ of PSIOA identifiers s. t. $\mathcal{A} \notin \mathbf{A}$, and let $s = \mathbf{S}(\mathcal{A})$. Then, by definition 5.3 of attributes of configuration, $out(C) = (\bigcup_{\mathcal{A}_i \in \mathbf{A}} out(\mathcal{A}_i)(\mathbf{S}(\mathcal{A}_i))) \cup out(\mathcal{A})(s)$. From $C \lhd_{\mathcal{AB}} D$ and definition , we have $D = (\mathbf{A} \cup \{\mathcal{B}\}, \mathbf{S}')$, where $\mathbf{S}'$ agrees with $\mathbf{S}$ on all $\mathcal{A}_i \in \mathbf{A}$, and $t = \mathbf{S}'(\mathcal{B})$ such that $ext(\mathcal{A})(s) = ext(\mathcal{B})(t)$. Hence $out(\mathcal{A})(s) = out(\mathcal{B})(t)$ and $in(\mathcal{A})(s) = in(\mathcal{B})(t)$. By definition 5.3 of configuration attributes, $out(D) = (\bigcup_{\mathcal{A}_i \in \mathbf{A}} out(\mathcal{A}_i)(\mathbf{S}'(\mathcal{A}_i))) \cup out(\mathcal{B})(t)$. Finally, $out(C) = out(D)$ since $\mathbf{S}'$ agrees with $\mathbf{S}$ on all $\mathcal{A} \in \mathbf{A}$ and $out(\mathcal{A})(s) = out(\mathcal{B})(t)$. We establish $in(C) = in(D)$ in the same manner, and omit the repetitive details. Hence $ext(C) = ext(D)$. □

Remark 3. *It is possible to have two configurations C, D s. t. $C \lhd_{\mathcal{A}\mathcal{A}} D$. 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 potentially have $int(\mathcal{A})(s) \neq int(\mathcal{A})(t)$.*

The next lemma states that $\lhd_{AB}$-corresponding configurations are equals if we omit the automata $\mathcal{A}$ and $\mathcal{B}$.

Lemma 10.2 (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 $(x_a, x_b) \in states(X_{\mathcal{A}}) \times states(X_{\mathcal{B}})$ s. t. $config(X_{\mathcal{A}})(x_a) \lhd_{AB} config(X_{\mathcal{B}})(x_b)$. Let $y_a = X_{\mathcal{A}}.\mu_s^{\mathcal{A}}(x_a)$, $y_b = X_{\mathcal{A}}.\mu_s^{\mathcal{A}}(x_b)$*

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 $\lhd_{AB}$). □

same comportment of two PCA modulo $\mathcal{A}, \mathcal{B}$. In this paragraph we formalise the fact that two PCA have the same comportment, excepting for $\mathcal{B}$ that supplants $\mathcal{A}$.

First, we formalise the fact that two PCA create some PSIOA in the same manner, excepting for $\mathcal{B}$ that supplants $\mathcal{A}$. Here again, this definition comes from [1].

Definition 10.3 (Creation corresponding configuration automata). Let X, Y be configuration automata and $\mathcal{A}, \mathcal{B}$ be PSIOA. 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 $\forall a \in \widehat{sig}(X)(x) \cap \widehat{sig}(Y)(y) : created(Y)(y)(a) = created(X)(x)(a)[\mathcal{B}/\mathcal{A}]$.

Naturally $[\mathcal{B}/\mathcal{A}]$-corresponding sets of created automata are deprived of $\mathcal{A}$ and $\mathcal{B}$ respectively, they becomes equal, which is formalised in next lemma.

Lemma 10.4 (Same creation after projection). *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 $\mathcal{A}$ ($\mathcal{B} \notin UA(X_{\mathcal{A}})$ and $\mathcal{A} \notin UA(X_{\mathcal{B}})$). 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 $y_a = X_{\mathcal{A}}.\mu_s^{\mathcal{A}}(x_a)$, $y_b = X_{\mathcal{B}}.\mu_s^{\mathcal{B}}(x_b)$*
Then $created(Y_{\mathcal{B}})(x_b)(act) = created(Y_{\mathcal{A}})(x_a)(act)$

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

Second, we formalise the fact that two PCA hide their actions in the same manner. The definition is strongly inspired by [1].

Definition 10.5 (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 $hidden\text{-}actions(Y)(y) = hidden\text{-}actions(X)(x)$.

Naturally if hidden actions of $\lhd_{\mathcal{A}\mathcal{B}}$-corresponding states are equal, it remains true after respective deprivation of $\mathcal{A}$ and $\mathcal{B}$ which is formalised in next lemma.

Lemma 10.6 (Same hidden-actions after projection). *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 $\mathcal{A}$ ($\mathcal{B} \notin UA(X_{\mathcal{A}})$ and $\mathcal{A} \notin UA(X_{\mathcal{B}})$). Let $Y_{\mathcal{A}} = X_{\mathcal{A}} \setminus \{\mathcal{A}\}$, $Y_{\mathcal{B}} = X_{\mathcal{B}} \setminus \{\mathcal{B}\}$. Let $(x_a, x_b) \in states(X_{\mathcal{A}}) \times states(X_{\mathcal{B}})$, $y_a = X_{\mathcal{A}}.\mu_s^{\mathcal{A}}(x_a)$, $y_b = X_{\mathcal{B}}.\mu_s^{\mathcal{B}}(x_b)$ s. t.*

- $x_a R_{conf}^{\setminus\{\mathcal{A}\}} x_b$, *i. e.* $y_a R_{conf} y_b$
- $hidden\text{-}actions(X_{\mathcal{B}})(x_b) = hidden\text{-}actions(X_{\mathcal{A}})(x_a)$

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

Proof. We note $C_{X_\mathcal{A}} = config(X_\mathcal{A})(x_a)$, $C_{X_\mathcal{B}} = config(X_\mathcal{B})(x_b)$, $C_{Y_\mathcal{A}} = config(Y_\mathcal{A})(y_a)$, $C_{Y_\mathcal{B}} = config(Y_\mathcal{B})(y_b)$. By assumption, $C_{X_\mathcal{A}} \setminus \{\mathcal{A}\} = C_{Y_\mathcal{A}} = C_{Y_\mathcal{B}} = C_{X_\mathcal{B}} \setminus \{\mathcal{B}\}$.

We note $h_{X_\mathcal{A}} = hidden\text{-}actions(X_\mathcal{A})(x_a)$, $h_{X_\mathcal{B}} = hidden\text{-}actions(X_\mathcal{B})(x_b)$, $h_{Y_\mathcal{A}} = hidden\text{-}actions(Y_\mathcal{A})(y_a)$, $h_{Y_\mathcal{B}} = hidden\text{-}actions(Y_\mathcal{B})(y_b)$. By assumption, $h_{X_\mathcal{A}} = h_{X_\mathcal{B}}$, while by construction, $h_{Y_\mathcal{A}} = h_{X_\mathcal{A}} \setminus pot\text{-}out(X_\mathcal{A})(\mathcal{A})$ and $h_{Y_\mathcal{B}} = h_{X_\mathcal{B}} \setminus pot\text{-}out(X_\mathcal{B})(\mathcal{B})$.

Case 1: $pot\text{-}out(X_\mathcal{A})(\mathcal{A})(x_a) = pot\text{-}out(X_\mathcal{B})(\mathcal{B})(x_b)$, the result is immediate, Case 2: $pot\text{-}out(X_\mathcal{A})(\mathcal{A})(x_a) \cap h_{X_\mathcal{A}} = pot\text{-}out(X_\mathcal{B})(\mathcal{B})(x_b) \cap h_{X_\mathcal{B}} = \emptyset$, the result is immediate.

Case 3: Without loss of generality, we assume $\underline{act} = pot\text{-}out(X_\mathcal{A})(\mathcal{A})(x_a) \cap h_{X_\mathcal{A}} \neq \emptyset$. For every $C \in auts(C_{Y_\mathcal{B}})$, $C \in auts(C_{Y_\mathcal{A}})$ since $C_{Y_\mathcal{A}} = C_{Y_\mathcal{B}}$ and $C \in auts(C_{X_\mathcal{A}})$ since $C_{Y_\mathcal{A}} = C_{X_\mathcal{A}} \setminus \{\mathcal{A}\}$. By compatibility of $C_{X_\mathcal{A}}$, $pot\text{-}out(X_\mathcal{A})(\mathcal{A})(x_a) \cap pot\text{-}out(X_\mathcal{A})(C)(x_a) = \emptyset$.

Case 3a) $\mathcal{B} \notin auts(C_{X_\mathcal{B}})$, which means both i) $\underline{act} \subset h_{X_\mathcal{B}}$, ii) $\underline{act} \cap out(C_{X_\mathcal{B}}) = \emptyset$ and iii) $h_{X_\mathcal{B}} \subset out(C_{X_\mathcal{B}})$ which is impossible. Thus we only consider

Case 3b) $\mathcal{B} \in auts(C_{X_\mathcal{B}})$. Since j) for every $C \in auts(C_{Y_\mathcal{B}})$, $pot\text{-}out(X_\mathcal{A})(\mathcal{A})(x_a) \cap pot\text{-}out(X_\mathcal{A})(C)(x_a) = \emptyset$ and jj) $h_{X_\mathcal{B}} \subset out(C_{X_\mathcal{B}})$, we have $\underline{act} \subset pot\text{-}out(X_\mathcal{B})(\mathcal{B})(x_b)$.

For symmetrical reason, we have both $pot\text{-}out(X_\mathcal{A})(\mathcal{A})(x_a) \cap h_{X_\mathcal{A}} \subset pot\text{-}out(X_\mathcal{B})(\mathcal{B})(x_b)$ and $pot\text{-}out(X_\mathcal{B})(\mathcal{B})(x_b) \cap h_{X_\mathcal{B}} \subset pot\text{-}out(X_\mathcal{A})(\mathcal{A})(x_A)$, which means $h_{X_\mathcal{A}} \setminus pot\text{-}out(X_\mathcal{B})(\mathcal{B})(x_b) = h_{X_\mathcal{B}} \setminus pot\text{-}out(X_\mathcal{B})(\mathcal{B})(x_b)$ and ends the proof

□

Now we are ready to define corresponding PCA w. r. t. PSIOA $\mathcal{A}$, $\mathcal{B}$, that is two PCA $X_\mathcal{A}$ and $X_\mathcal{B}$ that differ only on the fact that B supplants $\mathcal{A}$ in $X_\mathcal{B}$. Some additional assumptions are added to ensure monotonicity later. This definition is still inspired by definitions of [1].

Definition 10.7 (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:

- $config(X_\mathcal{A})(\bar{q}_{X_\mathcal{A}}) \lhd_{AB} config(X_\mathcal{B})(\bar{q}_{X_\mathcal{B}})$.
- $X_\mathcal{A}$ never contains $\mathcal{B}$ ($\mathcal{B} \notin UA(X_\mathcal{A})$), while $X_\mathcal{B}$ never contains $\mathcal{A}$ ($\mathcal{A} \notin UA(X_\mathcal{B})$).
- $X_\mathcal{A}, X_\mathcal{B}$ 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 exclusive creation from $\mathcal{A}$ and $\mathcal{B}$)
  - $\forall q_{X_\mathcal{A}} \in states(X_\mathcal{A})$, for every action act $\mathcal{A}$-exclusive, $created(X_\mathcal{A})(q_{X_\mathcal{A}})(act) = \emptyset$ and similarly
  - $\forall q_{X_\mathcal{B}} \in states(X_\mathcal{B})$, for every action act' $\mathcal{B}$-exclusive, $created(X_\mathcal{B})(q_{X_\mathcal{B}})(act') = \emptyset$

equivalent transitions to obtain semantic equivalence after projection. In this last paragraph of the section, we show that if two PCA $X_\mathcal{A}$ $X_\mathcal{B}$ are corresponding w. r. t. $\mathcal{A}$ and $\mathcal{B}$, then there respective projection $Y_\mathcal{A} = X_\mathcal{A} \setminus \{\mathcal{A}\}$ and $Y_\mathcal{B} = X_\mathcal{B} \setminus \{\mathcal{B}\}$ are semantically equivalents. To do so, we use notions of equivalent transitions. the idea is to recursively show that any corresponding executions of $Y_\mathcal{A}$ and $Y_\mathcal{B}$ lead to strictly equivalent transitions to finally build the complete bijective PCA executions-matching from $Y_\mathcal{A}$ to $Y_\mathcal{B}$.

We start by defining equivalent transitions.

Definition 10.8 (configuration-equivalence and strict-equivalence between two distributions). Let K, K' be PCA and $(\eta, \eta') \in Disc(states(K)) \times Disc(states(K'))$.

- We say that η and η' are *config-equivalent*, noted $\eta \xleftrightarrow[conf]{f} \eta'$, if it exists $f : states(K) \longrightarrow states(K')$ s. t. $\eta \xleftrightarrow{f} \eta'$ with $\forall q'' \in supp(\eta), q'' R_{conf} f(q'')$.
- If additionally, $\forall q'' \in supp(\eta), q'' R_{strict} f(q'')$, then we say that η and η' are *strictly-equivalent*, noted $\eta \xleftrightarrow[strict]{f} \eta'$.

Basically, equivalent transitions are transitions where the states with non-zero probability to be reached are mapped by a bijective function that preserves i) measure of probability and ii) configuration. A stricter version preserves also iii) future created automata and hidden-actions.

The next lemma states that if we take two corresponding transitions from strict equivalent states, then we obtain configuration equivalent transitions.

Lemma 10.9. *(strictly-equivalent states implies config-equivalent transition) Let K, K' be PCA and $(q, q') \in states(K) \times states(K')$ strictly-equivalent, i. e. $qR_{strict}q'$. Let $a \in \widehat{sig}(K)(q) = \widehat{sig}(K')(q')$ and $((q, a, \eta_{(K,q,a)}), (q', a, \eta_{(K',q',a)})) \in dtrans(K) \times dtrans(K')$. Then $\eta_{(K,q,a)}$ and $\eta_{(K',q',a)}$ are config-equivalent, i. e. $\exists f : states(K) \longrightarrow states(K')$ s. t. $\eta \xleftrightarrow[conf]{f} \eta'$.*

Proof. This is the direct consequence of constraint 2 and 3 of definition 5.14 of PCA. We note $C = config(K)(q) = config(K')(q')$ and $\varphi = created(K)(q)(a) = created(K')(q')(a)$. By constraint 2, applied to K, it exists η s. t. $\eta_{(K,q,a)} \xleftrightarrow{f^K} \eta$ with $f^K = config(K)$ and $config(K)(q) \overset{a}{\Longrightarrow}_{created(K)(q)(a)} \eta$ By constraint 2, applied to K', it exists η' s. t. $\eta_{(K',q',a)} \xleftrightarrow{f^{K'}} \eta'$ with $f^{K'} = config(K')$ and $config(K')(q') \overset{a}{\Longrightarrow}_{created(K')(q')(a)} \eta'$.

Since $qR_{strict}q'$, $C \triangleq config(K)(q) = config(K')(q')$ and $\varphi \triangleq created(K)(q)(a) = created(K')(q')(a)$.

Hence $C \overset{a}{\Longrightarrow}_{\varphi} \eta$ and $C \overset{a}{\Longrightarrow}_{\varphi} \eta'$ which means $\eta = \eta'$.

So $\eta_{(K,q,a)} \xleftrightarrow{f} \eta_{(K',q',a)}$ with $\tilde{f} = (\tilde{f^{K'}})^{-1} \circ \tilde{f^K}$ where $\tilde{f}$ (resp. $\tilde{f^{K'}}$, resp. $\tilde{f^K}$) is the restriction of f (resp. $f^{K'}$, resp. f^K) on $supp(\eta_{(K,q,a)})$ (resp. $supp(\eta_{(K',q',a)})$, resp. $supp(\eta_{(K,q,a)})$).

Thus, for every $(\tilde{q}, \tilde{q}') \in supp(\eta_{(K,q,a)}) \times supp(\eta_{(K',q',a)})$ s. t. $\tilde{q}' = f(\tilde{q})$, $f^K(\tilde{q}) = f^{K'}(\tilde{q}')$, that is $config(K)(\tilde{q}) = config(K')(\tilde{q}')$, i. e. $\tilde{q} R_{conf} \tilde{q}'$.

Hence $\eta_{(K,q,a)} \xleftrightarrow[conf]{f} \eta_{(K',q',a)}$ which ends the proof.

□

Now we start a sequence of lemma (from lemma 10.10 to lemma 10.12) to finally show in theorem 10.13 that if $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ are corresponding w. r. t. $\mathcal{A}, \mathcal{B}$ then $X_{\mathcal{A}} \setminus \{\mathcal{A}\}$ and $X_{\mathcal{B}} \setminus \{\mathcal{B}\}$ are semantically-equivalent.

The next lemma shows that we can always construct an execution $\tilde{\alpha}_X \in Execs(X)$ from an execution $\alpha_Y \in Execs(Y)$ with $Y = X \setminus \{\mathcal{A}\}$ that preserves the trace.

Lemma 10.10 ($Execs(X \setminus \{\mathcal{A}\})$ can be obtained by $Execs(X)$). *Let $\mathcal{A} \in Autids$, X a $\mathcal{A}$-fair PCA, $Y = X \setminus \{\mathcal{A}\}$.*

Let $\alpha_Y = q_Y^0, a^1, q_Y^1, ..., q_Y^n \in Execs(Y)$. Then it exists, $\tilde{\alpha}_X = \tilde{q}_X^0, a^1, \tilde{q}_X^1, ..., \tilde{q}_X^n \in Execs(X)$ s. t. $\forall i \in [0, n], q_Y^i = \mu_s^{\mathcal{A}}(\tilde{q}_X^i)$.

Proof. By induction on the size $s = |\alpha_Y^s|$ of prefix $\alpha_Y^s = q_Y^0, a^1, q_Y^1, ..., q_Y^s$.

Basis ($|\alpha_Y^s| = 0$): By definition 8.11, $\bar{q}_Y = X.\mu_s^{\mathcal{A}}(\bar{q}_X)$

Induction: let assume the proposition is true for prefix $\alpha_Y^s = q_Y^0, a^1, q_Y^1, ..., q_Y^s$ with $s < |\alpha_Y|$. We will show it is true for α_Y^{s+1}. We have $q_Y^s = X.\mu_s^{\mathcal{A}}(q_X^s)$. By construction of $dtrans(Y)$ provided by definition 8.11, it exists $\eta_{(X,q_X^s,a^{s+1})} \in$

$dtrans(X)$ s. t. $X.\mu_d^{\mathcal{A}}(\eta_{(X,q_X^s,a^{s+1})}) = \eta_{(Y,q_Y^s,a^{s+1})}$. By $X.\mu_d^{\mathcal{A}}$-correspondence of definition 8.11, $\eta_{(Y,q_Y^s,a^{s+1})}(q_Y^{s+1}) = \sum_{q'_X \in states(X),\mu_s(q'_X)=q_Y^{s+1}} \eta_{(X,q_X^s,a^{s+1})}(q'_X)$. By definition of an execution, $q_Y^{s+1} \in supp(\eta_{(Y,q_Y^s,a^{s+1})})$, which means it exists $q_X^{s+1} \in states(X)$ s. t. 1) $\mu_s^{\mathcal{A}}(q_X^{s+1}) = q_Y^{s+1}$ and 2) $q_X^{s+1} \in supp(\eta_{(X,q_X^s,a^{s+1})})$. Thus, it exist $\tilde{\alpha}_X^{s+1} = \tilde{q}_X^0, a^1, \tilde{q}_X^1, ..., \tilde{q}_X^{s+1} \in Execs(X)$ s. t. $\forall i \in [0, s+1], q_Y^i = \mu_s^{\mathcal{A}}(\tilde{q}_X^i)$, which ends the induction and so the proof. □

The next lemma states that, after projection, two configuration-equivalent states obtain via executions with the same trace are strictly equivalent.

