Polyadic History-Dependent Automata for the Fusion Calculus*

Emilio Tuosto[1], Björn Victor[2], and Kidane Yemane[2]

[1] Dipartimento di Informatica, Università di Pisa, Italy.
[2] Dept. of Information Technology, Uppsala University, Sweden.

Abstract. We extend History Dependent Automata to handle polyadic labels, and using a new symbolic semantics of fusion calculus we give a mapping into these Polyadic HDA with Negative Transitions, and show that the mapping is adequate with respect to hyperequivalence in the fusion calculus.
This lays the grounds for HD-automata-based tools applicable not only to the monadic π-calculus but also to the fusion calculus and polyadic π-calculus, allowing implementation efforts to be focused at a foundational level rather than being multiplied in several tools.

1 Introduction

With the development of the *History Dependent Automata* framework (HD-automata in brief) by Montanari, Pistore and Ferrari [6, 10, 7, 1], a promising path to the development of generic, reusable algorithms and tools for automated verification of mobile processes has opened. Rather than developing techniques and algorithms at the level of the mobile process calculus, adapting them from one calculus variant to the next, and often multiplying the work spent on optimization and specialization in each tool, this work can be focused at the level of HD-automata. What is necessary for this are mappings from each process calculus to an appropriate variant of HD-automata, and proof that the properties of interest are preserved by the mapping.

In this paper, we apply this approach to the *fusion calculus*, an extension and simplification of the π-calculus developed by Parrow and Victor [9], where the effects of communication are not necessarily local to the input end (as they are in the π-calculus). Our results are initially applicable to equivalence checking.

HD-automata provide an operational model for history dependent calculi, i.e., calculi whose semantics are defined in terms of transition systems such that the transitions may carry information generated in earlier transitions of the system, and this "historical" information can influence the future behaviour of the system. Mobile process calculi such as the π-calculus, where the communication topology may change dynamically, are typical such history dependent calculi. HD-automata give finite representations of classes of infinite labelled transition systems. Symbolic semantics at the process calculus level have also been used for this purpose, but HD-automata give additional benefits: compact representation of states using symmetries of names, and more importantly, a *unified* foundational framework for history-dependent calculi.

A tool based on HD-automata have been developed [2]. It works on a variant of HD (HDS) suitable for early and late semantics of π-calculus, and very efficiently performs minimization of automata. HDS are not applicable for open equivalence in π-calculus or hyperequivalence in fusion calculus. An extension of HDS, *History Dependent Automata with Negative Transitions* (HDN) [10], is required. In this paper we further extend HDN in order to model polyadic rather than monadic communication. This extension is useful not only for our application, that of modelling fusion calculus in HDN, but also for open semantics of the π-calculus. The same extension can also be applied to HDS, allowing polyadicity also in early and late semantics.

* Work supported by the PROFUNDIS FET-GC project.

Overview. In the following section we recapitulate the syntax, semantics, and bisimulation congruence of the fusion calculus [9]. In Section 3 we introduce a canonical symbolic semantics for the fusion calculus, along the lines of [12], and prove that the original hyperequivalence coincides with symbolic bisimulation. After an introduction to HD in Section 4, we extend the HDN from monadic to polyadic in Section 5. Finally, in Section 6, we give a mapping of fusion calculus into HDN using the symbolic semantics, and in Section 7 show that bisimulation in HDN coincides with hyperequivalence. Full proofs are found in Appendix A.

2 Syntax and Semantics of the Fusion Calculus

In this section we first review the syntax and operational semantics of the fusion calculus with match and recursion, and then its strong bisimulation equivalence and congruence from [9]. The novel contribution of the present paper begins by exploring the canonical symbolic semantics.

We assume an infinite enumerable set $\mathcal{N}$ of *names* ranged over by $u, v, \ldots, z$. Like in the π-calculus, names represent communication channels, which are also the values transmitted. We write $\tilde{x}$ for a (possibly empty) finite sequence $x_1 \cdots x_n$ of names. φ ranges over total equivalence relations over $\mathcal{N}$ (i.e. equivalence relations with $\mathrm{dom}(\varphi) = \mathcal{N}$) with only finitely many non-singular equivalence classes. We write $\{\tilde{x} = \tilde{y}\}$ to mean the smallest such equivalence relation relating each x_i with y_i, and write $\mathbf{1}$ for the identity relation. If x and y are related by φ we write $x\ \varphi\ y$.