Lemma 10.11 (After projection, configuration-equivalence obtain after same trace implies strict equivalence). *Let $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ be two PCA corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$. Let $Y_{\mathcal{A}} = X_{\mathcal{A}} \setminus \{\mathcal{A}\}$ and $Y_{\mathcal{B}} = X_{\mathcal{B}} \setminus \{\mathcal{B}\}$. Let $(\alpha_{Y_{\mathcal{A}}}, \pi_{Y_{\mathcal{B}}}) \in Execs(Y_{\mathcal{A}}) \times Execs(Y_{\mathcal{B}})$ with $lstate(\alpha_{Y_{\mathcal{A}}}) = q_{Y_{\mathcal{A}}}$ and $lstate(\pi_{Y_{\mathcal{B}}}) = q_{Y_{\mathcal{B}}}$. If*

- *$q_{Y_{\mathcal{A}}} R_{conf} q_{Y_{\mathcal{B}}}$ and*
- *$trace(\alpha_{Y_{\mathcal{A}}}) = trace(\pi_{Y_{\mathcal{B}}}) = \beta$,*

then $q_{Y_{\mathcal{A}}} R_{strict} q_{Y_{\mathcal{B}}}$

Proof. By lemma 10.10, it exists $(\tilde{\alpha}_{X_{\mathcal{A}}}, \tilde{\pi}_{X_{\mathcal{B}}}) \in Execs(X_{\mathcal{A}}) \times Execs(X_{\mathcal{B}})$ s. t. (i) $trace(\tilde{\alpha}_{X_{\mathcal{A}}}) = trace(\alpha_{Y_{\mathcal{A}}}) = trace(\pi_{Y_{\mathcal{B}}}) = trace(\tilde{\pi}_{X_{\mathcal{B}}})$ and (ii) $q_{Y_{\mathcal{A}}} = X_{\mathcal{A}}.\mu_s^{\mathcal{A}}(\tilde{q}_{X_{\mathcal{A}}})$ and $q_{Y_{\mathcal{B}}} = X_{\mathcal{B}}.\mu_s^{\mathcal{B}}(\tilde{q}_{X_{\mathcal{B}}})$ where $\tilde{q}_{X_{\mathcal{B}}} = lstate(\tilde{\pi}_{X_{\mathcal{B}}})$ and $\tilde{q}_{X_{\mathcal{A}}} = lstate(\tilde{\alpha}_{X_{\mathcal{A}}})$.

Since $trace(\tilde{\alpha}_{X_{\mathcal{A}}}) = trace(\tilde{\pi}_{X_{\mathcal{B}}})$, we have j) $hidden\text{-}actions(X_{\mathcal{A}})(\tilde{q}_{X_{\mathcal{A}}}) = hidden\text{-}actions(X_{\mathcal{B}})(\tilde{q}_{X_{\mathcal{B}}})$ by hiding-correspondence of definition 6.3 and jj) $\forall a \in \widehat{sig}(X_{\mathcal{A}})(\tilde{q}_{X_{\mathcal{A}}}) \cap \widehat{sig}(X_{\mathcal{B}})(\tilde{q}_{X_{\mathcal{B}}}), created(X_{\mathcal{A}})(\tilde{q}_{X_{\mathcal{A}}})(a) = created(X_{\mathcal{B}})(\tilde{q}_{X_{\mathcal{B}}})(a)$.

By lemma 10.6 we have (*) $hidden\text{-}actions(Y_{\mathcal{A}})(\tilde{q}_{Y_{\mathcal{A}}}) = hidden\text{-}actions(Y_{\mathcal{B}})(\tilde{q}_{Y_{\mathcal{B}}})$, and by lemma 10.4 we have (**) $\forall a \in \widehat{sig}(Y_{\mathcal{A}})(q_{Y_{\mathcal{A}}}) = \widehat{sig}(Y_{\mathcal{B}})(q_{Y_{\mathcal{B}}})$.

If we combine the definition $q_{Y_{\mathcal{A}}} R_{conf} q_{Y_{\mathcal{B}}}$ with (*) and (**), we obtain $q_{Y_{\mathcal{A}}} R_{strict} q_{Y_{\mathcal{B}}}$, which ends the proof.

□

Finally, the next lemma states that, after projection, two configuration-equivalent states obtain via executions with the same trace lead necessarily to strictly equivalent transitions.

Lemma 10.12 (After projection, configuration-equivalence obtain after same trace implies strict equivalent transitions). *Let $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ be two PCA corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$. Let $Y_{\mathcal{A}} = X_{\mathcal{A}} \setminus \{\mathcal{A}\}$ and $Y_{\mathcal{B}} = X_{\mathcal{B}} \setminus \{\mathcal{B}\}$. Let $(\alpha_{Y_{\mathcal{A}}}, \pi_{Y_{\mathcal{B}}}) \in Execs(Y_{\mathcal{A}}) \times Execs(Y_{\mathcal{B}})$ with $lstate(\alpha_{Y_{\mathcal{A}}}) = q_{Y_{\mathcal{A}}}$ and $lstate(\pi_{Y_{\mathcal{B}}}) = q_{Y_{\mathcal{B}}}$. If*

- *$q_{Y_{\mathcal{A}}} R_{conf} q_{Y_{\mathcal{B}}}$ and*
- *$trace(\alpha_{Y_{\mathcal{A}}}) = trace(\pi_{Y_{\mathcal{B}}}) = \beta$,*

then for every $a \in \widehat{sig}(Y_{\mathcal{A}})(q_{Y_{\mathcal{A}}}) = \widehat{sig}(Y_{\mathcal{B}})(q_{Y_{\mathcal{B}}}), \eta_{(Y_{\mathcal{A}},q_{Y_{\mathcal{A}}},a)}$ and $\eta_{(Y_{\mathcal{B}},q_{Y_{\mathcal{B}}},a)}$ are strictly equivalent, i. e. $\exists f : states(K) \to states(K')$ s. t. $\eta \xleftrightarrow[strict]{f} \eta'$

Proof. By previous lemma 10.11, $q_{Y_{\mathcal{A}}}$ and $q_{Y_{\mathcal{B}}}$ are strictly equivalent. Thus by previous lemma 10.9, it exists f s. t. $\eta_{(Y_{\mathcal{A}},q_{Y_{\mathcal{A}}},a)} \xleftrightarrow[conf]{f} \eta_{(Y_{\mathcal{B}},q_{Y_{\mathcal{B}}},a)}$. Let two corresponding states $(q'_{Y_{\mathcal{A}}}, q'_{Y_{\mathcal{B}}}) \in supp(\eta_{(Y_{\mathcal{A}},q_{Y_{\mathcal{A}}},a)}) \times \eta_{(Y_{\mathcal{B}},q_{Y_{\mathcal{B}}},a)}$ s. t. $f(q'_{Y_{\mathcal{A}}}) = q'_{Y_{\mathcal{B}}}$. We have $q'_{Y_{\mathcal{A}}} R_{conf} q'_{Y_{\mathcal{B}}}$ (*). Furthermore, since $q_{Y_{\mathcal{A}}} R_{strict} q_{Y_{\mathcal{B}}}$, $sig(Y_{\mathcal{A}})(q_{Y_{\mathcal{A}}}) = sig(Y_{\mathcal{B}})(q_{Y_{\mathcal{B}}})$, namely $ext(Y_{\mathcal{A}})(q_{Y_{\mathcal{A}}}) = ext(Y_{\mathcal{B}})(q_{Y_{\mathcal{B}}})$, which means $trace(\alpha_{Y_{\mathcal{A}}}^{\frown} q_{Y_{\mathcal{A}}} a q'_{Y_{\mathcal{A}}}) = trace(\pi_{Y_{\mathcal{B}}}^{\frown} q_{Y_{\mathcal{B}}} a q'_{Y_{\mathcal{B}}})$. So we can reapply previous lemma to obtain $q'_{Y_{\mathcal{A}}} R_{strict} q'_{Y_{\mathcal{B}}}$ which ends the proof.

□

Now we can finally show that if $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ are corresponding w. r. t. $\mathcal{A}$, $\mathcal{B}$ then $X_{\mathcal{A}} \setminus \{\mathcal{A}\}$ and $X_{\mathcal{B}} \setminus \{\mathcal{B}\}$ are semantically-equivalent which was the main aim of this subsection.

Theorem 10.13 ($X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ corresponding w. r. t. $\mathcal{A}, \mathcal{B}$ implies $X_{\mathcal{A}} \setminus \{\mathcal{A}\}$ and $X_{\mathcal{B}} \setminus \{\mathcal{B}\}$ semantically-equivalent). *Let $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ be two PCA corresponding w. r. t. $\mathcal{A}, \mathcal{B}$. Let $Y_{\mathcal{A}} = X_{\mathcal{A}} \setminus \{\mathcal{A}\}$ and $Y_{\mathcal{B}} = X_{\mathcal{B}} \setminus \{\mathcal{B}\}$.*

The PCA $Y_{\mathcal{A}}$ and $Y_{\mathcal{B}}$ are semantically-equivalent.

Proof. We recursively construct a strong complete bijective PCA executions-matching $(f_s, f_s^{tran}, f_s^{ex})$ where $f_s : reachable_{\leq s}(Y_{\mathcal{A}}) \rightarrow reachable_{\leq s}(Y_{\mathcal{B}})$ and $f_s^{ex} : \{\alpha \in Execs(Y_{\mathcal{A}}) || |\alpha| \leq s\} \rightarrow \{\pi \in Execs(Y_{\mathcal{B}}) || |\pi| \leq s\}$ s. t. $f_s^{ex}(\alpha) = \pi$ implies $lstate(\alpha) R_{strict} lstate(\pi)$.

Basis: $s = 0$, $reachable_{\leq 0}(Y_{\mathcal{A}}) = \{\bar{q}_{X_{\mathcal{A}}}\}$, while $reachable_{\leq 0}(Y_{\mathcal{B}}) = \{\bar{q}_{X_{\mathcal{B}}}\}$.

By definition 6.16 of corresponding automata $config(X_{\mathcal{A}})(\bar{q}_{X_{\mathcal{A}}}) \lhd_{\mathcal{AB}} config(X_{\mathcal{B}})(\bar{q}_{X_{\mathcal{B}}})$, while $(\bar{q}_{Y_{\mathcal{A}}}, \bar{q}_{Y_{\mathcal{B}}}) = (X_{\mathcal{A}}.\mu_s^{\mathcal{A}}(\bar{q}_{X_{\mathcal{A}}}), X_{\mathcal{B}}.\mu_s^{\mathcal{B}}(\bar{q}_{X_{\mathcal{B}}}))$ by definition 8.11 of PCA projection, which gives $\bar{q}_{Y_{\mathcal{A}}} R_{conf} \bar{q}_{Y_{\mathcal{B}}}$ by lemma 10.2. Moreover $trace_{Y_{\mathcal{A}}}(\bar{q}_{Y_{\mathcal{A}}}) = trace_{Y_{\mathcal{B}}}(\bar{q}_{Y_{\mathcal{B}}}) = \lambda$ (λ denotes the empty sequence). Thus we can apply lemma 10.11 to obtain $\bar{q}_{Y_{\mathcal{A}}} R_{strict} \bar{q}_{Y_{\mathcal{B}}}$. We construct $f_0(\bar{q}_{Y_{\mathcal{A}}}) = \bar{q}_{Y_{\mathcal{B}}}$, $f_0^{ex}(\bar{q}_{Y_{\mathcal{A}}}) = \bar{q}_{Y_{\mathcal{B}}}$. Clearly f_0 is a bijection from $reachable_0(Y_{\mathcal{A}})$ to $reachable_0(Y_{\mathcal{B}})$, while f_0^{ex} is a bijection from $Execs_0(Y_{\mathcal{A}})$ to $Execs_0(Y_{\mathcal{B}})$

Induction: We assume the result to be true for an integer $s \in \mathbb{N}$ and we will show it is then true for $s+1$. Let $Execs_s(Y_{\mathcal{A}}) = \{\alpha \in Execs(Y_{\mathcal{A}}) || |\alpha| = s\}$ and $Execs_s(Y_{\mathcal{B}}) = \{\alpha \in Execs(Y_{\mathcal{B}}) || |\pi| = s\}$.

We can build f_{s+1} (resp. f_{s+1}^{ex}) s. t. $\forall q \in reachable_{\leq s}, f_{s+1}(q) = f_s(q)$ (resp. s. t. $\forall \alpha \in Execs_{\leq s}(Y_{\mathcal{A}})\ f_{s+1}^{ex}(\alpha) = f_s^{ex}(\alpha)$) and $\forall q_{Y_{\mathcal{A}}}^j \in reachable_{s+1}, f_{s+1}(q^*)$ (resp. $\forall \alpha^{a,j} \in Execs_s(Y_{\mathcal{A}}), f_{s+1}^{ex}(\alpha')$) is built as follows:

We note $\alpha^{a,j} = \alpha \widehat{}_{Y_{\mathcal{A}}} q_{Y_{\mathcal{A}}} a q_{Y_{\mathcal{A}}}^j$ ($q_{Y_{\mathcal{A}}} = lstate(\alpha_{Y_{\mathcal{A}}})$). We note $\pi_{Y_{\mathcal{B}}} = f_s^{ex}(\alpha_{Y_{\mathcal{A}}})$. By induction assumption, $q_{Y_{\mathcal{A}}} R_{strict} q_{Y_{\mathcal{B}}}$ with $q_{Y_{\mathcal{A}}} = lstate(\alpha_{Y_{\mathcal{A}}})$ and $q_{Y_{\mathcal{B}}} = lstate(\pi_{Y_{\mathcal{B}}})$. Hence $sig(Y_{\mathcal{A}})(q_{Y_{\mathcal{A}}}) = sig(Y_{\mathcal{B}})(q_{Y_{\mathcal{B}}})$ and by previous lemma 10.12, for every $a \in sig(Y_{\mathcal{A}})(q_{Y_{\mathcal{A}}}) = sig(Y_{\mathcal{B}})(q_{Y_{\mathcal{B}}})$, $\exists g_a^j$, $\eta_{(Y_{\mathcal{A}}, q_{Y_{\mathcal{A}}}, a)} \xleftrightarrow[strict]{g_a^j} \eta_{(Y_{\mathcal{B}}, q_{Y_{\mathcal{B}}}, a)}$.

Hence, we define $f_{s+1}^{ex} : \alpha^{a,j} = \alpha \widehat{}_{Y_{\mathcal{A}}} q_{Y_{\mathcal{A}}} a q_{Y_{\mathcal{A}}}^j \mapsto f_{s+1}^{ex}(\alpha_{Y_{\mathcal{A}}}) \widehat{} f_s(q_{Y_{\mathcal{A}}}) a g_a^j(q_{Y_{\mathcal{A}}}^j)$, while f_{s+1} is naturally defined via f_{s+1}^{ex}, i. e. for every $q_{Y_{\mathcal{A}}}^j \in reachable_{s+1}(Y_{\mathcal{A}})$, we note $\alpha^{a,j} \in Execs_{s+1}(Y_{\mathcal{A}})$ s. t. $lstate(\alpha^{a,j}) = q_{Y_{\mathcal{A}}}^j$ and $f_{s+1}(q_{Y_{\mathcal{A}}}^j) = g_a^j(q_{Y_{\mathcal{A}}}^j) = lstate(f_{s+1}^{ex}(\alpha^{a,j}))$.

We finally define $f^{ex} : q^0 a^1 ... a^n q^n ... \mapsto f_0(q^0) a^1 ... a^n f_n(q^n)$, $f : q \mapsto f_n(q)$ where $q = lstate(q^0 a^1 ... q^n)$ and $f^{tr} : (q, a, \eta_{(Y_{\mathcal{A}}, q, a)}) \mapsto (f(q), a, \eta_{(Y_{\mathcal{B}}, f(q), a)})$.

Clearly (f, f^{tr}, f^{ex}) is strong since for every pair $(q_{Y_{\mathcal{A}}}, q_{Y_{\mathcal{B}}})$, s. t. $f(q_{Y_{\mathcal{A}}}) = q_{Y_{\mathcal{B}}}$, $q_{Y_{\mathcal{A}}} R_{strict} q_{Y_{\mathcal{B}}}$.

Moreover, (f, f^{tr}, f^{ex}) is complete since $dom(f) = reachable(Y_{\mathcal{A}}) = states(Y_{\mathcal{A}})$.

Finally, the bijectivity of f^{ex} is given by the inductive bijective construction.

Hence (f, f^{tr}, f^{ex}) is strong complete bijective PCA executions-matching from $Y_{\mathcal{A}}$ to $Y_{\mathcal{B}}$ which ends the proof.

□

11 TOP/DOWN CORRESPONDING CLASSES (BIS)

In previous section 10, we have shown in theorem 10.13 that if $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ are corresponding w. r. t. $\mathcal{A}$ and $\mathcal{B}$ (in the sense of definition 6.16), then $Y_{\mathcal{A}} = X_{\mathcal{A}} \setminus \{\mathcal{A}\}$ and $Y_{\mathcal{B}} = X_{\mathcal{B}} \setminus \{\mathcal{B}\}$ are semantically equivalent. We can note Y an arbitrary PCA semantically equivalent with both $Y_{\mathcal{A}}$ and $Y_{\mathcal{B}}$.

In section 9, we have shown in theorem 9.19 that for every PCA $\mathcal{E}$ environment of both $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$, $X_{\mathcal{A}} || \mathcal{E}$ and $\tilde{\mathcal{A}}^{sw} || Y_{\mathcal{A}} || \mathcal{E}$ (resp. $X_{\mathcal{B}} || \mathcal{E}$ and $\tilde{\mathcal{B}}^{sw} || Y_{\mathcal{B}} || \mathcal{E}$) are linked by a PCA executions-matching

It is time to combine this two results to realise that for every PCA $\mathcal{E}$ environment of both $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$, $X_{\mathcal{A}}||\mathcal{E}$ and $\tilde{\mathcal{A}}^{sw}||\mathcal{E}'$ (resp. $X_{\mathcal{B}}||\mathcal{E}$ and $\tilde{\mathcal{B}}^{sw}||\mathcal{E}'$) are linked by a PCA executions-matching where $\mathcal{E}' = \mathcal{E}||Y$.

Hence (*) if $\mathcal{E}'$ cannot distinguish $\tilde{\mathcal{A}}^{sw}$ from $\tilde{\mathcal{B}}^{sw}$, we will be able to show that $\mathcal{E}$ cannot distinguish $X_{\mathcal{A}}$ from $X_{\mathcal{B}}$.

In this section, we formalise (*) in theorem 11.25 of monotonicity of implementation relation. However, some assumptions are required to reduce the implementation of $X_{\mathcal{B}}$ by $X_{\mathcal{A}}$ into implementation of $\mathcal{B}$ by $\mathcal{A}$. These are all minor technical assumptions except for one: our implementation relation concerns only a particular subset of schedulers so-called *creation-oblivious*, i. e. in order to compute (potentially randomly) the next transition, they do not take into account the internal actions of a sub-automaton preceding its last destruction.

11.1 Creation-oblivious scheduler

Here we recall the definition of creation-oblivious scheduler (already introduced in subsection 6.4), that does not take into account previous internal actions of a particular sub-automaton to output its probability over transitions to trigger.

We start by defining *strict oblivious-schedulers* that output the same transition with the same probability for pair of execution fragments that differ only by prefixes in the same class of equivalence. This definition is inspired by the one provided in the thesis of Segala, but is more restrictive since we require a strict equality instead of a correlation (section 5.6.2 in [8]).

Definition 11.1 (strict oblivious scheduler (recall)). Let $\tilde{W}$ be a PCA or a PSIOA, let $\tilde{\sigma} \in schedulers(\tilde{W})$ and let $\equiv$ be an equivalence relation on $Frags^*(\tilde{W})$ verifying $\forall \tilde{\alpha_1}, \tilde{\alpha_2} \in Frags^*(\tilde{W})$ s. t. $\tilde{\alpha_1} \equiv \tilde{\alpha_2}$, $lstate(\alpha_1) = lstate(\alpha_2)$. We say that $\tilde{\sigma}$ is $(\equiv)$-strictly oblivious if $\forall \tilde{\alpha_1}, \tilde{\alpha_2}, \tilde{\alpha_3} \in Frags^*(\tilde{W})$ s. t. 1) $\alpha_1 \equiv \alpha_2$ and 2) $fstate(\tilde{\alpha_3}) = lstate(\tilde{\alpha_2}) = lstate(\tilde{\alpha_1})$, then $\tilde{\sigma}(\tilde{\alpha_1}^\frown \tilde{\alpha_3}) = \tilde{\sigma}(\tilde{\alpha_2}^\frown \tilde{\alpha_3})$.

Now we define the relation of equivalence that defines our subset of creation-oblivious schedulers. Intuitively, two executions fragments ending on $\mathcal{A}$ creation are in the same equivalence class if they differ only in terms of internal actions of $\mathcal{A}$.

Definition 11.2 ($\tilde{\alpha} \equiv^{cr}_{\mathcal{A}} \tilde{\alpha}'$ (recall)). Let $\tilde{\mathcal{A}}$ be a PSIOA, $\tilde{W}$ be a PCA, $\forall \tilde{\alpha}, \tilde{\alpha}' \in Frags^*(\tilde{W})$, we say $\tilde{\alpha} \equiv^{cr}_{\mathcal{A}} \tilde{\alpha}'$ iff:

(1) $\tilde{\alpha}, \tilde{\alpha}'$ both ends on $\mathcal{A}$-creation.
(2) $\tilde{\alpha}$ and $\tilde{\alpha}'$ differ only in the $\mathcal{A}$-exclusive actions and the states of $\mathcal{A}$, i. e. $\mu(\tilde{\alpha}) = \mu(\tilde{\alpha}')$ where $\mu(\tilde{\alpha} = \tilde{q}^0 a^1 \tilde{q}^1 ... a^n \tilde{q}^n) \in Frags^*(\tilde{W})$ is defined as follows:
   - remove the $\mathcal{A}$-exclusive actions
   - replace each state $\tilde{q}^i$ by its configuration $Config(\tilde{W})(\tilde{q}) = (\mathbf{A}^i, \mathbf{S}^i)$
   - replace each configuration $(\mathbf{A}^i, \mathbf{S}^i)$ by $(\mathbf{A}^i, \mathbf{S}^i) \setminus \{\mathcal{A}\}$
   - replace the (non-alternating) sequences of identical configurations (due to $\mathcal{A}$-exclusiveness of removed actions) by one unique configuration.
(3) $trace(\tilde{\alpha}) = trace(\tilde{\alpha}')$,
(4) $lstate(\alpha_1) = lstate(\alpha_2)$

We can remark that the items 4 can be deduced from 1 and 2 if X is configuration-conflict-free. We can also remark that if $\tilde{W}$ is a $\mathcal{A}$-conservative PCA, we can replace $\mu(\tilde{\alpha}) = \mu(\tilde{\alpha}')$, by $\mu^{\mathcal{A}}_e(\tilde{\alpha}) \upharpoonright (\tilde{W} \setminus \{\mathcal{A}\}) = \mu^{\mathcal{A}}_e(\tilde{\alpha}') \upharpoonright (\tilde{W} \setminus \{\mathcal{A}\})$ but we want to be as general as possible for next definition of *creation oblivious scheduler*:

Definition 11.3 (creation-oblivious scheduler). Let $\tilde{\mathcal{A}}$ be a PSIOA, $\tilde{W}$ be a PCA, $\tilde{\sigma} \in schedulers(\tilde{W})$. We say that $\tilde{\sigma}$ is $\mathcal{A}$-*creation oblivious* if it is $(\equiv^{cr}_{\mathcal{A}})$-strictly oblivious.

We say that $\tilde{\sigma}$ is *creation-oblivious* if it is $\mathcal{A}$-creation oblivious for every sub-automaton $\mathcal{A}$ of $\tilde{W}$ ($\mathcal{A} \in \bigcup_{q \in states(\tilde{W})} auts(config(\tilde{W})(q))$). We note $CrOB$ the function that maps every PCA $\tilde{W}$ to the set of creation-oblivious schedulers of $\tilde{W}$. If W is not a PCA but a PSIOA, $CrOB(W) = schedulers(W)$.

If $\tilde{W}$ is $\mathcal{A}$-conservative and $\tilde{\sigma}$ is $\mathcal{A}$-creation oblivious, we note $oblivious_{\mathcal{A},\beta,e}(\tilde{\sigma})$ (and usually $\tilde{\sigma}_{|\mathcal{A},\beta,e}$ when it is clear in the context) the (unique by definition) scheduler s. t. for every $\tilde{\alpha}, \tilde{\alpha}' \in Frags^*(\tilde{W})$ with i) $\tilde{\alpha}$ is ending on $\mathcal{A}$-creation , ii) $\tilde{W}.\mu_e^{\mathcal{A}}(\tilde{\alpha}) \upharpoonright (\tilde{W} \setminus \{\mathcal{A}\}) = e$, iii) $trace(\tilde{\alpha}) = \beta$ and iv) $fstate(\tilde{\alpha}') = lstate(\tilde{\alpha})$, then $\tilde{\sigma}_{|\mathcal{A},\beta,e}(\tilde{\alpha}') = \tilde{\sigma}(\tilde{\alpha}^\frown \tilde{\alpha}')$. Let us note that $lstate(\tilde{\alpha})$ is entirely defined with i) and ii). We remark that if such an execution fragment $\tilde{\alpha} \in Frags^*(\tilde{W})$ verifying i), ii) and iii) exists, then $\tilde{\sigma}_{|\mathcal{A},\beta,e} = \tilde{\sigma}_{|\tilde{\alpha}}$, the sub-scheduler conditioned by $\tilde{\sigma}$ and $\tilde{\alpha}$ in the sense of definition 11.4 stated immediately below.

Definition 11.4 (conditioned scheduler). Let $\mathcal{A}$ be a PSIOA, $\sigma \in schedulers(\mathcal{A})$ and let $\alpha_1 \in Frags^*(\mathcal{A})$. We note $\sigma_{|\alpha_1} : \{\alpha_2 \in Frags^*(\mathcal{A}) | fstate(\alpha_2) = lstate(\alpha_1)\} \to SubDisc(dtrans(\mathcal{A}))$ the *sub-scheduler conditioned by σ and α_1* that verifies $\forall \alpha_2 \in Frags^*(\mathcal{A}), fstate(\alpha_2) = lstate(\alpha_1), \sigma_{|\alpha_1}(\alpha_2) = \sigma(\alpha_1^\frown \alpha_2)$.

We take the opportunity to state a lemma of conditional probability that will be used later for lemma 11.24.

Lemma 11.5 (conditional measure law). *Let $\mathcal{A}$ be a PSIOA, $\sigma \in schedulers(\mathcal{A})$ and let $\alpha_1 \in Frags^*(\mathcal{A})$ and $\sigma_{|\alpha_1}$ the sub-scheduler conditioned by σ and α_1. Let $\alpha_o, \alpha_2 \in Frags^*(\mathcal{A}), fstate(\alpha_2) = lstate(\alpha_1) \triangleq q_{12}$. Then*

$$\epsilon_{\sigma,\alpha_o}(C_{\alpha_1^\frown \alpha_2}) =: \begin{cases} \epsilon_{\sigma,\alpha_o}(C_{\alpha_1}) \cdot \epsilon_{\sigma_{|\alpha_1},q_{12}}(C_{\alpha_2}) & if\, \alpha_1 \not\leq \alpha_o \\ \epsilon_{\sigma_{|\alpha_1},\alpha_o'}(C_{\alpha_2}) & if\, \alpha_o = \alpha_1^\frown \alpha_o' \end{cases}$$

Proof. We note $\alpha_{12} = \alpha_1^\frown \alpha_2$.

(1) $\alpha_1 \not\leq \alpha_o$:
   (a) $\alpha_1 \not\leq \alpha_o$ and $\alpha_o \not\leq \alpha_1$:
   This implies $\alpha_{12} \not\leq \alpha_o$ and $\alpha_o \not\leq \alpha_{12}$ thus $\epsilon_{\sigma,\alpha_o}(C_{\alpha_1^\frown \alpha_2}) = \epsilon_{\sigma,\alpha_o}(C_{\alpha_1}) = 0$ which ends the proof.
   (b) $\alpha_o \leq \alpha_1$:
   This implies $\alpha_o \leq \alpha_{12}$ By induction on size s of α_2. Basis: $s = 0$, i. e. $\alpha_2 = lstate(\alpha_1) = q_{12}$. Thus, we meet the second case of definition of $\epsilon_{\sigma|_{\alpha_1},q_{12}}(C_{\alpha_2})$: $\alpha_2 \leq q_{12}$, which means $\epsilon_{\sigma|_{\alpha_1},q_{12}}(C_{\alpha_2}) = 1$ and terminates the basis. Induction: We assume the result to be true up to size $s \in \mathbb{N}$ and we want to show it is still true for size $s+1$. Let $\alpha_2 \in Frags^*(\mathcal{A}), fstate(\alpha_2) = lstate(\alpha_1) \triangleq q_{12}$ with $|\alpha_2| = s+1$. We note $\alpha_2 = \alpha_2'^\frown q'aq$ and $\alpha_{12}' = \alpha_1^\frown \alpha_2'$. We have $|\alpha_2'| = s$ and $\alpha_o \leq \alpha_{12}'$
   By definition we have $\epsilon_{\sigma|_{\alpha_1},q_{12}}(C_{\alpha_2}) = \epsilon_{\sigma|_{\alpha_1},q_{12}}(C_{\alpha_2'}) \cdot \sigma(\alpha_2')(\eta_{(\mathcal{A},q',a)}) \cdot \eta_{(\mathcal{A},q',a)}(q)$.
   In Parallel, by definition: $\epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}}) = \epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}'}) \cdot \sigma(\alpha_{12}')(\eta_{(\mathcal{A},q',a)}) \cdot \eta_{(\mathcal{A},q',a)}(q)$ and by induction assumption, $\epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}}) = \epsilon_{\sigma,\alpha_o}(C_{\alpha_1}) \cdot \epsilon_{\sigma|_{\alpha_1},q_{12}}(C_{\alpha_2'}) \cdot \sigma(\alpha_{12}')(\eta_{(\mathcal{A},q',a)}) \cdot \eta_{(\mathcal{A},q',a)}(q)$ and so $\epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}}) = \epsilon_{\sigma,\alpha_o}(C_{\alpha_1}) \cdot \epsilon_{\sigma|_{\alpha_1},q_{12}}(C_{\alpha_2})$, which ends the induction and so the case.

(2) $\alpha_o = \alpha_1^\frown \alpha_o'$. By definition, $\epsilon_{\sigma,\alpha_o}(C_{\alpha_1}) = 1$
   (a) both $\alpha_{12} \not\leq \alpha_o$ and $\alpha_o \not\leq \alpha_{12}$. This implies $\alpha_2 \not\leq \alpha_o'$ and $\alpha_o' \not\leq \alpha_2$ Then, by definition, $\epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}}) = \epsilon_{\sigma|_{\alpha_1},\alpha_o'}(C_{\alpha_2}) = 0$.
   (b) $\alpha_{12} \leq \alpha_o$. This implies $\alpha_2 \leq \alpha_o'$. Then, by definition, $\epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}}) = \epsilon_{\sigma_{|\alpha_1},\alpha_o'}(C_{\alpha_2}) = 1$
   (c) $\alpha_o \leq \alpha_{12}$:
   We proceed by induction on size s of α_2.
   Basis: $s = 0$, i. e. $\alpha_2 = q_{12}$. Then by definition $\epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}}) = \epsilon_{\sigma,\alpha_o}(C_{\alpha_1}) = 1$. Moreover $q_{12} \leq \alpha_o'$ which means $\epsilon_{\sigma_{|\alpha_1},\alpha_o'}(C_{\alpha_2}) = 1$, which ends the basis.

Induction:
We assume the result to be true up to size $s \in \mathbb{N}$ and we want to show it is still true for size $s+1$. Let $\alpha_2 \in Frags^*(\mathcal{A}), fstate(\alpha_2) = lstate(\alpha_1) \triangleq q_{12}$ with $|\alpha_2| = s+1$. We note $\alpha_2 = \alpha_2'^\frown q'aq$ and $\alpha_{12}' = \alpha_1'^\frown \alpha_2'$. We have $|\alpha_2'| = s$ and $\alpha_o \leq \alpha_{12}'$.
By definition we have $\epsilon_{\sigma_{|\alpha_1},\alpha_o'}(C_{\alpha_2}) = \epsilon_{\sigma_{|\alpha_1},\alpha_o'}(C_{\alpha_2'}) \cdot \sigma(\alpha_2')(\eta_{(\mathcal{A},q',a)}) \cdot \eta_{(\mathcal{A},q',a)}(q)$.
In Parallel, by definition: $\epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}}) = \epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}'}) \cdot \sigma(\alpha_{12}')(\eta_{(\mathcal{A},q',a)}) \cdot \eta_{(\mathcal{A},q',a)}(q)$ and by induction assumption, $\epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}}) = \epsilon_{\sigma,\alpha_o}(C_{\alpha_1}) \cdot \epsilon_{\sigma_{|\alpha_1},\alpha_o'}(C_{\alpha_2'}) \cdot \sigma(\alpha_{12}')(\eta_{(\mathcal{A},q',a)}) \cdot \eta_{(\mathcal{A},q',a)}(q)$ and so $\epsilon_{\sigma,\alpha_o}(C_{\alpha_1^\frown\alpha_2}) = \epsilon_{\sigma,\alpha_o}(C_{\alpha_1}) \cdot \epsilon_{\sigma_{|\alpha_1},\alpha_o'}(C_{\alpha_2})$. Finally, since $\epsilon_{\sigma,\alpha_o}(C_{\alpha_1}) = 1$, we have $\epsilon_{\sigma,\alpha_o}(C_{\alpha_{12}}) = \epsilon_{\sigma_{|\alpha_1},\alpha_o'}(C_{\alpha_2})$ which ends the induction, the case and so the proof.

□

We have formally defined our notion of creation-oblivious scheduler. This will be a key property to ensure lemma 11.23 that allows to reduce the measure of a class of comportment as a function of measures of classes of shorter comportment where no creation of $\mathcal{A}$ or $\mathcal{B}$ occurs excepting potentially at very last action. This reduction is more or less necessary to obtain monotonicity of implementation relation.

11.2 Creation made explicit

In this subsection, we recall notion of *creation-explicitness* (already introduced in subsection 6.2). This property will allow us to obtain the reduction of lemma 11.23 mentioned in last paragraph.

Definition 11.6 (creation-explicit PCA). Let $\mathcal{A}$ be a PSIOA and X be a PCA. We say that X is $\mathcal{A}$*-creation-explicit* iff: it exists a set of actions, noted $creation\text{-}actions(X)(\mathcal{A})$, s. t. $\forall q_X \in states(X)$, $\forall a \in \widehat{sig}(X)(q_X)$, if we note $\mathbf{A}_X = auts(config(X)(q_X))$ and $\varphi_X = created(X)(q_X)(a)$, then $\mathcal{A} \notin \mathbf{A}_X \wedge \mathcal{A} \in \varphi_X \iff a \in creation\text{-}actions(X)(\mathcal{A})$.

This property of creation-explicitness will clarify the condition to obtain surjectivity of $\tilde{\mu}_e^{\mathcal{A},+}$ since it suffices to consider this function with a restricted range where no action of $creation\text{-}actions(X)(\mathcal{A})$ appears before last action.

Lemma 11.7 (Partial surjectivity with explicit creation). *Let $\mathcal{A}$ be a PSIOA and X be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit PCA. Let $\tilde{\mathcal{E}}$ be partially-compatible with X. Let $Y = X \setminus \{\mathcal{A}\}$. Let $\mathcal{E}_{\mathcal{A}} = \tilde{\mathcal{E}}||Y$. Let $(((\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A}}, (\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A},+}), (\tilde{\mathcal{E}}||X).\tilde{\mu}_{tr}^{\mathcal{A},+}, (\tilde{\mathcal{E}}||X).\tilde{\mu}_e^{\mathcal{A},+})$ the $\tilde{\mathcal{E}}$-extension of $((X.\tilde{\mu}_z^{\mathcal{A}}, X.\tilde{\mu}_z^{\mathcal{A},+}), X.\tilde{\mu}_{tr}^{\mathcal{A},+}, X.\tilde{\mu}_e^{\mathcal{A},+})$. Let $\alpha, \alpha' \in Execs(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw})$ s. t. $creation\text{-}actions(X)(\mathcal{A}) \cap actions(\alpha) = \emptyset$*

1) Then $\exists \tilde{\alpha} \in dom(\tilde{\mu}_e^{\mathcal{A}})$ s. t. $\tilde{\mu}_e^{\mathcal{A},+}(\tilde{\alpha}) = \tilde{\mu}_e^{\mathcal{A}}(\tilde{\alpha}) = \alpha$.

2) If $\alpha' = \alpha^\frown q, a_!, q'$ with $a_! \in creation\text{-}actions(X)(\mathcal{A})$, then $\exists \tilde{\alpha}' \in dom(\tilde{\mu}_e^{\mathcal{A},+})$ s. t. $\tilde{\mu}_e^{\mathcal{A},+}(\tilde{\alpha}') = \alpha'$.