Definition 1. *The* free actions*, ranged over by* α*, and the* agents*, ranged over by* $P, Q, \ldots$*, are defined by*

$$\begin{array}{lll} \alpha ::= u\tilde{x} & \text{(Input)} & P ::= \mathbf{0} \mid \alpha\,.\,Q \mid Q + R \mid Q \mid R \\ \quad\;\; \overline{u}\tilde{x} & \text{(Output)} & \qquad\;\; \mid (x)Q \mid [x = y]Q \mid A(\tilde{x}) \\ \quad\;\; \varphi & \text{(Fusion)} & \end{array}$$

In the fusion calculus prefixes correspond exactly to the free actions defined above. An input action $u\tilde{x}$ means "input objects along the port u and replace $\tilde{x}$ with these objects." Contrary to the situation in the π-calculus this action does not bind $\tilde{x}$. In other words, the scope of $\tilde{x}$ is not bounded by the action. The output action $\overline{u}\tilde{x}$ is familiar from the π-calculus and means "output the objects $\tilde{x}$ along the port u". A fusion action $\{\tilde{x} = \tilde{y}\}$ represents an obligation to make $\tilde{x}$ and $\tilde{y}$ equal everywhere.

Input and output actions are collectively called *free communication* actions. In these, the names $\tilde{x}$ are the *objects* of the action, and the name u is the *subject*. We write a to stand for either u or $\overline{u}$, thus $a\tilde{x}$ is the general form of a communication action. Fusion actions have neither subject nor objects.

The name x is said to be *bound* in $(x)P$. We write $(\tilde{x})P$ for $(x_1)\cdots(x_n)P$. The *free names* in P, denoted $\mathrm{fn}(P)$, are the names in P with a non-bound occurrence; here $\mathrm{fn}(\varphi)$ is defined to be the names in the non-singular equivalence classes, i.e., in the relation $\varphi - \mathbf{1}$. Notice that since the syntax $\{x = x\}$ means $\mathbf{1}$, the free names of that fusion do not include x, but is the empty set. As usual we will not distinguish between alpha-variants of agents, i.e., agents differing only in the choice of bound names.

The action of a transition may be free or bound:

Definition 2. *The* actions*, ranged over by* γ *and* δ*, consist of the fusion actions and of communication actions of the form* $(z_1)\cdots(z_n)a\tilde{x}$ *(written* $(\tilde{z})a\tilde{x}$*), where* $n \geq 0$ *and all names in* $\tilde{z}$ *are also in* $\tilde{x}$*. If* $n > 0$ *we say it is a* bound *action.*

Note that there are no bound fusion actions. In the communication actions above, $\tilde{z}$ are the *bound objects* and the names in $\tilde{x}$ that are not in $\tilde{z}$ are the *free objects*. Free actions have no bound objects. We further write $\mathrm{n}(\gamma)$ to mean all names occurring in γ, both bound and free.

We let M, N, L range over sequences of match operators, and we say that M *implies* N, written $M \Rightarrow N$, if the conjunction of all matches in M logically implies all elements in N, and that $M \Leftrightarrow N$ if M and N imply each other.

We define $\varphi \backslash z$ to mean $\varphi \cap (\mathcal{N} - \{z\})^2 \cup \{(z, z)\}$, i.e., the equivalence relation φ with all references to z removed (except for the identity). For example, $\{x = z, z = y\}\backslash z = \{x = y\}$, and $\{x = y\}\backslash y = \mathbf{1}$.

$$\text{PREF}\ \frac{-}{\alpha\,.\,P \xrightarrow{\alpha} P} \qquad \text{SUM}\ \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \text{PAR}\ \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\text{COM}\ \frac{P \xrightarrow{u\tilde{x}} P',\ Q \xrightarrow{\overline{u}\tilde{y}} Q',\ |\tilde{x}| = |\tilde{y}|}{P \mid Q \xrightarrow{\{\tilde{x}=\tilde{y}\}} P' \mid Q'} \qquad \text{SCOPE}\ \frac{P \xrightarrow{\varphi} P',\ z\,\varphi\,x,\ z \neq x}{(z)P \xrightarrow{\varphi \backslash z} P'\{x/z\}}$$

$$\text{PASS}\ \frac{P \xrightarrow{\alpha} P',\ z \notin \mathrm{n}(\alpha)}{(z)P \xrightarrow{\alpha} (z)P'} \qquad \text{OPEN}\ \frac{P \xrightarrow{(\tilde{y})a\,\tilde{x}} P',\ z \in \tilde{x} - \tilde{y},\ a \notin \{z, \overline{z}\}}{(z)P \xrightarrow{(z\tilde{y})a\,\tilde{x}} P'} \qquad \text{MATCH}\ \frac{P \xrightarrow{\alpha} P'}{[x = x]P \xrightarrow{\alpha} P'}$$

Table 1. Transition rules for the Fusion Calculus.