Proof. We proof the results in the same order they are stated in the lemma:

(1) We note $\alpha = q^0, a^1, ..., a^n, q^n ...$ and we proof the result by induction on the prefix size s. Basis: the result trivially holds for any execution α of size 0 by construction of $X \setminus \{\mathcal{A}\}$ that requires $X.\mu_s^{\mathcal{A}}(start(X)) = start(X \setminus \{\mathcal{A}\})$. We assume the result holds up to prefix size s and we show it still holds for prefix size $s+1$. We note $\alpha_s = q^0, a^1, ..., a^s, q^s$ and $\tilde{\alpha}^s \in Execs(\tilde{\mathcal{E}}||X)$ s. t. $\tilde{\mu}_e^{\mathcal{A}}(\tilde{\alpha}_s) = \alpha_s$. By lemma 9.17 of signature preservation $a^{s+1} \in sig(\tilde{\mathcal{E}}||X)(\tilde{q}_s)$. Moreover, by assumption $a^{s+1} \notin creation\text{-}actions(X)(\mathcal{A})$ which means the application of lemma 9.8 of homomorphic transitions lead us to $\eta_{(\tilde{\mathcal{E}}||X),\tilde{q}^s,a^{s+1}} \xleftrightarrow{\mu_z^{\mathcal{A}}} \eta_{(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw}),q^s,a^{s+1}}$. So it exists $\tilde{q}^{s+1} \in supp(\eta_{(\tilde{\mathcal{E}}||X),\tilde{q},a_!})$ with $\mu_z^{\mathcal{A}}(\tilde{q}) = q$. So $\mu_e^{\mathcal{A}}(\tilde{\alpha}_s^\frown \tilde{q}^s a^{s+1} \tilde{q}^{s+1}) = \alpha_{s+1}$. This ends the induction and so the proof of 1. .

(2) We apply 1. and note $\tilde{\alpha} \in Execs(\tilde{\mathcal{E}}||X)$ s. t. $\tilde{\mu}_e^{\mathcal{A}}(\tilde{\alpha}) = \alpha$. By lemma 9.17 of signature preservation $a_! \in sig(\tilde{\mathcal{E}}||X)(\tilde{q})$ with $\tilde{q} = lstate(\alpha)$. Moreover, by lemma 9.8 of homomorphic transition, $\eta_{(\tilde{\mathcal{E}}||X),\tilde{q},a_!} \overset{\mu_z^{\mathcal{A},+}}{\longleftrightarrow} \eta_{(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw}),q,a_!}$. So it exists $\tilde{q}' \in supp(\eta_{(\tilde{\mathcal{E}}||X),\tilde{q},a_!})$ with $\mu_z^{\mathcal{A},+}(\tilde{q}') = q'$. So $\mu_e^{\mathcal{A},+}(\tilde{\alpha}^\frown \tilde{q}a_!\tilde{q}') = \alpha'$ which ends the proof.

□

Since we i) classify executions in some classes according to their projection on an environment and ii) are concerned by the actions of the execution that create $\mathcal{A}$, the next lemma will simplify this classification. It states that if the projection e of an execution $\alpha \in Execs(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw})$ on the environment $\mathcal{E}_{\mathcal{A}}$ ends by an action $a_! \in creation\text{-}actions(X)(\mathcal{A})$, then the execution necessarily ends by $a_!$.

Lemma 11.8 (environment projection ends on creation implies the execution itself ends on creation). *Let $\mathcal{A}$ be a PSIOA and X be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit PCA. Let $\tilde{\mathcal{E}}$ be partially-compatible with X. Let $Y = X \setminus \{\mathcal{A}\}$. Let $\mathcal{E}_{\mathcal{A}} = \tilde{\mathcal{E}}||Y$.*

Let $(((\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A}}, (\tilde{\mathcal{E}}||X).\tilde{\mu}_z^{\mathcal{A},+}), (\tilde{\mathcal{E}}||X).\tilde{\mu}_{tr}^{\mathcal{A},+}, (\tilde{\mathcal{E}}||X).\tilde{\mu}_e^{\mathcal{A},+})$ the $\tilde{\mathcal{E}}$-extension of $((X.\tilde{\mu}_z^{\mathcal{A}}, X.\tilde{\mu}_z^{\mathcal{A},+}), X.\tilde{\mu}_{tr}^{\mathcal{A},+}, X.\tilde{\mu}_e^{\mathcal{A},+})$.

Let $a_! \in creation\text{-}actions(X)(\mathcal{A})$ and $\alpha \in Execs(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw})$ s. t. $\alpha \upharpoonright \mathcal{E}_{\mathcal{A}} = e' = e^\frown q, a_!, q'$ with $creation\text{-}actions(X)(\mathcal{A}) \cap actions(e) = \emptyset$.

Then it exist $\tilde{\alpha} \in dom(\tilde{\mu}_e^{\mathcal{A},+})$ s. t. $\tilde{\mu}_e^{\mathcal{A},+}(\tilde{\alpha}) = \alpha$.

Proof. We note $\alpha = \alpha^{1\frown} q_\ell^1, a_!, q_f^{2\frown}\alpha^2$

We have $q_\ell^1 \upharpoonright \tilde{\mathcal{A}}^{sw} = q^\phi_{\tilde{\mathcal{A}}^{sw}}$. Indeed let assume the contrary: $q_\ell^1 \upharpoonright \tilde{\mathcal{A}}^{sw} \neq q^\phi_{\tilde{\mathcal{A}}^{sw}}$. Then $q \upharpoonright \tilde{\mathcal{A}}^{sw} \neq q^\phi_{\tilde{\mathcal{A}}^{sw}}$ for every state $q \in \alpha^1$. Since $creation\text{-}actions(X)(\mathcal{A}) \cap actions(e) = \emptyset$, $creation\text{-}actions(X)(\mathcal{A}) \cap actions(\alpha^1) = \emptyset$. Thus we apply lemma 11.7 of partial surjectivity with explicit creation to obtain, it exists $\tilde{\alpha}^1 \in Execs(\tilde{\mathcal{E}}||X)$ s. t. $\tilde{\mu}_e^{\mathcal{A},+}(\tilde{\alpha}^1) = \alpha^1$ with both $\mathcal{A} \in auts(config(X)(lstate(\tilde{\alpha}^1) \upharpoonright X))$ and $a_! \in creation\text{-}actions(X)(\mathcal{A}) \cap sig(X)(lstate(\tilde{\alpha}^1)) \upharpoonright X)$ which is impossible.

Since $q_\ell^1 \upharpoonright \tilde{\mathcal{A}}^{sw} = q^\phi_{\tilde{\mathcal{A}}^{sw}}$, $q \upharpoonright \tilde{\mathcal{A}}^{sw} = q^\phi_{\tilde{\mathcal{A}}^{sw}}$ for every state $q \in \alpha^2$. Hence, $\alpha^2 = q_f^2$ to respect $\alpha \upharpoonright \mathcal{E}_{\mathcal{A}} = e'$, which means $\alpha = \alpha^{1\frown} q_\ell^1, a_!, q_f^2$.

Since $creation\text{-}actions(X)(\mathcal{A}) \cap actions(e) = \emptyset$, $creation\text{-}actions(X)(\mathcal{A}) \cap actions(\alpha^{12}) = \emptyset$. Thus we apply lemma 11.7 of partial surjectivity with explicit creation to obtain $\exists \tilde{\alpha} \in Execs(\tilde{\mathcal{E}}||X)$ s. t. $\tilde{\mu}_e^{\mathcal{A},+}(\tilde{\alpha}) = \alpha$.

□

Here we recall the notion of print (already introduced in 4.6). This notion captures the perception of a system $\tilde{\mathcal{E}}||X$ by the environment $\tilde{\mathcal{E}}$. This notion allows us to propose an intuitive definition of implementation that is monotonic w. r. t. PSIOA creation.

Definition 11.9 (print = trace + environment projection). Let K be a PSIOA (resp. a PCA). Let $\mathcal{E}$ be a PSIOA (resp. a PCA) partially-compatible with K. We note $print_{(\mathcal{E},K)} : \alpha \in Execs(\mathcal{E}||K) \mapsto (trace_{\mathcal{E}||K}(\alpha), \alpha \upharpoonright \mathcal{E})$. For every $\zeta = (\beta, e)$ where β is a sequence of actions and e an alternating sequence of states and action, we note $Class(Execs(\mathcal{E}||K), print_{(\mathcal{E},K)}, \zeta) = \{\alpha \in Execs(\mathcal{E}||K) | print_{(\mathcal{E},K)}(\alpha) = \zeta\}$.

The next definition allows to aggregate print-based classification into a larger classification in order to prepare the reduction of the implementation of a system $X_{\mathcal{B}}$ by a system $X_{\mathcal{A}}$ into the implementation of $\mathcal{B}$ by $\mathcal{A}$.

Definition 11.10 (aggregate print without creation). Let $\mathcal{A}$ be a PSIOA, let X be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit PCA. Let $\tilde{\mathcal{E}}$ be a PCA partially-compatible with X. Let $Y = X \setminus \{\mathcal{A}\}$. For every $\zeta = (\beta, e)$ where β is a sequence of actions and $e = e'^\frown q' a q$ an alternating sequence of states and actions with $actions(e') \cap creation\text{-}actions(X)(\mathcal{A}) = \emptyset$, then we note $Class(Execs(\tilde{\mathcal{E}}||X), print^{proxy,\mathcal{A}}_{(\tilde{\mathcal{E}},X)}, \zeta) = \{\tilde{\alpha} \in Execs(\tilde{\mathcal{E}}||X) | trace(\tilde{\alpha}) = \beta \wedge (\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A},+}_e(\tilde{\alpha}) \upharpoonright (\tilde{\mathcal{E}}||Y) = e\}$.

Now we state the bijectivity of $\tilde{\mu}^{\mathcal{A},+}_e$ (when no creation occurs) in terms of corresponding classes of external perception.

Lemma 11.11 ($\tilde{\mu}^{\mathcal{A},+}_e$ is a bijection from $\tilde{C}$ to C). *Let $\mathcal{A}$ be a PSIOA and X be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit PCA. Let $\tilde{\mathcal{E}}$ be partially-compatible with X. Let $Y = X\setminus\{\mathcal{A}\}$. Let $\mathcal{E}_{\mathcal{A}} = \tilde{\mathcal{E}}||Y$. Let $(((\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A}}_z, (\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A},+}_z), (\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A},+}_{tr}, (\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A},+}_e)$ the $\tilde{\mathcal{E}}$-extension of $((X.\tilde{\mu}^{\mathcal{A}}_z, X.\tilde{\mu}^{\mathcal{A},+}_z), X.\tilde{\mu}^{\mathcal{A},+}_{tr}, X.\tilde{\mu}^{\mathcal{A},+}_e)$.*

For every $\zeta = (\beta, e' = e^\frown q, a, q')$ where β is a sequence of actions and e' an alternating sequence of states and action with $creation\text{-}actions(X)(\mathcal{A}) \cap actions(e) = \emptyset$, $(\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A},+}_e$ is a bijection from $\tilde{C}$ to C, where

- $\tilde{C} = Class(Execs(\tilde{\mathcal{E}}||X), print^{proxy,\mathcal{A}}_{(\mathcal{E},X)}, \zeta)$
- $C = Class(Execs(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw}), print_{\mathcal{E}_{\mathcal{A}},\tilde{\mathcal{A}}^{sw}}, \zeta)$

Proof.
- Injectivity is immediate by lemma 7.11, item (2).
- Surjectivity: Let $\alpha \in C$. By previous lemma 11.7, $\exists \tilde{\alpha} \in Execs(\tilde{\mathcal{E}}||X)$ s. t. $\tilde{\mu}^{\mathcal{A},+}_e(\tilde{\alpha}) = \alpha$. Finally, since $\tilde{\mu}^{\mathcal{A},+}_e$ is a strong executions-matching, the trace is preserved which means $trace(\tilde{\alpha}) = trace(\alpha)$.

□

This bijectivity allows us to obtain the preservation of measure of probability for corresponding classes of external perception.

Lemma 11.12 (equiprobability of top/down corresponding cones). *Let $\mathcal{A}$ be a PSIOA and X be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit PCA. Let $\tilde{\mathcal{E}}$ be partially-compatible with X. Let $Y = X\setminus\{\mathcal{A}\}$. Let $\mathcal{E}_{\mathcal{A}} = \tilde{\mathcal{E}}||Y$. Let $(((\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A}}_z, (\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A},+}_z), (\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A}}_{tr}$ the $\tilde{\mathcal{E}}$-extension of $((X.\tilde{\mu}^{\mathcal{A}}_z, X.\tilde{\mu}^{\mathcal{A},+}_z), X.\tilde{\mu}^{\mathcal{A},+}_{tr}, X.\tilde{\mu}^{\mathcal{A},+}_e)$.*

Let $\zeta = (\beta, e' = e^\frown q, a, q')$ where β is a sequence of actions and e' an alternating sequence of states and action with $creation\text{-}actions(X)(\mathcal{A}) \cap actions(e) = \emptyset$,

Let $C_{\tilde{C}} = \bigcup_{\tilde{\alpha}\in\tilde{C}}\{\tilde{\alpha}' \in Execs(\tilde{\mathcal{E}}||X) | \tilde{\alpha} \preceq \tilde{\alpha}'\}$ and $C_C = \bigcup_{\alpha\in C}\{\alpha' \in Execs(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw}) | \alpha \preceq \alpha'\}$ with $\tilde{C} = Class(Execs(\tilde{\mathcal{E}}||X), print^{proxy,\mathcal{A}}_{(\mathcal{E},X)}, \zeta)$ and $C = Class(Execs(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw}), print_{\mathcal{E}_{\mathcal{A}},\tilde{\mathcal{A}}^{sw}}, \zeta)$.

Then for every $\tilde{\sigma} \in schedulers(\tilde{\mathcal{E}}||X)$, for σ $(((\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A}}_z, (\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A},+}_z), (\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A},+}_{tr}, (\tilde{\mathcal{E}}||X).\tilde{\mu}^{\mathcal{A},+}_e)$-alter ego of $\tilde{\sigma}$,

$\epsilon_{\tilde{\sigma},\delta_{\tilde{q}_{(\tilde{\mathcal{E}}||X)}}}(C_{\tilde{C}}) = \epsilon_{\sigma,\delta_{\tilde{q}_{(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw})}}}(C_C)$

Proof. By lemma 11.11, $\tilde{\mu}^{\mathcal{A},+}_e$ is a bijection from $\tilde{C}$ to C. We note $\{(\tilde{\alpha}_i, \alpha_i)\}_{i\in I} = \tilde{C}\times C$ the related pairs of executions s. t. $\tilde{\mu}^{\mathcal{A},+}_e(\tilde{\alpha}_i) = \alpha_i$. We obtain $\epsilon_{\tilde{\sigma},\delta_{\tilde{q}_{(\tilde{\mathcal{E}}||X)}}}(C_{\tilde{C}}) = \sum_{i\in I}\epsilon_{\tilde{\sigma},\delta_{\tilde{q}_{(\tilde{\mathcal{E}}||X)}}}(\tilde{\alpha}_i)$ and $\epsilon_{\sigma,\delta_{\tilde{q}_{(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw})}}}(C_C) = \sum_{i\in I}\epsilon_{\sigma,\delta_{\tilde{q}_{(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw})}}}(C_{\alpha_i})$.

Thus it is enough to show that $\forall i \in I, \epsilon_{\tilde{\sigma},\delta_{\tilde{q}_{(\tilde{\mathcal{E}}||X)}}}(C_{\tilde{\alpha}_i}) = \epsilon_{\sigma,\delta_{\tilde{q}_{(\mathcal{E}_{\mathcal{A}}||\tilde{\mathcal{A}}^{sw})}}}(C_{\alpha_i})$ which is given by theorem 7.10 that can be applied since $\tilde{\mu}^{\mathcal{A},+}_e$ is a continued executions-matching by theorem 9.23.

□

In next subsection, we want to extend this lemma 11.12 to any case with an arbitrary number of destructions and creations of the sub-automaton $\mathcal{A}$.

11.3 Monotonicity

In this subsection, we want to reduce the computation of the probability measure of a class of executions to the computation of several classes without creation (excepting at very last action) in order to apply lemma 11.12 and then use the implementation of $\mathcal{B}$ by $\mathcal{A}$.

We start by a sequence of definitions to identify easily an execution α as the concatenations of several execution fragments α_i where each α_i does not contain creation actions except for very last action.

Definition 11.13 (n-building-vector for traces). Let β be a sequence of actions . Let $n \in \mathbb{N} \cup \{\infty\}$. A *n-building-vector* of β is a (potentially infinite) vector $\overrightarrow{\beta} = (\beta^1, ..., \beta^i, ...)$ of $|\overrightarrow{\beta}| = n$ sequences of actions s. t. $\beta^1{}^\frown ... \beta^{i-1}{}^\frown \beta^i{}^\frown ... = \beta$. We note $Building\text{-}vectors(\beta, n)$ the set of n-building-vector of β and $\overrightarrow{\beta} \overset{n}{:} \beta$ to say $\overrightarrow{\beta} \in Building\text{-}vectors(\beta, n)$. We note $\overrightarrow{\beta}[i] = \beta^i$ and $\overrightarrow{\beta}[:i] = \beta^1{}^\frown ...{}^\frown \beta^{i-1}$.

Definition 11.14 (n-building-vector for executions). Let e be an alternating sequence states and actions starting by state and finishing by a state if e is finite. Let $n \in \mathbb{N} \cup \{\infty\}$. A *n-building-vector* of e is a (potentially infinite) vector $\overrightarrow{e} = (e^1, ..., e^i, ...)$ of $|\overrightarrow{e}| = n$ alternating sequences of states and actions starting by state and finishing by a state (excepting potentially the last one if it is infinite) s. t. $e^1{}^\frown ... e^{i-1}{}^\frown e^i{}^\frown ... = e$ (with $\forall i \in [1, |\overrightarrow{e}| - 1], fstate(e_{i+1}) = lstate(e_i)$). We note $Building\text{-}vectors(e, n)$ the set of n-building-vector of e and $\overrightarrow{e} \overset{n}{:} e$ to say $\overrightarrow{e} \in Building\text{-}vectors(e, n)$. We note $Building\text{-}vectors(e) = \bigcup_{n \in \mathbb{N} \cup \{\infty\}} Building\text{-}vectors(e, n)$ and $\overrightarrow{e} : e$ to say $e \in Building\text{-}vectors(e)$. We note $\overrightarrow{e}[i] = e^i$ and $\overrightarrow{e}[:i] = e^1{}^\frown ...{}^\frown e^{i-1}$.

Definition 11.15 ($\overrightarrow{e} \underset{(X,\mathcal{A})}{:} e$). Let W and X be two PCA s. t. X is $\mathcal{A}$-creation-explicit, $e \in Frags(W)$. We note $\overrightarrow{e} \underset{(X,\mathcal{A})}{:} e$ (and $\overrightarrow{e} \underset{\mathcal{A}}{:} e$ when X is clear in the context) the (clearly unique) vector $\overrightarrow{e} \in Building\text{-}vectors(e)$ of execution fragments s. t.

(1) $\forall i \in [1, n], (actions(\overrightarrow{e}[i]) \setminus laction(\overrightarrow{e}[i])) \cap creation\text{-}actions(X)(\mathcal{A}) = \emptyset$ and

(2) $\forall i \in [1, n-1], laction(\overrightarrow{e}[i])) \in creation\text{-}actions(X)(\mathcal{A})$.

We write $\overrightarrow{e} \underset{(X,\mathcal{A})}{\overset{n}{:}}$ or $\overrightarrow{e} \underset{\mathcal{A}}{\overset{n}{:}}$ to indicate that $|\overrightarrow{e}| = n$.

Definition 11.16 ($\mathcal{A}$-decomposition). Let $\mathcal{A}$ be a PSIOA and X be a PCA. Let $\alpha = q^0 a^1 ... a^n q^n ... \in Frags(X)$. We say that

- α is a *$\mathcal{A}$-open-portion* iff α does not create $\mathcal{A}$, i. e. $\forall i \in [1, |\alpha|] \mathcal{A} \notin auts(config(X)(q^{i-1})) \implies \mathcal{A} \notin auts(config(X)(q^i))$.
- α is a *$\mathcal{A}$-closed-portion* iff α does not create $\mathcal{A}$ excepting at very last last action, i. e. $\forall i \in [1, |\alpha|] \mathcal{A} \notin auts(config(X)(q^{i-1})) \wedge \mathcal{A} \in auts(config(X)(q^i)) \iff i = |\alpha|$.
- α is a *$\mathcal{A}$-portion* of X if it is either a $\mathcal{A}$-open-portion or a $\mathcal{A}$-closed-portion.

We call $\mathcal{A}$-decomposition of α, note $\mathcal{A}\text{-}decomposition(\alpha)$, the unique vector $(\alpha^1, ..., \alpha^n, ...) \in Building\text{-}vectors(\alpha)$ s. t.

- $\forall i \in [1, |\mathcal{A}\text{-}decomposition(\alpha)| - 1]$, α^i is a $\mathcal{A}$-closed-portion of X and
- if $|\mathcal{A}\text{-}decomposition(\alpha)| = n \in \mathbb{N}$, α^i is a $\mathcal{A}$-portion of X.

Lemma 11.17 ($\overrightarrow{\alpha} \underset{(X,\mathcal{A})}{:} \alpha$ means $\overrightarrow{\alpha} = \mathcal{A}\text{-}decomposition(\alpha)$). *Let $\mathcal{A}$ be a PSIOA and X be a $\mathcal{A}$-creation-explicit PCA. Let $\alpha \in Frags(X)$. Let $\overrightarrow{\alpha} = \mathcal{A}\text{-}decomposition(\alpha)$. Then $\overrightarrow{\alpha} \underset{(X,\mathcal{A})}{\overset{n}{:}} \alpha$.*

Proof. By definition, $\overrightarrow{\alpha} \in \mathit{Building\text{-}vectors}(\alpha)$. Still by definition, $\forall i \in [1, |\mathcal{A}\text{-}\mathit{decomposition}(\alpha)| - 1]$, α^i is a $\mathcal{A}$-closed-portion of X, i. e. α^i does not create $\mathcal{A}$ excepting at very last last action $\mathit{laction}(\alpha_i)$. By definition of creation-explicitness, the two item of definition 11.15 are verified for every $i \in [1, |\mathcal{A}\text{-}\mathit{decomposition}(\alpha)| - 1]$. Finally, by definition, if $|\mathcal{A}\text{-}\mathit{decomposition}(\alpha)| = n \in \mathbb{N}$, α^n is a $\mathcal{A}$-portion of X, i. e. α^n does not create $\mathcal{A}$ excepting potentially at very last last action if α^n is finite. Again, by definition of creation-explicitness, the first item of definition 11.15 is verified.

□

After this sequence of definitions we can start our process of decomposition. First, we partition each class of equivalence into an aggregation several "proxy classes" that will able the application of lemma 11.12 after their decomposition.

Lemma 11.18 (Partitioning of class $\tilde{C}^{\beta,\tilde{e}}$ into classes $\hat{C}^{\overrightarrow{\beta},\overrightarrow{e}}$). *Let $\mathcal{A}$ be a PSIOA, let X be a $\mathcal{A}$-conservative PCA and $\tilde{\mathcal{E}}$ partially-compatible with X. Let β be a sequence of actions, let $\tilde{e}$ an alternating sequence of states and actions starting by a state and finishing by a state if finite and let $\tilde{\zeta} = (\beta, \tilde{e})$. Let*

- $\tilde{C}^{\tilde{\zeta}} \triangleq \mathit{Class}(\mathit{Execs}(\tilde{\mathcal{E}}||X), \mathit{print}_{(\tilde{\mathcal{E}},X)}, \tilde{\zeta}) \triangleq \{\tilde{\alpha} \in \mathit{Execs}(\tilde{\mathcal{E}}||X) | \mathit{print}_{(\tilde{\mathcal{E}},X)}(\tilde{\alpha}) = \tilde{\zeta}\}$.
- $\mathfrak{C}^{\tilde{\zeta}} = \{\hat{C}^{\overrightarrow{e},\overrightarrow{\beta}} | n \in \mathbb{N} \cup \{\infty\}, \overrightarrow{\beta} \overset{n}{\vdots} \beta, e \in \mathit{Execs}(\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}), e \upharpoonright \tilde{\mathcal{E}} = \tilde{e}, \overrightarrow{e} \overset{n}{\vdots} e\}$, *where*
- $\hat{C}^{\overrightarrow{e},\overrightarrow{\beta}} = \{\alpha | \overrightarrow{\alpha} = \mathcal{A}\text{-}\mathit{decomposition}(\alpha), \forall i \in [1, |\overrightarrow{\alpha}|], \mathit{trace}(\overrightarrow{\alpha}[i]) = \overrightarrow{\beta}[i], (\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\overrightarrow{\alpha}[i]) \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}) = \overrightarrow{e}[i]\}$

Then $\mathfrak{C}^{\tilde{\zeta}}$ is a partition of $\tilde{C}^{\tilde{\zeta}}$.

Proof. The proof is immediate by construction since the $\mathcal{A}$-decomposition is unique. We first show that $\tilde{C}^{(\beta,\tilde{e})} = \bigcup_{n\in\mathbb{N}\cup\{\infty\}} \bigcup_{\overrightarrow{\beta}\overset{n}{\vdots}\beta} \bigcup_{e\in\mathit{Execs}(\tilde{\mathcal{E}}||X\setminus\{\mathcal{A}\}), e\upharpoonright\tilde{\mathcal{E}}=\tilde{e}} \bigcup_{\overrightarrow{e}\overset{n}{\vdots}e} \hat{C}^{\overrightarrow{e},\overrightarrow{\beta}}$.

By double inclusion:

- $\subseteq$ Let $\alpha \in \tilde{C}^{(\beta,\tilde{e})}$. We note $\overrightarrow{\alpha} = \mathcal{A}\text{-}\mathit{decomposition}(\alpha)$. We note $\overrightarrow{\beta} = \mathit{trace}(\overrightarrow{\alpha})$, $\overrightarrow{\tilde{e}} = \mathit{trace}(\overrightarrow{\alpha}) \upharpoonright \tilde{\mathcal{E}}$, $\overrightarrow{e} = (\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\overrightarrow{\tilde{e}})$. By construction, we have $\overrightarrow{\beta} \overset{n}{\vdots} \beta$ and $\overrightarrow{\tilde{e}} \overset{n}{\vdots} \tilde{e}$ for some $n \in \mathbb{N}\cup\{\infty\}$. Thus $\alpha \in \bigcup_{n\in\mathbb{N}\cup\{\infty\}} \bigcup_{\overrightarrow{\beta}\overset{n}{\vdots}\beta} \bigcup_{e\in\mathit{Execs}(\tilde{\mathcal{E}}||X\setminus\{\mathcal{A}\}),e\upharpoonright\tilde{\mathcal{E}}}$

  Hence $\tilde{C}^{(\beta,\tilde{e})} \subseteq \bigcup_{n\in\mathbb{N}\cup\{\infty\}} \bigcup_{\overrightarrow{\beta}\overset{n}{\vdots}\beta} \bigcup_{e\in\mathit{Execs}(\tilde{\mathcal{E}}||X\setminus\{\mathcal{A}\}),e\upharpoonright\tilde{\mathcal{E}}=\tilde{e}} \bigcup_{\overrightarrow{e}\overset{n}{\vdots}e} \hat{C}^{\overrightarrow{e},\overrightarrow{\beta}}$.
- $\supseteq$: Let $\alpha \in \bigcup_{n\in\mathbb{N}\cup\{\infty\}} \bigcup_{\overrightarrow{\beta}\overset{n}{\vdots}\beta} \bigcup_{e\in\mathit{Execs}(\tilde{\mathcal{E}}||X\setminus\{\mathcal{A}\}),e\upharpoonright\tilde{\mathcal{E}}=\tilde{e}} \bigcup_{\overrightarrow{e}\overset{n}{\vdots}e} \hat{C}^{\overrightarrow{e},\overrightarrow{\beta}}$. By construction, $\mathit{trace}(\alpha) = \beta$ and $\alpha \upharpoonright \tilde{\mathcal{E}} = \tilde{e}$. Hence, $\bigcup_{n\in\mathbb{N}\cup\{\infty\}} \bigcup_{\overrightarrow{\beta}\overset{n}{\vdots}\beta} \bigcup_{e\in\mathit{Execs}(\tilde{\mathcal{E}}||X\setminus\{\mathcal{A}\}),e\upharpoonright\tilde{\mathcal{E}}=\tilde{e}} \bigcup_{\overrightarrow{e}\overset{n}{\vdots}e} \hat{C}^{\overrightarrow{e},\overrightarrow{\beta}} \subseteq \tilde{C}^{(\beta,\tilde{e})}$

We show that $\forall((\overrightarrow{\beta},\overrightarrow{e}),(\overrightarrow{\beta}',\overrightarrow{e}'))$ s. t. $(\overrightarrow{\beta},\overrightarrow{e}) \neq (\overrightarrow{\beta}',\overrightarrow{e}')$, $\hat{C}^{\overrightarrow{e},\overrightarrow{\beta}} \cap \hat{C}^{\overrightarrow{e}',\overrightarrow{\beta}'} = \emptyset$. Let $(\alpha,\alpha') \in \hat{C}^{\overrightarrow{e},\overrightarrow{\beta}} \times \hat{C}^{\overrightarrow{e}',\overrightarrow{\beta}'}$ and $\overrightarrow{\alpha} = \mathcal{A}\text{-}\mathit{decomposition}(\alpha)$ and $\overrightarrow{\alpha}' = \mathcal{A}\text{-}\mathit{decomposition}(\alpha')$. If $\overrightarrow{\alpha} \neq \overrightarrow{\alpha}'$, then necessarily $\alpha \neq \alpha$. We proceed by contradiction. Let assume $\overrightarrow{\alpha} = \overrightarrow{\alpha}'$. We note $n = |\overrightarrow{\alpha}| = |\overrightarrow{\alpha}'|$ with $n \in \mathbb{N} \cup \{\infty\}$. Then $\forall i \in [1,n]$, $\mathit{trace}(\overrightarrow{\alpha}[i]) = \mathit{trace}(\overrightarrow{\alpha}'[i]) = \overrightarrow{\beta}[i] = \overrightarrow{\beta}'[i]$ and $(\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\overrightarrow{\alpha}[i]) \upharpoonright \mathcal{E}_{\mathcal{A}} = (\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\overrightarrow{\alpha}[i]') \upharpoonright \mathcal{E}_{\mathcal{A}} = \overrightarrow{e}[i] = \overrightarrow{e}'[i])$ which is not possible since $(\overrightarrow{\beta},\overrightarrow{e}) \neq (\overrightarrow{\beta}',\overrightarrow{e}')$. Thus $\overrightarrow{\alpha} \neq \overrightarrow{\alpha}$ which means $\alpha \neq \alpha'$ and so $\forall((\overrightarrow{\beta},\overrightarrow{e}),(\overrightarrow{\beta}',\overrightarrow{e}'))$ s. t. $(\overrightarrow{\beta},\overrightarrow{e}) \neq (\overrightarrow{\beta}',\overrightarrow{e}')$, $\hat{C}^{\overrightarrow{e},\overrightarrow{\beta}} \cap \hat{C}^{\overrightarrow{e}',\overrightarrow{\beta}'} = \emptyset$.

□

Now we use the $\mathcal{A}$-creation explicitness to prove the lemma 11.19 and 11.20 in order to restate the previous lemma 11.18 into a simpler version to use, stated in lemma 11.21.

Lemma 11.19 (chunks ending on creation). *Let $\mathcal{A}$ be a PSIOA, let X be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit PCA and $\tilde{\mathcal{E}}$ partially-compatible with X. Let $\tilde{\alpha} \in Frags(\tilde{\mathcal{E}}||X)$ and $e \in Frags(\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\})$ s. t. $(\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A},+}(\tilde{\alpha}) \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}) = e$.*

Then

- $laction(\tilde{\alpha}) = a_! \in creation\text{-}actions(X)(\mathcal{A}) \Longrightarrow laction(e) = a_! \in creation\text{-}actions(X)(\mathcal{A})$.
- *if* $\tilde{\alpha} \in dom(\tilde{\mu}_e^{\mathcal{A},+})$,
  $laction(\tilde{\alpha}) = a_! \in creation\text{-}actions(X)(\mathcal{A}) \Longleftarrow laction(e) = a_! \in creation\text{-}actions(X)(\mathcal{A})$.

Proof. We prove the two implications in the same order.

- $\Longrightarrow$) Let assume $a_! \triangleq laction(\tilde{\alpha}) \in creation\text{-}actions(X)(\mathcal{A})$. Since X is $\mathcal{A}$-creation-explicit, we have $\tilde{\alpha} = \tilde{\alpha}' \frown q' a_! q$ with $\mathcal{A} \notin auts(config(X)(q'))$. Thus $laction(e) = a_! \in creation\text{-}actions(X)(\mathcal{A})$.
- $\Longleftarrow$) Let assume $a_! \triangleq laction(e) \in creation\text{-}actions(X)(\mathcal{A})$. Thus $a_! \in actions(\tilde{\alpha})$. Since X is $\mathcal{A}$-creation-explicit, it implies $\tilde{\alpha} = \tilde{\alpha}^1 \frown q_\ell^1, a_!, q_f^2 \frown \tilde{\alpha}^2$ where $\mathcal{A} \notin auts(config(X)(q_\ell^1))$ and $\mathcal{A} \in auts(config(X)(q_f^2))$. But $\tilde{\alpha} \in dom((\tilde{\mathcal{E}}||X).\tilde{\mu}_e^{\mathcal{A},+})$, so $\tilde{\alpha}^2 = q_f^2$ and hence $laction(\tilde{\alpha}) = a_! \in creation\text{-}actions(X)(\mathcal{A})$

□

Lemma 11.20 (common states for pasting). *Let $\mathcal{A}$ be a PSIOA, X be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit PCA and $\tilde{\mathcal{E}}$ partially-compatible with X. Let β be a sequence of actions, let e be an alternating sequence of states and actions starting by a state and finishing by a state. Let $n \in \mathbb{N} \cup \{\infty\}$, $\overrightarrow{\beta} \overset{n}{:} \beta$, $\overrightarrow{e} \overset{n}{:} e$. Let $\hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}} = \{\alpha | \overrightarrow{\alpha} = \mathcal{A}\text{-}decomposition(\alpha), \forall i \in [1, |\overrightarrow{\alpha}|], trace(\overrightarrow{\alpha}[i]) = \overrightarrow{\beta}[i], (\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\overrightarrow{\alpha}[i]) \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}) = \overrightarrow{e}[i]\}$.*

(1) *If $\hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}} \neq \emptyset$, then*
   (a) $\overrightarrow{e} \underset{(X,\mathcal{A})}{:} e$
   (b) $\forall \alpha, \alpha' \in \hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}}$, *for* $\overrightarrow{\alpha} \overset{n}{\underset{(X,\mathcal{A})}{:}} \alpha$ *and* $\overrightarrow{\alpha}' \overset{n}{\underset{(X,\mathcal{A})}{:}} \alpha', \forall i \in [1, n-1]$,
   $lstate(\overrightarrow{\alpha}[i])) = lstate(\overrightarrow{\alpha}'[i])) = fstate(\overrightarrow{\alpha}[i+1])) = fstate(\overrightarrow{\alpha}'[i+1]))$
(2) *if $\overrightarrow{e} \neq \overrightarrow{e}'$ with $\overrightarrow{e}' \underset{(X,\mathcal{A})}{:} e$, then $\hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}} = \emptyset$.*

Proof. (1) (a) Let $\alpha \in \hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}}$ and $\overrightarrow{\alpha} = \mathcal{A}\text{-}decomposition(\alpha)$. Since X is $\mathcal{A}$-creation-explicit, we have $\overrightarrow{\alpha} \underset{(X,\mathcal{A})}{:} \alpha$ by lemma 11.17. Thus, for every $i \in [1, n-1]$ we have both $\overrightarrow{\alpha}[i]$ that ends on creation of $\mathcal{A}$ and $(\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\overrightarrow{\alpha}[i]) \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}) = \overrightarrow{e}[i]$. Thus, since X is $\mathcal{A}$-creation-explicit, we can apply previous lemma 11.19 to obtain $\forall i \in [1, n-1], laction(\overrightarrow{e}[i]) \in creation\text{-}actions(X)(\mathcal{A})$. Let $i \in \mathbb{N}$ and e_i' a strict prefix of $\overrightarrow{e}[i]$. By construction of $\mu_e^{\mathcal{A},+}$, it exists a strict prefix of $\overrightarrow{\alpha}[i]$, noted α_i', s. t. $(\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\alpha_i') \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}) = e_i'$. By definition of $\overrightarrow{\alpha} \underset{(X,\mathcal{A})}{:} \alpha, laction(\alpha_i') \notin creation\text{-}actions(X)(\mathcal{A})$. By contraposition of lemma 11.19, $laction(e_i') \notin creation\text{-}actions(X)(\mathcal{A})$. Since the results holds for any strict prefix, $\overrightarrow{e} \underset{(X,\mathcal{A})}{:} e$.