Definition 3. *The* structural congruence, $\equiv$, *between agents is the least congruence satisfying the abelian monoid laws for Summation and Composition (associativity, commutativity and* **0** *as identity), and the scope laws* $(x)\mathbf{0} \equiv \mathbf{0},\ (x)(y)P \equiv (y)(x)P,\ (x)(P + Q) \equiv (x)P + (x)Q,\ (x)MP \equiv M(x)P$ *if* $x \notin \mathrm{n}(M)$ *and also the* scope extension *law* $P \mid (z)Q \equiv (z)(P \mid Q)$ *where* $z \notin \mathrm{fn}(P)$*, and the law for process identifiers:* $A(\tilde{y}) \equiv P\{\tilde{y}/\tilde{x}\}$ *if* $A(\tilde{x}) \stackrel{def}{=} P$.

We proceed by giving the labelled transition semantics of fusion calculus.

Definition 4. *The family of* transitions $P \xrightarrow{\gamma} Q$ *is the least family satisfying the laws in Table 1. In this definition structurally equivalent agents are considered the same, i.e., if* $P \equiv P'$ *and* $Q \equiv Q'$ *and* $P \xrightarrow{\gamma} Q$ *then also* $P' \xrightarrow{\gamma} Q'$.

Definition 5. *A substitution* σ agrees with *the fusion* φ *if* $\forall x, y : x\,\varphi\,y \Leftrightarrow \sigma(x) = \sigma(y)$. *A* substitutive effect *of a fusion* φ *is an idempotent substitution* σ *agreeing with* φ *such that* $\forall x, y : \sigma(x) = y \Rightarrow x\,\varphi\,y$ *(i.e.,* σ *sends all members of the equivalence class to one representative of the class). The only substitutive effect of a communication action is the identity substitution. For any action* γ *we write* σ_γ *for its substitutive effect.*

For example, the substitutive effects of $\{x = y\}$ are $\{x/y\}$ and $\{y/x\}$. Note that not all substitutions which agree with φ are substitutive effects of φ. For example any injective substitution agrees with $\mathbf{1}$, but the only substitutive effect of $\mathbf{1}$ is the identity substitution.

Lemma 6. *If* $P \xrightarrow{\gamma} P'$*, then* $P\sigma \xrightarrow{\gamma\sigma} P'\sigma$*, for any substitution* σ.

We end this section by reviewing the definition of *hyperbisimulation* [9].

Definition 7. *A* fusion bisimulation *is a binary symmetric relation* $\mathcal{S}$ *between agents such that* $(P, Q) \in \mathcal{S}$ *implies:*
If $P \xrightarrow{\gamma} P'$ *with* $\mathrm{bn}(\gamma) \cap \mathrm{fn}(Q) = \emptyset$*, then* $Q \xrightarrow{\gamma} Q'$ *and* $(P'\sigma_\gamma, Q'\sigma_\gamma) \in \mathcal{S}$. P *is* fusion bisimilar *to* Q*, written* $P \stackrel{.}{\sim} Q$*, if* $(P, Q) \in \mathcal{S}$ *for some fusion bisimulation* $\mathcal{S}$.

A hyperbisimulation *is a substitution closed fusion bisimulation, i.e., a fusion bisimulation* $\mathcal{S}$ *with the property that* $(P, Q) \in \mathcal{S}$ *implies* $(P\sigma, Q\sigma) \in \mathcal{S}$ *for any substitution* σ. *Two agents* P *and* Q *are* hyper-equivalent, *written* $P \sim Q$*, if they are related by a hyperbisimulation.*

The interesting point in this definition is the treatment of fusion actions. A fusion $\{x = y\}$ represents an obligation to make x and y equal everywhere. Therefore, if γ above is such a fusion, it only makes sense to relate P' and Q' when a substitution $\{y/x\}$ or $\{x/y\}$ has been performed. Note that it does not matter which substitution we choose, since $P\{x/y\} \stackrel{.}{\sim} Q\{x/y\}$ implies $P\{y/x\} \stackrel{.}{\sim} Q\{y/x\}$, by the simple fact that $P\{x/y\}\{y/x\} \equiv P\{y/x\}$ and that bisimulation is closed under injective substitutions.

$$\text{S-PREF}\ \frac{-}{\alpha\,.\,P \overset{\emptyset,\alpha}{\longmapsto} P\sigma_\alpha} \qquad \text{S-SUM}\ \frac{P \overset{M,\alpha}{\longmapsto} P'}{P+Q \overset{M,\alpha}{\longmapsto} P'} \qquad \text{S-PAR}\ \frac{P \overset{M,\alpha}{\longmapsto} P'}{P \mid Q \overset{M,\alpha}{\longmapsto} P' \mid Q\sigma_M\sigma_\alpha}$$

$$\text{S-COM}\ \frac{P \overset{M,u\tilde{x}}{\longmapsto} P',\ \ Q \overset{N,\overline{v}\tilde{y}}{\longmapsto} Q',\ \ |\tilde{x}| = |\tilde{y}|,\ \ L = MN[u=v],\ \varphi = \{\tilde{x} = \tilde{y}\}\sigma_L}{P \mid Q \overset{L,\varphi}{\longmapsto} (P' \mid Q')\sigma_L\sigma_\varphi}$$

$$\text{S-SCOPE}\ \frac{P \overset{M,\varphi}{\longmapsto} P',\ z\,\varphi\,x,\ z \neq x,\ z \notin \mathrm{n}(M)}{(z)P \overset{M,\varphi\backslash z}{\longmapsto} P'\{x/z\}} \qquad \text{S-PASS}\ \frac{P \overset{M,\alpha}{\longmapsto} P',\ z \notin \mathrm{n}(M,\alpha)}{(z)P \overset{M,\alpha}{\longmapsto} (z)P'}$$

$$\text{S-OPEN}\ \frac{P \overset{M,(\tilde{y})a\,\tilde{x}}{\longmapsto} P',\ z \in \tilde{x} - \tilde{y},\ a \notin \{z, \overline{z}\},\ z \notin \mathrm{n}(M)}{(z)P \overset{M,(z\tilde{y})a\,\tilde{x}}{\longmapsto} P'} \qquad \text{S-MATCH}\ \frac{P \overset{M,\alpha}{\longmapsto} P'}{[x=y]P \overset{M[x=y],\alpha}{\longmapsto} P'\sigma_{M[x=y]}}$$

Table 2. Canonical symbolic transition system for the Fusion Calculus.

Theorem 8. [9] *Hyperequivalence is the largest congruence in fusion bisimilarity.*

For further examples and explanations we refer the reader to [9, 5].

3 Canonical Symbolic Semantics of Fusion Calculus

In this section we present a *canonical* symbolic semantics for the fusion calculus, along the lines of symbolic semantics for the π-calculus [12, 11]. Symbolic semantics are often used to give efficient characterizations of bisimulation equivalences for value-passing calculi.

In Table 2 we present the symbolic transition system. Like in [12] a symbolic transition is of the form $P \overset{M,\gamma}{\longmapsto} Q$, where M is the enabling condition of the action γ in the sense that M represents the equalities a minimal substitution σ_M must make true in order for $P\sigma_M$ to perform the corresponding action in the original labelled transition system.

Following Pistore and Sangiorgi's work [11], our transition rules apply substitutions to the continuation of a transition: like [11], a substitution σ_M, making the condition for the transition true, and in addition a substitution σ_γ, the substitutive effect of the action. The motivation for this is to make the definition of bisimulation simpler and more in line with the algorithms used in the HD framework (see Section 5). We show later in this section that bisimulation using the symbolic semantics coincides with the original non-symbolic version.

While we can in general use any σ_M which agrees with M and which is identity for all names not in M, and any substitutive effect σ_γ of the action, like Pistore and Sangiorgi [11] we choose *canonical* substitutions, which selects the *minimal* representative of each equivalence class of M or γ and maps all members to it. Henceforth, σ_M and σ_γ refer to canonical substitutions. Note that hyperequivalence (Definition 7) does not change, since we can use any substitutive effect there.

Using canonical substitutions gives us pleasant properties like the following:

Lemma 9. *If $P \overset{M,\gamma}{\longmapsto} P'$, then $\gamma = \gamma\sigma_M$ and $P' = P'\sigma_M = P'\sigma_\gamma = P'\sigma_M\sigma_\gamma$.*

The definition of symbolic hyperbisimulation is similar to that of symbolic open bisimulation [12, 11], but does not have the complication of distinctions.

Definition 10. *A binary symmetric process relation $\mathcal{S}$ is a* symbolic hyperbisimulation *if $(P,Q) \in \mathcal{S}$ implies:*
If $P \overset{M,\gamma}{\longmapsto} P'$ with $\mathrm{bn}(\gamma) \cap \mathrm{fn}(Q) = \emptyset$ then $Q \overset{N,\gamma'}{\longmapsto} Q'$ such that

– $M \Rightarrow N$,
– $\gamma = \gamma' \sigma_M$,   (note $\gamma = \gamma\sigma_M$)
– and $(P', Q'\sigma_M) \in \mathcal{S}$ (note $P' = P'\sigma_M$).

P is symbolically hyperequivalent *to Q, written $P \simeq Q$, if $(P, Q) \in \mathcal{S}$ for some symbolic hyperbisimulation $\mathcal{S}$.*

Since the symbolic semantics applies the substitution effects, we can leave most of that out of the bisimulation definition. It is still necessary to apply substitution corresponding to the stronger condition, σ_M, to the label and continuation of the transition of Q. (Note that $Q'\sigma_M = Q'\sigma_M\sigma_\gamma$.)