(b) Let $\alpha, \alpha' \in \hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}}$, Let $\overrightarrow{\alpha} = \mathcal{A}\text{-}decomposition(\alpha)$, $\overrightarrow{\alpha}' = \mathcal{A}\text{-}decomposition(\alpha')$, let $|\overrightarrow{\alpha}| = |\overrightarrow{\alpha}'| = n$ and let $i \in [1, n-1]$. By construction we have directly $lstate(\overrightarrow{\alpha}[i]) = fstate(\overrightarrow{\alpha}[i+1])$ and $lstate(\overrightarrow{\alpha}'[i]) =$

$fstate(\overrightarrow{\alpha}'[i+1])$. Moreover, we have both (1) $(\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\overrightarrow{\alpha}[i]) \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}) = (\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\overrightarrow{\alpha}'[i]) \upharpoonright (\tilde{\mathcal{E}}||X\setminus\{\mathcal{A}\}) = \overrightarrow{e}[i]$ and (2) $map(config(X)(lstate(\overrightarrow{\alpha}[i])))(\mathcal{A}) = map(config(X)(lstate(\overrightarrow{\alpha}'[i])))(\mathcal{A}) = \bar{q}_{\mathcal{A}}$. The property (1) implies (3) $auts(config(X)(lstate(\overrightarrow{\alpha}[i])))\setminus\{\mathcal{A}\} = auts(config(X)(lstate(\overrightarrow{\alpha}'[i])))\setminus\{\mathcal{A}\} \triangleq \mathbf{A}'$ and $\forall \mathcal{B} \in \mathbf{A}', map(config(X)(lstate(\overrightarrow{\alpha}[i])))(\mathcal{B}) = map(config(X)(lstate(\overrightarrow{\alpha}'[i])))(\mathcal{B})$. Thus by (2) and (3) we obtain (4) $auts(config(X)(lstate(\overrightarrow{\alpha}[i]))) = auts(config(X)(lstate(\overrightarrow{\alpha}'[i]))) \triangleq \mathbf{A}$ and $\forall \mathcal{B} \in \mathbf{A}, map(config(X)(lstate(\overrightarrow{\alpha}[i])))(\mathcal{B}) = map(config(X)(lstate(\overrightarrow{\alpha}'[i])))(\mathcal{B})$. Since X is configuration-conflict-free, $lstate(\overrightarrow{\alpha}[i])) = lstate(\overrightarrow{\alpha}'[i]))$ which ends the proof.

(2) By contraposition of former item 1a.

□

Now we can simplify lemma 11.17. We can already guess in $\hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}}$ the form of "proxy aggregated class" introduced in definition 11.10.

Lemma 11.21 (Partitioning of class $\tilde{\mathcal{C}}^{\beta,\tilde{e}}$ into classes $\hat{\mathcal{C}}^{\overrightarrow{\beta},\overrightarrow{e}}$ ending on $\mathcal{A}$ creation). *Let $\mathcal{A}$ be a PSIOA, let X be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit PCA and $\tilde{\mathcal{E}}$ partially-compatible with X. Let $\mathcal{E}_{\mathcal{A}} = \tilde{\mathcal{E}}||(X \setminus \{\mathcal{A}\})$. Let β be a sequence of actions, let $\tilde{e}$ an alternating sequence of states and actions starting by a state and finishing by a state if finite and let $\tilde{\zeta} = (\beta, \tilde{e})$. Let*

- $\tilde{\mathcal{C}}^{\tilde{\zeta}} \triangleq Class(Execs(\tilde{\mathcal{E}}||X), print_{(\tilde{\mathcal{E}},X)}, \tilde{\zeta}) \triangleq \{\tilde{\alpha} \in Execs(\tilde{\mathcal{E}}||X) | print_{(\tilde{\mathcal{E}},X)}(\tilde{\alpha}) = \tilde{\zeta}\}$.
- $\tilde{\mathfrak{C}}^{\tilde{\zeta}}_{CrEx} \triangleq \{\hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}} | \exists e \in Execs(\mathcal{E}_{\mathcal{A}}), e \upharpoonright \tilde{\mathcal{E}} = \tilde{e}, \overrightarrow{e} \overset{n}{\underset{(X,\mathcal{A})}{\vdots}} e, \overrightarrow{\beta} \overset{n}{\vdots} \beta\}$ *where*
- $\hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}} = \{\alpha \in Execs(\tilde{\mathcal{E}}||X) | \overrightarrow{\alpha} \underset{(X,\mathcal{A})}{:} \alpha, |\overrightarrow{\alpha}| = |\overrightarrow{e}| = |\overrightarrow{\beta}|, \forall i \in [1, |\overrightarrow{\alpha}|], trace(\overrightarrow{\alpha}[i]) = \overrightarrow{\beta}[i] \wedge (\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\overrightarrow{\alpha}[i]) \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}) = \overrightarrow{e}[i]\}$

Then, $\tilde{\mathfrak{C}}^{\tilde{\zeta}}_{CrEx}$ is a partition of $\tilde{\mathcal{C}}^{\zeta}$.

Proof. By conjunction of lemma 11.17, lemma 11.18 and lemma 11.20, item 1a. □

In next paragraph, we will isolate "proxy aggregated class without creation" introduced in definition 11.10.

Decomposition. We start this paragraph, by showing that $\hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}}$ is equal to the set of concatenated executions issued to some fixed "proxy aggregated classes without creation" introduced in definition 11.10.

Lemma 11.22 (decomposition into simple classes). *Let $\mathcal{A}$ be a PSIOA, X be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit a PCA and $\tilde{\mathcal{E}}$ partially-compatible with X. Let $\mathcal{E}_{\mathcal{A}} = \tilde{\mathcal{E}}||(X \setminus \{\mathcal{A}\})$. Let β be a sequence of actions, let $e \in Execs(\mathcal{E}_{\mathcal{A}})$. Let $n \in \mathbb{N} \cup \{\infty\}$, $\overrightarrow{\beta} \overset{n}{\vdots} \beta$, $\overrightarrow{e} \overset{n}{\underset{(X,\mathcal{A})}{\vdots}} e$. Let*

$\hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}} = \{\alpha \in Execs(\tilde{\mathcal{E}}||X) | \overrightarrow{\alpha} \underset{(X,\mathcal{A})}{:} \alpha, \forall i \in [1, |\overrightarrow{\alpha}|], trace(\overrightarrow{\alpha}[i]) = \overrightarrow{\beta}[i], (\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\overrightarrow{\alpha}[i]) \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}) = \overrightarrow{e}[i]\}$.

Then, $\hat{\mathcal{C}}^{\overrightarrow{e},\overrightarrow{\beta}} = \bigotimes_i^n \hat{\mathcal{C}}^{\overrightarrow{e}[i],\overrightarrow{\beta}[i],+}$ with

- $\hat{\mathcal{C}}^{e^i,\beta^i,+} = \{\alpha^i \in Execs((\tilde{\mathcal{E}}^i||X^i)) \cap dom((\tilde{\mathcal{E}}^i||X^i).\tilde{\mu}_e^{\mathcal{A},+}) \, | trace(\alpha^i) = \beta^i, (\tilde{\mathcal{E}}^i||X^i).\tilde{\mu}_e^{\mathcal{A},+}(\alpha^i) \upharpoonright (\tilde{\mathcal{E}}^i||X^i \setminus \{\mathcal{A}\}) = e_i\}$
- $\tilde{\mathcal{E}}^1 = \tilde{\mathcal{E}}$ *and* $\forall i \in [2,n], \tilde{\mathcal{E}}^i = \tilde{\mathcal{E}}_{\bar{q}_{\mathcal{E}} \to q^i_{\mathcal{E}}}$ *with* $q^i_{\mathcal{E}} = fstate(e_i) \upharpoonright \tilde{\mathcal{E}}$, *as per definition 9.9*

- $X^1 = X$ *and* $\forall i \in [2,n]$, $X^i = X_{\bar{q}_X \to q_X^i}$ *(as per definition 9.9) with* q_X^i *the unique state (by configuration-conflict-free property) s. t.* $config(X)(q_X^i) = C_Y^i \cup C_{\mathcal{A}}^i$ *with* $C_Y^i = config(X \setminus \{\mathcal{A}\})(fstate(e_i) \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}))$ *and* $C_{\mathcal{A}}^i = (\{\mathcal{A}\}, (\mathcal{A}, \bar{q}_{\mathcal{A}}))$.
- $\bigotimes_i^n C^i = C^1 \otimes C^2 \otimes ... \otimes C^n$
- $C^1 \otimes C^2 = \{\alpha_1^\frown \alpha_2 | \alpha_1 \in C^1, \alpha_2 \in C^2\}$

Proof. By double inclusion.

- $\subseteq$) Let $\alpha \in \hat{\mathcal{C}}^{\vec{e},\vec{\beta}}$, i. e. $\vec{\alpha} = \mathcal{A}\text{-}decomposition(\alpha)$ and $\forall i \in [1, |\vec{\alpha}|], trace(\vec{\alpha}[i]) = \vec{\beta}[i], (\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A}}(\vec{\alpha}[i]) \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}) = \vec{e}[i]$. We need to show that $\forall i \in [1, |\vec{\alpha}|]$, $\vec{\alpha}[i] \in \hat{\mathcal{C}}^{\vec{e}[i],\vec{\beta}[i],+}$. By construction due to $\mathcal{A}$-decomposition, $\forall i \in [2,n]$, $fstate(\vec{\alpha}[i]) = lstate(\vec{\alpha}[i-1])$ where $\vec{\alpha}[i-1]$ ends on $\mathcal{A}$-creation (1). Moreover, $\forall i \in [1,n]$, $(\tilde{\mathcal{E}}||X).\mu_e^{\mathcal{A},+}(\vec{\alpha}[i]) \upharpoonright (\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}) = \vec{e}[i]$ (2). By (1) and (2), $fstate(\vec{\alpha}[i]) = start(\tilde{\mathcal{E}}^i||X^i)$ where $\tilde{\mathcal{E}}^i$ and X^i are defined like in the lemma (3). By construction due to $\mathcal{A}$-decomposition, $\vec{\alpha}[i]$ does not create $\mathcal{A}$ before its very last action (4), thus by (3) and (4) $\vec{\alpha}[i] \in dom((\tilde{\mathcal{E}}^i||X^i).\tilde{\mu}_e^{\mathcal{A},+})$ (5). Then $\hat{\mathcal{C}}^{\vec{e},\vec{\beta}} \subseteq \bigotimes_i^n \hat{\mathcal{C}}^{\vec{e}[i],\vec{\beta}[i],+}$
- $\supseteq$) Let $(\alpha_i)_{i \in [1,n]}$ s. t. $\forall i \in [1,n], \alpha_i \in \hat{\mathcal{C}}^{\vec{e}[i],\vec{\beta}[i],+}$. We note $\vec{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_i, ...)$ with $|\vec{\alpha}| = n$. By construction, $\forall i \in [1, n-1]$, $lstate(\vec{e}[i]) = fstate(\vec{e}[i+1])$ (6). Moreover $\forall i \in [1, n-1]$, $laction(\vec{e}[i]) \in creation\text{-}actions(X)(\mathcal{A})$ so $\forall i \in [1, n-1], laction(\alpha_i) \in creation\text{-}actions(X)(\mathcal{A})$ (7) by lemma 11.19. Thus, $\forall i \in [1, n-1]$, $map(config(\tilde{\mathcal{E}}||X)(lstate(\alpha_i)))(\mathcal{A})) = start(\mathcal{A})$ (8). Since $\forall i \in [1,n]$, $(\tilde{\mathcal{E}}^i||X^i).\tilde{\mu}_e^{\mathcal{A},+}(\alpha_i) \upharpoonright (\tilde{\mathcal{E}}^i||X^i \setminus \{\mathcal{A}\}) = \vec{e}[i]$, by (6), (8) and configuration-conflict-free property, we deduce that $\forall i \in [1, n-1], lstate(\alpha_i) = fstate(\alpha_{i+1})$ (9). Now, we note $\alpha = \alpha^1{}^\frown \alpha^2{}^\frown ... \alpha^i{}^\frown$ s. t. $\vec{\alpha} \overset{\vec{\beta}}{:} \alpha$. By assumption, $\forall i \in [1,n]$, $\vec{\alpha}[i] \in dom((\tilde{\mathcal{E}}||X).\tilde{\mu}_e^{\mathcal{A},+})$, i. e. $\mathcal{A}$ is not created before last action of α_i which means $\vec{\alpha} = \mathcal{A}\text{-}decomposition(\alpha)$ (10). Thus $\alpha \in \hat{\mathcal{C}}^{\vec{e},\vec{\beta}}$. Hence, $\bigotimes_i^n \hat{\mathcal{C}}^{\vec{e}[i],\vec{\beta}[i],+} \subseteq \hat{\mathcal{C}}^{\vec{e},\vec{\beta}}$.

□

Now we can reduce the measure of the entire class of external perception into measures of some fixed "proxy aggregated classes without creation" introduced in definition 11.10 to eventually apply the lemma 11.12. We start by an immediate summation in lemma 11.23 before a slightly more subtle product in lemma 11.24

Lemma 11.23 (measure after partitioning and decomposition). *Let $\mathcal{A}$ be a PSIOA, X be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit PCA and $\tilde{\mathcal{E}}$ partially-compatible with X. Let $\mathcal{E}_{\mathcal{A}} = \tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\}$ Let β be a sequence of actions, let $\tilde{e}$ be an alternating sequence of states and actions starting by a state and finishing by a state. Let $\tilde{\sigma} \in schedulers(\tilde{\mathcal{E}}||X)$.*

$\epsilon_{\tilde{\sigma}}(C_{\hat{\mathcal{C}}^{\beta,\tilde{e}}}) = \sum_{e \in Execs(\mathcal{E}_{\mathcal{A}}), e \upharpoonright \tilde{\mathcal{E}} = \tilde{e}, \vec{e} \overset{n}{\underset{\mathcal{A}}{:}} e, \sum_{\vec{\beta}:_\beta^n \beta} \epsilon_{\tilde{\sigma}}(C_{\bigotimes_i^n \hat{\mathcal{C}}^{\vec{e}[i],\vec{\beta}[i],+}})$.

Proof. Immediate by lemma 11.18 and 11.22 □

Now we want to transform the term $\epsilon_{\tilde{\sigma}}(C_{\bigotimes_i^n \hat{\mathcal{C}}^{\vec{e}[i],\vec{\beta}[i],+}})$ as a function of some terms $\epsilon_{\tilde{\sigma}^i}(C_{\hat{\mathcal{C}}^{\vec{e}[i],\vec{\beta}[i],+}})$ where $\tilde{\sigma}^i$ must be defined. The critical point is that the occurrence of these events might not be independent with (*) a perfect-information scheduler that chooses the measure of class $\hat{\mathcal{C}}^{\vec{e}[i],\vec{\beta}[i],+}$ as a function of the concrete prefix in class $\hat{\mathcal{C}}^{\vec{e}[i-1],\vec{\beta}[i-1],+}$. This observation enforced us to weaken the implementation definition to make it monotonic w.r.t. PSIOA creation by handling only creation-oblivious schedulers that cannot make the choice (*).

Lemma 11.24 (measure after decomposition for oblivious creation scheduler). *Let $\mathcal{A}$ be a PSIOA, X be a $\mathcal{A}$-conservative, $\mathcal{A}$-creation-explicit PCA and $\tilde{\mathcal{E}}$ partially-compatible with X.*

Let $\overrightarrow{\beta}$ be a vector of sequences of actions, let $e \in Execs(\tilde{\mathcal{E}}||X \setminus \{\mathcal{A}\})$, and $\overrightarrow{e} \underset{(X,\mathcal{A})}{\overset{n}{\vdots}} e$. Let $\tilde{\sigma} \in schedulers(\tilde{\mathcal{E}}||X)$ that is $\mathcal{A}$-creation-oblivious.

Then $\epsilon_{\tilde{\sigma}}(C_{\bigotimes_i^n \hat{C}^{\overrightarrow{e}[i],\overrightarrow{\beta}[i],+}}) = \prod_i^n \epsilon_{\tilde{\sigma}^i}(C_{C^{\overrightarrow{\beta}[i],\overrightarrow{e}[i]}})$ with $\forall i \in [1,n], \tilde{\sigma}^i = oblivious_{\mathcal{A},\overrightarrow{\beta}[:i],\overrightarrow{e}[:i]}(\tilde{\sigma})$.

Proof. We recall the remark of definition 11.3 of $\mathcal{A}$-creation-oblivious scheduler for a $\mathcal{A}$-conservative PCA that raises the fact that if an execution fragment $\tilde{\alpha} \in Frags^*(\tilde{W})$ verifying i) $\tilde{\alpha}$ ends on $\mathcal{A}$-creation, ii) $\tilde{W}.\mu_e^{\mathcal{A},+}(\tilde{\alpha}) = e$ and iii) $trace(\tilde{\alpha}) = \beta$ exists, then $\tilde{\sigma}_{|\mathcal{A},\beta,e} = \tilde{\sigma}_{|\tilde{\alpha}}$, the sub-scheduler conditioned by $\tilde{\sigma}$ and $\tilde{\alpha}$ in the sense of definition 11.4. Then we simply apply lemma 11.5, which states that for every $\alpha = \alpha_x^\frown \alpha_y \in Frags^*(\tilde{\mathcal{E}}||X)$, for $\tilde{\sigma}_{|\alpha_x}$ the sub-scheduler conditioned by $\tilde{\sigma} \in schedulers(\tilde{\mathcal{E}}||X)$ and α_x (in the sense of definition 11.4), for $\epsilon_{\tilde{\sigma}}$ generated by $\tilde{\sigma}$, $\epsilon_{\tilde{\sigma}}(C_\alpha) = \epsilon_{\tilde{\sigma}}(C_{\alpha_x}) \cdot \epsilon_{\tilde{\sigma}_{|\alpha_x}}(C_{\alpha_y})$ with $\tilde{\sigma}_{|\alpha_x}(\alpha_z) = \tilde{\sigma}(\alpha_x^\frown \alpha_z)$ for every α_z with $fstate(\alpha_z) = lstate(\alpha_x)$.

For every $\alpha \in \bigotimes_i^n \hat{C}^{\overrightarrow{e}[i],\overrightarrow{\beta}[i],+}$, for $\overrightarrow{\alpha} =$ $\mathcal{A}$-decomposition, $\epsilon_{\tilde{\sigma}}(C_\alpha) = \prod_i^n \epsilon_{\tilde{\sigma}_{|\overrightarrow{\alpha}[1:i-1]},q_i}(C_{\overrightarrow{\alpha}[i]})$ with $\overrightarrow{\alpha}[1:i-1] = \alpha^1{}^\frown \ldots {}^\frown \alpha^{i-1}$ and $q_i = start(\tilde{\mathcal{E}}^i||X^i)$ where $\tilde{\mathcal{E}}^i||X^i$ is defined as in lemma 11.22.

By $\mathcal{A}$-creation-oblivious property of $\tilde{\sigma}$, for every pair of vectors $\overrightarrow{\alpha}[1:i-1], \overrightarrow{\alpha}'[1:i-1]$ s. t. $\forall j \in [1:i-1]$, $trace(\overrightarrow{\alpha}[j]) = trace(\overrightarrow{\alpha}'[j]) = \beta_j$ and $(\tilde{\mathcal{E}}^j||X^j).\tilde{\mu}_e^{\mathcal{A},+}(\overrightarrow{\alpha}[j]) \upharpoonright (\tilde{\mathcal{E}}^j||X^j) = (\tilde{\mathcal{E}}^j||X^j).\tilde{\mu}_e^{\mathcal{A},+}(\overrightarrow{\alpha}'[j]) \upharpoonright (\tilde{\mathcal{E}}^j||X^j) = e_j$, for every $\pi \in dom((\tilde{\mathcal{E}}^i||X^i).\tilde{\mu}_e^{\mathcal{A},+})$, $\tilde{\sigma}_{|\overrightarrow{\alpha}[1:i-1]}(\pi) = \tilde{\sigma}_{|\overrightarrow{\alpha}'[1:i-1]}$. Hence, for every $i \in [1,n]$ we note $\tilde{\sigma}^i \in schedulers(\tilde{\mathcal{E}}^i||X^i)$ that matches $\tilde{\sigma}_{|\overrightarrow{\alpha}[1:i-1]}$ on $dom((\tilde{\mathcal{E}}^i||X^i).\tilde{\mu}_e^{\mathcal{A},+})$ for an arbitrary $\overrightarrow{\alpha}[1:i-1]$. This lead us to: $\forall \alpha \in \bigotimes_i^n \hat{C}^{\overrightarrow{e}[i],\overrightarrow{\beta}[i],+}$, for $\overrightarrow{\alpha} \underset{(X,\mathcal{A})}{\overset{:}{\alpha}}$, $\epsilon_{\tilde{\sigma}}(C_\alpha) = \prod_i^n \epsilon_{\tilde{\sigma}^i}(C_{\overrightarrow{\alpha}[i]})$

Thus $\epsilon_{\tilde{\sigma}}(C_{\bigotimes_i^n \hat{C}^{\overrightarrow{e}[i],\overrightarrow{\beta}[i],+}}) = \sum_{\overrightarrow{\alpha} \underset{(X,\mathcal{A})}{\overset{:}{\alpha}}, \ \alpha \in \bigotimes_i^n \hat{C}^{\overrightarrow{e}[i],\overrightarrow{\beta}[i],+}} \prod_i^n \epsilon_{\tilde{\sigma}^i}(C_{\overrightarrow{\alpha}[i]})$ and by lemma 11.22,

$\epsilon_{\tilde{\sigma}}(C_{\bigotimes_i^n \hat{C}^{\overrightarrow{e}[i],\overrightarrow{\beta}[i],+}}) = \sum_{\alpha_1 \in C^{\overrightarrow{e}[1],\overrightarrow{\beta}[1],+}} \ldots \sum_{\alpha_n \in C^{\overrightarrow{e}[i],\overrightarrow{\beta}[i],+}} \prod_i^n \epsilon_{\tilde{\sigma}^i}(C_{\alpha_i}) = \prod_i^n \epsilon_{\tilde{\sigma}^i}(C_{C^{\overrightarrow{\beta}[i],\overrightarrow{e}[i]},+})$

□

Now thanks to lemma 11.23, 11.24 and 11.12 we are ready to prove the main theorem of this paper: the monotonicity of implementation w.r.t. PSIOA destruction/creation.

Theorem 11.25 (monotonicity). *Let $\mathcal{A}$ and $\mathcal{B}$ be two PSIOA, let $X_\mathcal{A}$ be a $\mathcal{A}$-conservative and $\mathcal{A}$-creation-explicit PCA, let $X_\mathcal{B}$ be a $\mathcal{B}$-conservative and $\mathcal{B}$-creation-explicit PCA, s. t. $X_\mathcal{A}$ and $X_\mathcal{B}$ are corresponding w. r. t. $\mathcal{A}, \mathcal{B}$ with $creation\text{-}actions(X_\mathcal{A})(\mathcal{A}) = creation\text{-}actions(X_\mathcal{B})(\mathcal{B}) \triangleq CrActs$.*

If $\mathcal{A} \leq^{print}_{CrOb} \mathcal{B}$, then $X_\mathcal{A} \leq^{print}_{CrOb} X_\mathcal{B}$.

Proof. Let $\tilde{\mathcal{E}} \in env(X_\mathcal{A}) \cap env(X_\mathcal{B})$. Let $Y_\mathcal{A} = X_\mathcal{A} \setminus \{\mathcal{A}\}$, $Y_\mathcal{B} = X_\mathcal{B} \setminus \{\mathcal{B}\}$, $\mathcal{E}_\mathcal{A} = \tilde{\mathcal{E}}||Y_\mathcal{A}$, $\mathcal{E}_\mathcal{B} = \tilde{\mathcal{E}}||Y_\mathcal{B}$ and $\mathcal{E}$ an arbitrary PCA semantically equivalent to both $\mathcal{E}_\mathcal{A}$ and $\mathcal{E}_\mathcal{B}$ with $\mathcal{E} \in env(\tilde{\mathcal{A}}^{sw}) \cap env(\tilde{\mathcal{B}}^{sw})$ by theorem 10.13. We note $\mu_{\mathcal{A}C}$ the (complete, strong and bijective) PCA executions-matching from $\mathcal{E}_\mathcal{A}$ to $\mathcal{E}$ and $\mu_{C\mathcal{B}}$ the (complete, strong and bijective) PCA executions-matching from $\mathcal{E}$ to $\mathcal{E}_\mathcal{B}$. We also note $\mu^\times_{\mathcal{A}C}$ the (complete, strong and bijective) PCA executions-matching from $\mathcal{E}_\mathcal{A}||\tilde{A}^{sw}$ to $\mathcal{E}||\tilde{A}^{sw}$ and $\mu^\times_{C\mathcal{B}}$ the (complete, strong and bijective) PCA executions-matching from $\mathcal{E}||\tilde{B}^{sw}$ to $\mathcal{E}_\mathcal{B}||\tilde{B}^{sw}$.

In the remaining we note $(\tilde{\mathcal{E}}||X_{\mathcal{A}})^{\downarrow e}$ the automaton $(\tilde{\mathcal{E}}||X_{\mathcal{A}})_{\bar{q}_{(\tilde{\mathcal{E}}||X_{\mathcal{A}})}\rightarrow q}$ (as per definition 9.9) where q is the unique state of $\tilde{\mathcal{E}}||X_{\mathcal{A}}$ s. t. $map(config(\tilde{\mathcal{E}}||X_{\mathcal{A}})(q))(\mathcal{A}) = \bar{q}_{\mathcal{A}}$ and $config(\tilde{\mathcal{E}}||X_{\mathcal{A}})(q) \setminus \{\mathcal{A}\} = config(\mathcal{E}_{\mathcal{A}})(lstate(e))$. Respectively, we note $(\tilde{\mathcal{E}}||X_{\mathcal{B}})^{\downarrow e}$ the automaton $(\tilde{\mathcal{E}}||X_{\mathcal{B}})_{\bar{q}_{(\tilde{\mathcal{E}}||X_{\mathcal{B}})}\rightarrow q}$ (as per definition 9.9) where q is the unique state of $\tilde{\mathcal{E}}||X_{\mathcal{B}}$ s. t. $map(config(\tilde{\mathcal{E}}||X_{\mathcal{B}})(q))(\mathcal{A}) = \bar{q}_{\mathcal{B}}$ and $config(\tilde{\mathcal{E}}||X_{\mathcal{B}})(q) \setminus \{\mathcal{B}\} = config(\mathcal{E}_{\mathcal{B}})(lstate(e))$. Finally, we note $\tilde{\mathcal{E}}^e = \tilde{\mathcal{E}}_{\bar{q}_{\mathcal{E}}\rightarrow lstate(e)}$.

Let $\tilde{\sigma} \in CrOB(\tilde{\mathcal{E}}||X_{\mathcal{A}})$. We need to show it exists $\tilde{\sigma}' \in CrOB(\tilde{\mathcal{E}}||X_{\mathcal{B}})$ s. t. for every sequence of actions β, for every $\tilde{e} \in Execs(\tilde{\mathcal{E}})$, for $\tilde{\zeta} = (\beta, \tilde{e})$, $\epsilon_{\tilde{\sigma}}(C_{\tilde{C}^{\tilde{\zeta}}_{X_{\mathcal{A}}}}) = \epsilon_{\tilde{\sigma}'}(C_{\tilde{C}^{\tilde{\zeta}}_{X_{\mathcal{B}}}})$ where $\tilde{C}^{\tilde{\zeta}}_{X_{\mathcal{A}}} = Class(Execs(\tilde{\mathcal{E}}||X_{\mathcal{A}}), print_{(\tilde{\mathcal{E}},X_{\mathcal{A}})}, \tilde{\zeta})$ and $\tilde{C}^{\tilde{\zeta}}_{X_{\mathcal{B}}} = Class(Execs(\tilde{\mathcal{E}}||X_{\mathcal{B}}), print_{(\tilde{\mathcal{E}},X_{\mathcal{B}})}, \tilde{\zeta})$.

$\forall \beta \in trace(\tilde{\mathcal{E}}||X_{\mathcal{A}}) \cup trace(\tilde{\mathcal{E}}||X_{\mathcal{B}})$, $\forall e^a \in Execs(\mathcal{E}_{\mathcal{A}})$, we note $\sigma_{|\mathcal{A},\beta,e^a}$ the $((\tilde{\mathcal{E}}||X_{\mathcal{A}})^{\downarrow e^a}).\tilde{\mu}^{\mathcal{A},+}_e$ alter-ego of $\tilde{\sigma}_{|\mathcal{A},\beta,e^a}$. Let $e = \mu_{\mathcal{A}C}(e^a)$ We note $\sigma^c_{|\mathcal{A},\beta,e} \in schedulers(\mathcal{E}^e||\tilde{\mathcal{A}}^{sw})$ the $\mu^{\times}_{\mathcal{A}C}$ alter-ego of $\sigma_{|\mathcal{A},\beta,e^a}$

(*) Since $\mathcal{A} \leq_{CrOb} \mathcal{B}$, it exists $\sigma^d_{|\mathcal{B},\beta,e} \in scheduler(\mathcal{E}^e||\tilde{\mathcal{B}}^{sw})$ balanced with $\sigma^c_{|\mathcal{A},\beta,e}$, i. e. for every sequence of actions β', for every $e' \in Execs(\mathcal{E}^e)$, $\sigma^c_{|\mathcal{A},\beta,e}(C_{\check{C}^{(\beta',e')}_{\mathcal{A}}}) = \sigma^d_{|\mathcal{B},\beta,e}(C_{\check{C}^{(\beta',e')}_{\mathcal{B}}})$ where:

- $\check{C}^{(\beta',e')}_{(\mathcal{E},\mathcal{A})} = Class(Execs(\mathcal{E}^e||\tilde{\mathcal{A}}^{sw}), print_{(\mathcal{E}^e,\tilde{\mathcal{A}}^{sw})}, (\beta', e'))$ and
- $\check{C}^{(\beta',e')}_{(\mathcal{E},\mathcal{B})} = Class(Execs(\mathcal{E}^e||\tilde{\mathcal{B}}^{sw}), print_{(\mathcal{E}^e,\tilde{\mathcal{B}}^{sw})}, (\beta', e'))$

.

Let $e^b = \mu_{C\mathcal{B}}(e)$ We note $\sigma'_{|\mathcal{B},\beta,e^b}$ the $\mu^{\times}_{C\mathcal{B}}$ alter-ego of $\sigma^d_{|\mathcal{B},\beta,e}$.

We build $\tilde{\sigma}' \in CrOB(\tilde{\mathcal{E}}||X_{\mathcal{B}})$ as follows:

$\forall \beta \in trace(\tilde{\mathcal{E}}||X_{\mathcal{A}}) \cup trace(\tilde{\mathcal{E}}||X_{\mathcal{B}})$, $\forall e^b \in Execs(\mathcal{E}_{\mathcal{B}})$, $\forall \tilde{\pi}, \tilde{\pi}' \in Frags^*(\tilde{\mathcal{E}}||X_{\mathcal{B}})$ s.t. i) $\tilde{\pi}$ ends on $\mathcal{B}$ creation, ii) $(\tilde{\mathcal{E}}||X_{\mathcal{B}}).\tilde{\mu}^{\mathcal{A}}_e(\tilde{\pi}) \upharpoonright \mathcal{E}_{\mathcal{B}} = e^b$, iii) $trace(\tilde{\pi}) = \beta$, iv) $lstate(\tilde{\pi}) = fstate(\tilde{\pi}')$, $\tilde{\sigma}'(\tilde{\pi}^\frown\tilde{\pi}') = \tilde{\sigma}'_{|\mathcal{B},\beta,e^b}(\tilde{\pi}')$ s. t. $\tilde{\sigma}'_{|\mathcal{B},\beta,e^b}$ and $\sigma'_{|\mathcal{B},\beta,e^b}$ are $((\tilde{\mathcal{E}}||X_{\mathcal{B}})^{e^b}).\tilde{\mu}^{\mathcal{B},+}_e$ alter-ego.

Now we show that $\tilde{\sigma}$ and $\tilde{\sigma}'$ are balanced:

Let $\tilde{\zeta} = (\beta, \tilde{e}) \in print(\tilde{\mathcal{E}}, X_{\mathcal{A}}) \cup print(\tilde{\mathcal{E}}, X_{\mathcal{B}})$, i. e. $\beta \in trace(\tilde{\mathcal{E}}||X_{\mathcal{A}}) \cup trace(\tilde{\mathcal{E}}||X_{\mathcal{B}})$, and $\tilde{e} \in Execs(\tilde{\mathcal{E}})$. Let

- $\tilde{C}^{(\beta,\tilde{e})}_{\mathcal{A}} = Class(Execs(\tilde{\mathcal{E}}||X_{\mathcal{A}}), print_{(\tilde{\mathcal{E}},X_{\mathcal{A}})}, (\beta, \tilde{e}))$ and
- $\tilde{C}^{(\beta,\tilde{e})}_{\mathcal{B}} = Class(Execs(\tilde{\mathcal{E}}||X_{\mathcal{B}}), print_{(\tilde{\mathcal{E}},X_{\mathcal{A}})}, (\beta, \tilde{e}))$

.

We need to show that $\epsilon_{\tilde{\sigma}}(C_{\tilde{C}^{(\beta,\tilde{e})}_{\mathcal{A}}}) = \epsilon_{\tilde{\sigma}'}(C_{\tilde{C}^{(\beta,\tilde{e})}_{\mathcal{B}}})$:

We apply lemma 11.23 to obtain:

- $\epsilon_{\tilde{\sigma}}(C_{\tilde{C}^{\beta,\tilde{e}}_{\mathcal{A}}}) = \sum_{e^a \in Execs(\mathcal{E}_{\mathcal{A}}), e^a \upharpoonright \tilde{\mathcal{E}} = \tilde{e}} \sum_{\vec{e}^a \overset{n}{\underset{\mathcal{A}}{:}} e^a} \sum_{\vec{\beta} \overset{n}{:} \beta} \epsilon_{\tilde{\sigma}}(C_{\bigotimes_i \hat{C}^{\vec{e}^a[i],\vec{\beta}[i],+}_{\mathcal{A}}})$.
- $\epsilon_{\tilde{\sigma}}(C_{\tilde{C}^{\beta,\tilde{e}}_{\mathcal{B}}}) = \sum_{e^b \in Execs(\mathcal{E}_{\mathcal{B}}), e^b \upharpoonright \tilde{\mathcal{E}} = \tilde{e}} \sum_{\vec{e}^b \overset{n}{\underset{\mathcal{A}}{:}} e^b} \sum_{\vec{\beta} \overset{n}{:} \beta} \epsilon_{\tilde{\sigma}'}(C_{\bigotimes_i \hat{C}^{\vec{e}^b[i],\vec{\beta}[i],+}_{\mathcal{B}}})$.

Since $\mathcal{E}_{\mathcal{A}}$ and $\mathcal{E}_{\mathcal{B}}$ are semantically equivalent, the sets $\{e^a \in Execs(\mathcal{E}_{\mathcal{A}}) | e^a \upharpoonright \tilde{\mathcal{E}} = \tilde{e}\}$ and $\{e^b \in Execs(\mathcal{E}_{\mathcal{B}}) | e^b \upharpoonright \tilde{\mathcal{E}} = \tilde{e}\}$ are in bijection. Hence, it is enough to show that $\forall (e^{ac}, e^{bc}) \in Execs(\mathcal{E}_{\mathcal{A}}) \times Execs(\mathcal{E}_{\mathcal{B}})$ with $e^{bc} = \mu_{\mathcal{A}C} \circ \mu_{C\mathcal{B}}(e^{ac})$ and $e^{bc} \upharpoonright \tilde{\mathcal{E}} = e^{ac} \upharpoonright \tilde{\mathcal{E}} = \tilde{e}$, for $\vec{e}^{ac} \overset{n}{\underset{\mathcal{A}}{:}} e^{ac}$, $\vec{e}^{bc} \overset{n}{\underset{\mathcal{A}}{:}} e^{bc}$, for every $\vec{\beta} \overset{n}{:} \beta$, then $\epsilon_{\tilde{\sigma}}(C_{\bigotimes_i \hat{C}^{\vec{e}^{ac}[i],\vec{\beta}[i],+}_{\mathcal{A}}}) = \epsilon_{\tilde{\sigma}'}(C_{\bigotimes_i \hat{C}^{\vec{e}^{bc}[i],\vec{\beta}[i],+}_{\mathcal{B}}})$.