In the remainder of this section we establish the correspondence between symbolic hyperequivalence (Definition 10) and the standard hyperequivalence (Definition 7) by proving Theorem 15: $P \sim Q$ iff $P \simeq Q$.

Lemma 11.

1. $\sigma\sigma_{R\sigma} = \sigma_R\rho$*, for any substitution σ and some ρ, where R is an equivalence relation.*
2. *If $M \Rightarrow N$ then $M\sigma = N\sigma\rho$, for some substitution ρ*
3. $\sigma_R\sigma_{S\sigma_R} = \sigma_S\sigma_R$*, where R and S are equivalence relations.*

Lemma 12.

1. *If $P \xmapsto{M,\gamma} P'$, then $P\sigma \xmapsto{M\sigma,\gamma\sigma} P'\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma}$.*
2. *if $P\sigma \xmapsto{N,\gamma'} P'$, then $P \xmapsto{M,\gamma} P''$ with $M\sigma \Leftrightarrow N$, $\gamma\sigma = \gamma'$, and $P' = P''\sigma\sigma_N\sigma_{\gamma'}$.*

Proof. By transition induction, using Lemma 11.

Lemma 13. *$P \simeq Q$ implies $P\sigma \simeq Q\sigma$*

Proof. Straightforward diagram chasing, using Lemmas 11 and 12.

Lemma 14.

1. *If $P \xmapsto{M,\gamma} P'$, then $P\sigma_M \xrightarrow{\gamma} P''$ s.t. $P' = P''\sigma_\gamma$;*
2. *if $M \Rightarrow N$ and $P\sigma_M \xrightarrow{\gamma} P'$, then $P \xmapsto{N,\gamma'} P''$ such that $\gamma = \gamma'\sigma_M$ and $P'\sigma_\gamma = P''\sigma_M$.*

Proof. Again by transition induction, using Lemmas 11 and 6.

Theorem 15. *$P \sim Q$ iff $P \simeq Q$*

Proof of $\Rightarrow$: by showing $\mathcal{S} = \{(P, Q) : P \sim Q\}$ is a symbolic hyperbisimulation, using Lemmas 14 and 11.

Proof of $\Leftarrow$: We already have closure under substitution (Lemma 13), and show that $\mathcal{S} = \{(P, Q) : P \simeq Q\}$ is a fusion bisimulation using Lemmas 14 and 11.

4 History Dependent Automata

Verification of systems that are modelled as mobile processes (i.e. concurrent systems whose communication topology may change dynamically) is hard. For instance, in fusion calculus, agents can perform transitions that generate new names, e.g. the OPEN rule in Table 1 establishes that an agent, $(z)P$, can create a new name and can extrude it. The OPEN rule introduces an infinite branching in the automata corresponding to agents that perform such transitions since z can be freely replaced with any name in $\mathcal{N} \setminus \mathrm{fn}(P)$.

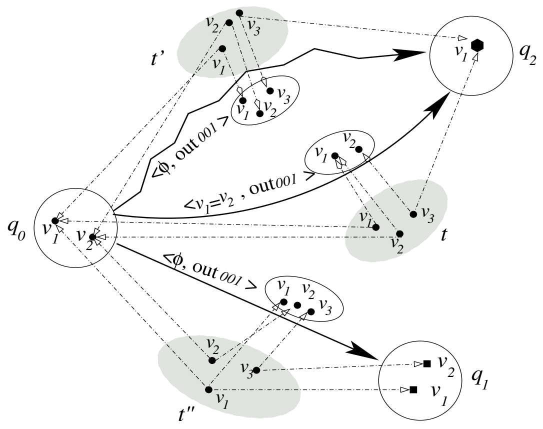

Fig. 1. HD-automaton for a fusion calculus agent

History Dependent automata (HD-automata in brief) have been proposed in [10, 6, 7, 1] as a new operational model for history dependent calculi, i.e., calculi whose semantics are defined in terms of a labelled transition system such that the labels may carry information generated in earlier transitions of the system that can influence the future behaviour of the system, e.g. the fusion calculus.

HD-automata aim at giving a finite representation of otherwise infinite LTSs; similarly to ordinary automata, they consist of states and labeled transitions, however, states and transitions of HD-automata are equipped with names which are no longer treated as syntactic components of labels, but become an explicit part of the operational model. This permits modelling of typical linguistic mechanisms of name passing calculi, such as creation and extrusion of names. Moreover, it allows for compact representation of agent behaviour by collapsing states that differ only for renaming of local names.

Various families of HD-automata have been introduced. Roughly speaking each class of HD-automata corresponds to a class of history dependent calculi or different behavioural semantics. In order to deal with asynchronous and open semantics of π-calculus, HD-automata *with negative transitions* have been proposed (the reader is referred to [10] for details).

Negative transitions "cover" regular transitions which may become *redundant*. For example, let us consider the fusion agent $P = (y)(\overline{u}\,vy\,.\,B(u,y) + [u = v]\overline{v}\,vy\,.\,B(y,y))$. Provided that $B(y,y)$ is equivalent to $B(u,y)\{v/u\}$, according to symbolic hyperbisimulation (Definition 10), P is equivalent to $(y)\overline{u}\,vy\,.\,B(u,y)$ because the transition $P \xmapsto{[u=v],(y)\overline{v}\,vy} B(y,y)$ is matched by transition $P \xmapsto{\emptyset,(y)\overline{u}\,vy} B(u,y)$; thus it is *redundant* in P.

Figure 1 depicts the HD-automaton corresponding to the fusion agent P [3]. It has three states, q_0, q_1 and q_2, that respectively correspond to agents P, $B(u,y)$ and $B(y,y)$, and three transitions t, t' and t''. Since names are local to states, labels and transitions of HD-automata, they have been renamed in order to "normalise" them, as will be more clear in Section 6 (where we also comment on the "001" of `out` labels). We need a mechanism for describing how names correspond to each other along transitions. Graphically, we represent such correspondences using dashed wires that connect names of the label and the source and target states of transitions. All transitions have three names v_1, v_2 and v_3. Name v_3 is connected to v_2 in q_1 and is observed in the label, but it has no counterpart in q_0, meaning that it is a name created by t'' (and similarly for t and t'). Transitions t and t'', represented with straight arrows, are *positive* transitions, while transition t', the zig-zag arrow, is *negative*. It is used to detect the *redundancy* of t; as defined later, a positive transition t is redundant when there is a negative transition that corresponds to t, leading to an equivalent

[3] The algorithm for mapping P in the automaton of Figure 1 will be given in Section 6. Here we just give an intuition of the interplay between positive and negative transitions.

state. Redundant transitions can be safely removed from the HD-automaton during the bisimulation check without affecting bisimilarity (Theorem 27). Notice also that if we remove t and t' from Figure 1 we obtain the automaton corresponding to $(y)\overline{u}\,vy\,.\,B(y,y)$ (equivalent to P).

5 Polyadic HD-automata with Negative Transitions

This section, after some notation, introduces *polyadic HD-automata* which can address polyadic calculi. Basically, we set the framework for formally defining HD-automata and the mapping to these from fusion calculus agents.

Given a relation $f \subseteq A \times B$, we define $\mathrm{dom}(f) = \{a \in A : \exists b \in B.(a,b) \in f\}$ and $\mathrm{cod}(f) = \{b \in B : \exists a \in A.(a,b) \in f\}$. When $\mathrm{cod}(f) = B$, we say that f is an *embedding of* B *in* A (written $f : A \multimap\!\!\rightarrow B$ or $f : B \leftarrow\!\!\multimap A$). An embedding $f \subseteq A \times B$ associates elements of A to elements of B. *Inverse injections* can be simply defined as follows:

Definition 16. *A relation $f \subseteq A \times B$ is an* inverse injection *(written $f : A \hookleftarrow\!\!\!\rightharpoonup B$) iff f is an embedding of B in A, $(a,b),(a',b) \in f \Rightarrow a = a'$ and f^{-1} is injective.*

An inverse injection $f : A \hookleftarrow\!\!\!\rightharpoonup B$ can be seen as an embedding of B into (a subset of) A. Condition $\mathrm{cod}(f) = B$ states that each element of B is associated to an element of A, while injectivity of f^{-1} states that such element is unique; furthermore, each element of A is related to *at most* one element of B. Therefore, f can be thought of as being a partial injective function from A to B such that f^{-1} is an injective function. In the sequel, we often write $f(a) = b$ for $(a,b) \in f$.

We use *polyadic named functions* (p-named functions for short) for handling polyadicity:

Definition 17. *A* named set E *is a set E and a family of subset of names indexed by E, or equivalently $\mathsf{E}[\_]$ is a map from E to $\wp(\mathcal{N})$ (in the following, if X is a named set, then X denotes its underlying set).*

Let E and F be two named sets. A p-named function $\mathsf{m} : \mathsf{E} \to \mathsf{F}$ *is*

- *a function $m : E \to F$ and*
- *a family $\{\mathsf{m}[e] : \mathsf{E}[e] \multimap\!\!\rightarrow \mathsf{F}[f]\}_{(e,f)\in m}$ of name embeddings indexed by m such that, for any $e \in E$, $(a,b),(a,b') \in \mathsf{m}[e] \Longrightarrow b = b'$.*

A named function $\mathsf{m} : \mathsf{E} \to \mathsf{F}$ *is a p-named function such that, for each $e \in E$, $\mathsf{m}[e]$ is an inverse injection.*

Hereafter, if $\mathsf{m} : \mathsf{E} \to \mathsf{F}$ is a (p-)named function, m denotes the underlying mapping and, for any $e \in E$, $\mathsf{m}[e]$ is the embedding from $\mathsf{E}[e]$ to $\mathsf{F}[m(e)]$.