By definition, $\tilde{\sigma}$ is $\mathcal{A}$-creation-oblivious, and by construction, $\tilde{\sigma}$ is $\mathcal{B}$-creation-oblivious. This allows us to apply lemma 11.24 to obtain:

- $\epsilon_{\tilde{\sigma}}(C_{\bigotimes_i \hat{\mathcal{C}}_{\mathcal{A}}^{\vec{e}^{ac}_{[i]},\vec{\beta}_{[i],+}}}) = \prod_i^n \epsilon_{\tilde{\sigma}^i}(C_{\hat{\mathcal{C}}_{\mathcal{A}}^{\vec{\beta}[i],\vec{e}^{ac}_{[i]}}})$ with $\forall i \in [1,n], \tilde{\sigma}^i = oblivious_{\mathcal{A},\vec{\beta}[:i],\vec{e}^{ac}_{[:i]}}(\tilde{\sigma})$.
- $\epsilon_{\tilde{\sigma}}(C_{\bigotimes_i \hat{\mathcal{C}}_{\mathcal{B}}^{\vec{e}^{bc}_{[i]},\vec{\beta}_{[i],+}}}) = \prod_i^n \epsilon_{\tilde{\sigma}'^i}(C_{\hat{\mathcal{C}}_{\mathcal{B}}^{\vec{\beta}[i],\vec{e}^{ac}_{[i]}}})$ with $\forall i \in [1,n], \tilde{\sigma}'^i = oblivious_{\mathcal{B},\vec{\beta}[:i],\vec{e}^{bc}_{[:i]}}(\tilde{\sigma}')$.
- where $\vec{z}[:i] = \vec{z}[1]^\frown ... ^\frown \vec{z}[i-1]$ for $\vec{z} \in \{\vec{\beta}, \vec{e}^{ac}, \vec{e}^{bc}\}$
- $\hat{\mathcal{C}}_{\mathcal{A}}^{\vec{\beta}[i],\vec{e}^{ac}[i]} = class(Execs(\tilde{\mathcal{E}}||X_{\mathcal{A}})^{\downarrow \vec{e}^{ac}[:i])}, print^{proxy,\mathcal{A}}_{(\tilde{\mathcal{E}},X_{\mathcal{A}})}, (\vec{\beta}[i], \vec{e}^{ac}[i]))$
- $\hat{\mathcal{C}}_{\mathcal{B}}^{\vec{\beta}[i],\vec{e}^{bc}[i]} = class(Execs(\tilde{\mathcal{E}}||X_{\mathcal{B}})^{\downarrow \vec{e}^{bc}[:i])}, print^{proxy,\mathcal{B}}_{(\tilde{\mathcal{E}},X_{\mathcal{B}})}, (\vec{\beta}[i], \vec{e}^{bc}[i]))$

Thus it is enough to show that $\forall i \in [1,n], \epsilon_{\tilde{\sigma}^i}(C_{\hat{\mathcal{C}}_{\mathcal{A}}^{\vec{\beta}[i],\vec{e}^{ac}_{[i]}}}) = \epsilon_{\tilde{\sigma}'^i}(C_{\hat{\mathcal{C}}_{\mathcal{B}}^{\vec{\beta}[i],\vec{e}^{bc}_{[i]}}}))$. Let $i \in [1,n]$

We can apply lemma 11.12 to obtain:

- $\epsilon_{\tilde{\sigma}^i}(C_{\hat{\mathcal{C}}_{\mathcal{A}}^{\vec{\beta}[i],\vec{e}^{ac}_{[i]}}}) = \epsilon_{\sigma_{|\mathcal{A},\vec{\beta}[:i],\vec{e}^{ac}_{[:i]}}}(\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}^{ac}[i])}_{(\mathcal{E}_{\mathcal{A}},\mathcal{A})})$
- $\epsilon_{\tilde{\sigma}'^i}(C_{\hat{\mathcal{C}}_{\mathcal{B}}^{\vec{\beta}[i],\vec{e}^{bc}_{[i]}}}) = \epsilon_{\sigma'_{|\mathcal{B},\vec{\beta}[:i],\vec{e}^{bc}_{[:i]}}}(\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}^{bc}[i])}_{(\mathcal{E}_{\mathcal{B}},\mathcal{B})}$.

where:

- $\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}^{ac}[i])}_{(\mathcal{E}_{\mathcal{A}},\mathcal{A})} = Class(Execs(\mathcal{E}_{\mathcal{A}}^{\vec{e}^{ac}[:i]}||\tilde{\mathcal{A}}^{sw}), print_{(\mathcal{E}_{\mathcal{A}}^{\vec{e}^{ac}[:i]},\tilde{\mathcal{A}}^{sw})}, (\vec{\beta}[i], \vec{e}^{ac}[i]))$ and
- $\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}^{bc}[i])}_{(\mathcal{E}_{\mathcal{B}},\mathcal{B})} = Class(Execs(\mathcal{E}_{\mathcal{B}}^{\vec{e}^{bc}[:i]}||\tilde{\mathcal{B}}^{sw}), print_{(\mathcal{E}_{\mathcal{B}}^{\vec{e}^{bc}[:i]},\tilde{\mathcal{B}}^{sw})}, (\vec{\beta}[i], \vec{e}^{bc}[i]))$
- $\sigma_{|\mathcal{A},\vec{\beta}[:i],\vec{e}^{ac}[:i]}$ is the $((\tilde{\mathcal{E}}||X_{\mathcal{A}})^{\downarrow \vec{e}^{ac}[:i]}).\tilde{\mu}_e^{\mathcal{A},+}$ alter-ego of $\tilde{\sigma}^i$.
- $\sigma'_{|\mathcal{B},\vec{\beta}[:i],\vec{e}^{bc}[:i]}$ is the $((\tilde{\mathcal{E}}||X_{\mathcal{B}})^{\downarrow \vec{e}^{bc}[:i]}).\tilde{\mu}_e^{\mathcal{B},+}$ alter-ego of $\tilde{\sigma}'^i$.

.

Hence it is sufficient to show that $\epsilon_{\sigma_{|\mathcal{A},\vec{\beta}[:i],\vec{e}^{ac}[:i]}}(C_{\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}^{ac}[i])}_{(\mathcal{E}_{\mathcal{A}},\mathcal{A})}}) = \epsilon_{\sigma'_{|\mathcal{B},\vec{\beta}[:i],\vec{e}^{bc}[:i]}}(C_{\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}^{bc}[i])}_{(\mathcal{E}_{\mathcal{B}},\mathcal{B})}})$.

Finally, we find again our construction (*):

- $\epsilon_{\sigma_{|\mathcal{A},\vec{\beta}[:i],\vec{e}^{ac}[:i]}}(C_{\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}^{ac}[i])}_{(\mathcal{E}_{\mathcal{A}},\mathcal{A})}}) = \epsilon_{\sigma^c_{|\mathcal{A},\vec{\beta}[:i],\vec{e}[:i]}}(C_{\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}[i])}_{(\mathcal{E},\mathcal{A})}})$
- $\epsilon_{\sigma'_{|\mathcal{B},\vec{\beta}[:i],\vec{e}^{bc}[:i]}}(C_{\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}^{bc}[i])}_{(\mathcal{E}_{\mathcal{B}},\mathcal{B})}}) = \epsilon_{\sigma^d_{|\mathcal{B},\vec{\beta}[:i],\vec{e}[:i]}}(C_{\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}[i])}_{(\mathcal{E},\mathcal{B})}})$
- $\epsilon_{\sigma^c_{|\mathcal{A},\vec{\beta}[:i],\vec{e}[:i]}}(C_{\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}[i])}_{(\mathcal{E},\mathcal{A})}}) = \epsilon_{\sigma^d_{|\mathcal{B},\vec{\beta}[:i],\vec{e}[:i]}}(C_{\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}[i])}_{(\mathcal{E},\mathcal{B})}})$

where:

- $\vec{e}$ is the vector of $(Frags^*(\mathcal{E}))^n$ s. t. $\forall j \in [1:n], \vec{e}[j] = \mu_{\mathcal{AC}}(\vec{e}^{ac}[j]) = \mu_{\mathcal{CB}}^{-1}(\vec{e}^{bc}[j])$.
- $\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}[i])}_{(\mathcal{E},\mathcal{A})} = Class(Execs(\mathcal{E}^{\vec{e}[:i])}||\tilde{\mathcal{A}}^{sw}), print_{(\mathcal{E}^{\vec{e}[:i])},\tilde{\mathcal{A}}^{sw})}, (\vec{\beta}[i], \vec{e}[i]))$ and
- $\check{\mathcal{C}}^{(\vec{\beta}[i],\vec{e}[i])}_{(\mathcal{E},\mathcal{B})} = Class(Execs(\mathcal{E}^{\vec{e}[:i])}||\tilde{\mathcal{B}}^{sw}), print_{(\mathcal{E}^{\vec{e}[:i])},\tilde{\mathcal{B}}^{sw})}, (\vec{\beta}[i], \vec{e}[i]))$

.

This lead us to $\epsilon_{\sigma_{|\mathcal{A},\vec{\beta}[:i],\vec{e}^{ac}[:i]}}(C_{\check{C}^{(\vec{\beta}[i],\vec{e}^{ac}[i])}_{(\mathcal{E}_{\mathcal{A}},\mathcal{A})}}) = \epsilon_{\sigma'_{|\mathcal{B},\vec{\beta}[:i],\vec{e}^{bc}[:i]}}(C_{\check{C}^{(\vec{\beta}[i],\vec{e}^{bc}[i])}_{(\mathcal{E}_{\mathcal{B}},\mathcal{B})}})$, which ends the proof.

□

12 TASK SCHEDULE

We have shown in previous section that $\leq^{print}_{CrOb}$ was a monotonic relationship. In this section, we explain why an easy to use off-line scheduler introduced by Canetti & al. [4] is not creation-oblivious which suprisingly prevent us to obtain monotonicity of implementation for scheduler schema.

12.1 Task-schedule

Here we present a subclass of fully off-line schedulers, called *task-schedules*. Before explaining what is a task-schedule, we introduce the definition of task, adapted from [2] to the dynamic setting. In practice a task is invoked by the scheduler to potentially trigger an action from a sub-component of the automaton. An automaton can be the result of the composition of several automata, themselves the result of the composition of several automata and so on and forth. We assume the existence of a subset $Autids_0 \subset Autids$ that represents the "atomic entities" of our formalism. Any automaton is the result of the composition of automata in $Autids_0$.

Definition 12.1 (Constitution). For every $\mathcal{A} \in Autids$, we note

$constitution(\mathcal{A}) : \begin{cases} states(\mathcal{A}) & \rightarrow \quad \mathcal{P}(Autids_0) \text{ where } \mathcal{P}(Autids_0) \text{ denotes the power set of } Autids_0 \\ q & \mapsto \quad constitution(\mathcal{A})(q) \end{cases}$

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

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

The constitution of a PCA is defined recursively through it configuration. For every PCA X, for every $q \in states(X)$, if we note $(\mathbf{A}, \mathbf{S}) = config(X)(q)$, $constitution(X)(q) = \bigcup_{\mathcal{A}\in\mathbf{A}} constitution(\mathcal{A})(\mathbf{S}(\mathcal{A}))$.

Definition 12.2 (Task). A task T is a pair $(id, actions)$ where $id \in Autids_0$ and $actions \subset actions(aut(id))$ is a set of action labels. Let $T = (id, actions)$, we note $id(T) = id$ and $actions(T) = actions$.

Definition 12.3 (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 $\eta \in Disc(Q_{\mathcal{A}})$ s. t. $(q, a, \eta) \in D_{\mathcal{A}}$.

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

Now we are ready to define a task-schedule, which is a particular subclass of schedulers.

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

Definition 12.4 (task-schedule). A task-schedule $\rho = T_1, T_2, T_3, ...$ is a (finite or infinite) sequence of tasks.

Since our task-schedule is defined, we are ready to solve the non-determinism and define a probability on the executions of a PSIOA. We use the measure of [2].

Definition 12.5. (task-based probability on executions: $apply_{\mathcal{A}}(\mu, \rho) : Frags(\mathcal{A}) \rightarrow [0, 1]$) 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:
- $p_1(\alpha) = \begin{cases} \mu(\alpha')\eta_{(\mathcal{A},q',a)}(q) & \text{if } \alpha = \alpha'^\frown(a,q),\ q' = lstate(\alpha') \text{ and } a \text{ is triggered by } T \text{ enabled after } \alpha' \\ 0 & \text{otherwise} \end{cases}$
- $p_2(\alpha) = \begin{cases} \mu(\alpha) & \text{if } T \text{ is not enabled after } \alpha \\ 0 & \text{otherwise} \end{cases}$

(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 $apply_{\mathcal{A}}(\mu, \rho) := \lim_{i \to \infty} pm_i$.

Proposition 2. *Let $\mathcal{A}$ be a PSIOA, For each measure μ on $Frags^*(\mathcal{A})$ and task schedule ρ, there is scheduler σ for $\mathcal{A}$ such that $apply(\mu, \rho)$ is the generalized probabilistic execution fragment $\epsilon_{\sigma,\mu}$.*

Proof. The result has been proven in [2], appendix B.4. □

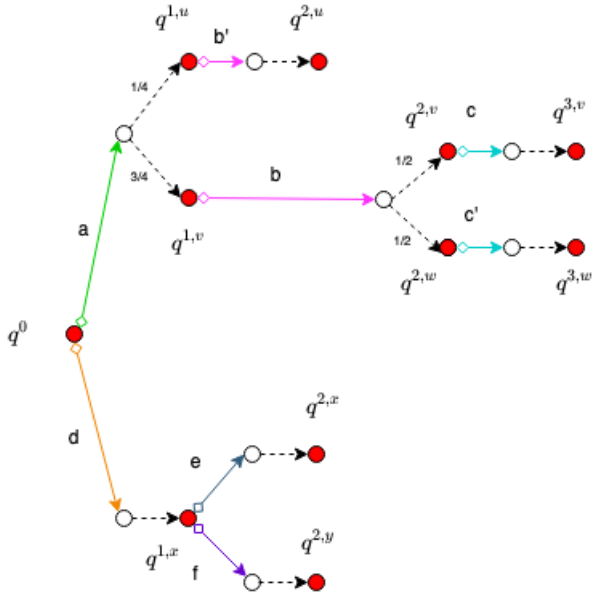


Fig. 31. Non-deterministic execution: The scheduler allows us to solve the non-determinism, by triggering an action among the enabled one. We give an example with an automaton $\mathcal{A} = (Q_{\mathcal{A}}, \bar{q}_{\mathcal{A}} = q_0, sig(\mathcal{A}), D_{\mathcal{A}})$ and the tasks T_g, T_o, T_p, 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 $q^{2,w}$.

12.2 Why a task-scheduler is not creation-oblivious ?

Let imagine the following example. The class C^x is composed of two executions $\alpha^{x,1}$ and $\alpha^{x,2}$, the class C^y is composed of two executions $\alpha^{y,1}$ and $\alpha^{y,2}$ and the class C^z is composed of four executions $\alpha^{z,11} = \alpha^{x,1}\frown\alpha^{y,1}$, $\alpha^{z,12} = \alpha^{x,1}\frown\alpha^{y,2}$, $\alpha^{z,21} = \alpha^{x,2}\frown\alpha^{y,1}$, $\alpha^{z,22} = \alpha^{x,2}\frown\alpha^{y,2}$. Let $\rho = \rho^1\frown\rho^2$ be a task-schedule. We do not have $apply(.,\rho)(C^z)) = apply(.,\rho^1)(C^x) \cdot apply(.,\rho^2)(C^y)$! Indeed, the executions $\alpha^{x,1}$ and $\alpha^{x,2}$ can differ s. t. they do not ignore the same tasks. Typically, ρ^1 could be written $\rho^1 = \rho^{1,a}\frown\rho^{1,b}$ where the last action of $\alpha^{x,1}$ is triggered by the last task of $\rho^{1,a}$ and $\rho^{1,b}$ is "ignored by $\alpha^{x,1}$. The issue comes if both $apply(.,\rho^2)(C^y) \neq \emptyset$ and $apply(.,\rho^{1,b}\frown\rho^2)(C^y) \neq \emptyset$. The point is that C^z can be obtained with different cut-paste: cut-paste A: $\rho^{1,a}$ for C^x and $\rho^{1,b}\frown\rho^2$ for C^y ; cut-paste B: ρ^1 for C^x and ρ^2 for C^y.

13 CONCLUSION

We extended *dynamic I/O Automata* formalism of Attie & Lynch [1] to probabilistic settings in order to cope with emergent distributed systems such as peer-to-peer networks, robot networks, adhoc networks or blockchains. Our formalism includes operators for parallel composition, action hiding, action renaming, automaton creation and use a refined definition of probabilistic configuration automata in order to cope with dynamic actions. The key result of our framework is as follows: the implementation of probabilistic configuration automata is monotonic to automata creation and destruction. That is, if systems $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ differ only in that $X_{\mathcal{A}}$ dynamically creates and destroys automaton $\mathcal{A}$ instead of creating and destroying automaton $\mathcal{B}$ as $X_{\mathcal{B}}$ does, and if $\mathcal{A}$ implements $\mathcal{B}$ (in the sense they cannot be distinguished by any external observer), then $X_{\mathcal{A}}$ implements $X_{\mathcal{B}}$. This results is particularly interesting in the design and refinement of components and subsystems in isolation. In our construction we exhibit the need of considering only *creation-oblivious* schedulers in the implementation relation, i. e. a scheduler that, upon the (dynamic) creation of a sub-automaton $\mathcal{A}$, does not take into account the previous internal actions of $\mathcal{A}$ to output (randomly) a transition.

Interestingly and of independent interest, motivated by the monotonicity of execution w.r.t. to automata creation, we introduce new proof techniques to deduce certain properties of a system $X_{\mathcal{A}}$ from a sub-automaton $X_{\mathcal{A}}$ dynamically created and destroyed by $X_{\mathcal{A}}$. This proof technique is used to construct a homomorphism between the probabilistic spaces of automata executions. Then we expose such homomorphism from a system $X_{\mathcal{A}}$ to a new system resulting from the composition of $\mathcal{A}$ and $X_{\mathcal{A}} \setminus \{\mathcal{A}\}$. The latter corresponds intuitively to the system $X_{\mathcal{A}}$ deprived of $\mathcal{A}$. Furthermore, the homomorphism is used to show that under certain minor technical assumptions, if $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ differ only in the fact that $X_{\mathcal{A}}$ dynamically creates and destroys the automaton $\mathcal{A}$ instead of creating and destroying the automaton $\mathcal{B}$ as $X_{\mathcal{B}}$ does, then $X_{\mathcal{A}} \setminus \{\mathcal{A}\}$ and $X_{\mathcal{B}} \setminus \{\mathcal{B}\}$ are semantically equivalent, i.e. they only differ syntactically. The homomorphism is finally reused to establish the monotonicity of the implementation relation. Our technique can be used in extensions of our formalism with time and cryptography notions.

As future work we plan to extend the composable secure-emulation of Canetti et al. [4] to dynamic settings. This extension is necessary for formal verification of protocols combining probabilistic distributed systems and cryptography in dynamic settings (e.g. blockchains, secure distributed computation, cybersecure distributed protocols etc).

REFERENCES

[1] Paul C. Attie and Nancy A. Lynch. 2016. Dynamic input/output automata: A formal and compositional model for dynamic systems. *Inf. Comput.* 249 (2016), 28–75. https://doi.org/10.1016/j.ic.2016.03.008

[2] Ran Canetti, Ling Cheung, Dilsun Kaynar, Moses Liskov, Nancy Lynch, Olivier Pereira, and Roberto Segala. 2018. Task-Structured Probabilistic {I/O} Automata. *J. Comput. System Sci.* 94 (2018), 63—-97. https://doi.org/10.1016/j.jcss.2017.09.007

[3] Ran Canetti, Ling Cheung, Dilsun Kirli Kaynar, Moses D. Liskov, Nancy A. Lynch, Olivier Pereira, and Roberto Segala. 2005. Using Probabilistic I/O Automata to Analyze an Oblivious Transfer Protocol. *IACR Cryptol. ePrint Arch.* (2005), 452. http://eprint.iacr.org/2005/452

[4] Ran Canetti, Ling Cheung, Dilsun Kirli Kaynar, Nancy A. Lynch, and Olivier Pereira. 2007. Compositional Security for Task-PIOAs. In *20th IEEE Computer Security Foundations Symposium, CSF 2007, 6-8 July 2007, Venice, Italy*. IEEE Computer Society, 125–139. https://doi.org/10.1109/CSF.2007.15

[5] Jing Chen and Silvio Micali. 2019. Algorand: A secure and efficient distributed ledger. *Theor. Comput. Sci.* 777 (2019), 155–183. https://doi.org/10.1016/j.tcs.2019.02.001

[6] Nancy Lynch, Michael Merritt, William Weihl, and Alan Fekete. 1988. A theory of atomic transactions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 326 LNCS (1988), 41–71. https://doi.org/10.1007/3-540-50171-1_3

[7] Martin L. Puterman. 1994. *Markov decision processes: discrete stochastic dynamic programming* (1 ed.). John Wiley & Sons.

[8] Roberto Segala. 1995. *Modeling and Verification of Randomized Distributed Real-Time Systems.* Ph.D. Dissertation. Massachusettes Institute of technology.