Definition 18. *A* Polyadic History Dependent Automaton with Negative Transitions *(pHDN), denoted by $\mathcal{A}$, is a tuple $\langle \mathsf{Q}, \mathsf{L}, \mathsf{T}, \overline{\mathsf{T}}, \mathsf{s}, \mathsf{d}, \mathsf{o}, q_0, \sigma_0 \rangle$, where*

- Q *is a named set of* states*;*
- L *is a named set of* labels*;*
- T *is a named set of (positive)* transitions*;*
- $\overline{\mathsf{T}}$ *is a named set of* negative transitions*;*
- $\mathsf{s}, \mathsf{d} : \mathsf{T} \uplus \overline{\mathsf{T}} \to \mathsf{Q}$ *are the* source *and* destination *named functions;*
- $\mathsf{o} : \mathsf{T} \uplus \overline{\mathsf{T}} \to \mathsf{L}$ *is the* observation *p-named function;*
- $q_0 \in Q$ *is the* initial state*;*
- $\sigma_0 : \mathcal{N} \hookleftarrow\!\!\!\rightharpoonup \mathsf{Q}[q_0]$ *is the* initial embedding *of the local names of q_0 into $\mathcal{N}$.*

We define $\mathsf{T}[t]_{\mathrm{old}} \stackrel{def}{=} \mathsf{T}[t] \cap \mathrm{dom}(\mathsf{s}[t])$ and $\mathsf{T}[t]_{\mathrm{new}} \stackrel{def}{=} \mathsf{T}[t] \setminus \mathrm{dom}(\mathsf{s}[t])$ and require that $\mathsf{T}[t]_{\mathrm{new}} \subseteq \mathrm{dom}(\mathsf{o}[t])$, for each $t \in T$. We use corresponding definitions for negative transitions.

We write $t : q \xrightarrow{l} q'$ (resp. $t : q \overset{l}{\rightsquigarrow} q'$) for denoting a positive (resp. negative) transition t such that $\mathsf{s}[t] = q$, $\mathsf{d}[t] = q'$ and $\mathsf{o}[t] = l$.

The source and destination functions associate states to each transition; they injectively embed the names of the source/destination states in the names of their transition. Similarly, the observation function associates a label to each transition, and embeds the names of the label in the names of the transition. It should be clear that Figure 1 fits Definition 18 if q_0 is the initial state and the initial embedding is $\{(u, v_1), (v, v_2)\}$. Notice that $\mathsf{s}[t]$ and $\mathsf{d}[t]$ must be inverse injections, whereas $\mathsf{o}[t]$ is an embedding of names of $\mathsf{T}[t]$ into names of the observation. Hence it is not required that $\mathsf{o}[t]$ be injective; for instance, transition t in Figure 1 embeds both label names v_1 and v_2 of t in v_1 thus representing the two occurrences of v in the corresponding transition of the agent P. The conditions on $\mathsf{s}[t]$ and $\mathsf{d}[t]$ mean that names of source and destination of t have at most one possible “meaning” along the transitions, while for $\mathsf{o}[t]$ multiple occurrences can appear in the observations [4].

Bisimulation on pHDNs must consider names of states, labels and transitions. We first give the concept of *partial bijection*:

Definition 19. *A relation $R \subseteq A \times B$ is a* partial bijection *between A and B (written $R : A \hookrightarrow B$) iff whenever $(a, b), (a', b') \in R$, then $a = a' \Leftrightarrow b = b'$.*

Bisimulations on pHDNs are relations $\mathcal{R}$ made of triples $\langle q_1, \delta, q_2 \rangle$ where q_1 and q_2 are the related states and $\delta : \mathsf{Q}[q_1] \hookrightarrow \mathsf{Q}[q_2]$ is a correspondence among names of q_1 and q_2. We say that $\mathcal{R}$ is *symmetric* if $\langle q_1, \delta, q_2 \rangle \in \mathcal{R}$ implies $\langle q_2, \delta^{-1}, q_1 \rangle \in \mathcal{R}$.

Definition 20. *Let $\mathcal{A} = \langle \mathsf{Q}, \mathsf{L}, \mathsf{T}, \overline{\mathsf{T}}, \mathsf{s}, \mathsf{d}, \mathsf{o}, q_0, \sigma_0 \rangle$ be a pHDN and $\mathcal{R} \subseteq \{\langle q_1, \delta, q_2 \rangle : q_1, q_2 \in Q, \delta : \mathsf{Q}[q_1] \hookrightarrow \mathsf{Q}[q_2]\}$ be a symmetric set of triples on $\mathcal{A}$. The set $\mathrm{red}[\mathcal{R}]$ of* redundant transitions *for $\mathcal{R}$ is the set of positive transitions $t \in \mathsf{T}$ such that there exist some negative transition $t' \in \overline{\mathsf{T}}$ such that $s(t) = s(t')$ and there are some $\xi : \mathsf{T}[t]_{\mathrm{new}} \leftrightarrow \overline{\mathsf{T}}[t']_{\mathrm{new}}$ and some $\zeta : \mathsf{T}[t] \hookrightarrow \overline{\mathsf{T}}[t']$ such that, if $\zeta = (\mathsf{s}[t]; \mathsf{s}[t']^{-1}) \cup \xi$, then*

$$\mathsf{o}[t] = \zeta; \mathsf{o}[t'] \quad \textit{and} \quad \langle \mathsf{d}[t], \delta', \mathsf{d}[t'] \rangle \in \mathcal{R}, \textit{ for some } \delta' \subseteq (\mathsf{d}[t]^{-1}; \zeta; \mathsf{d}[t']).$$

A transition which is not redundant (for $\mathcal{R}$) is said to be irredundant *(for $\mathcal{R}$).*

Definition 21. *Let $\mathcal{A} = \langle \mathsf{Q}, \mathsf{L}, \mathsf{T}, \overline{\mathsf{T}}, \mathsf{s}, \mathsf{d}, \mathsf{o}, q_0, \sigma_0 \rangle$ be a pHDN. A* pHDN-bisimulation *for $\mathcal{A}$ is a symmetric set of triples $\mathcal{R} \subseteq \{\langle q_1, \delta, q_2 \rangle : q_1 \in Q, q_2 \in Q, \delta : \mathsf{Q}[q_1] \hookrightarrow \mathsf{Q}[q_2]\}$ such that whenever $\langle q_1, \delta, q_2 \rangle \in \mathcal{R}$, then for each $t_1 : q_1 \xrightarrow{l} q_1' \notin \mathrm{red}[\mathcal{R}]$, there exist $t_2 : q_2 \xrightarrow{l} q_2'$, $\xi : \mathsf{T}[t_1]_{\mathrm{new}} \leftrightarrow \mathsf{T}[t_2]_{\mathrm{new}}$ and $\zeta : \mathsf{T}[t_1] \hookrightarrow \mathsf{T}[t_2]$ such that, if $\zeta = (\mathsf{s}[t_1]; \delta; \mathsf{s}[t_2]^{-1}) \cup \xi$, then*

$$\mathsf{o}[t_1] = \zeta; \mathsf{o}[t_2] \quad \textit{and} \quad \langle q_1', \delta', q_2' \rangle \in \mathcal{R}, \textit{ for some } \delta' \subseteq (\mathsf{d}[t_1]^{-1}; \zeta; \mathsf{d}[t_2])$$

We write $\mathcal{A}_1 \uplus \mathcal{A}_2$ for the pHDN obtained by taking the disjoint union of $\mathcal{A}_1$ and $\mathcal{A}_2$, and whose initial state coincides with the initial state of $\mathcal{A}_1$. (Of course, it does not matter whether we choose the initial state of $\mathcal{A}_1 \uplus \mathcal{A}_2$ to be the initial state of $\mathcal{A}_1$ or of $\mathcal{A}_2$.)

Definition 22. *Let $\mathcal{A}_i = \langle \mathsf{Q}_i, \mathsf{L}, \mathsf{T}_i, \overline{\mathsf{T}}_i, \mathsf{s}_i, \mathsf{d}_i, \mathsf{o}_i, q_{0_i}, \sigma_{0_i} \rangle$ ($i \in \{1, 2\}$) be two pHDNs. $\mathcal{A}_1$ and $\mathcal{A}_2$ are* pHDN-bisimilar, *written $\mathcal{A}_1 \sim \mathcal{A}_2$, if there exists some pHDN-bisimulation $\mathcal{R}$ for $\mathcal{A}_1 \uplus \mathcal{A}_2$ such that $\langle q_{0_1}, \delta, q_{0_2} \rangle \in \mathcal{R}$ for some $\delta \subseteq (\sigma_{0_1}; \sigma_{0_2}^{-1})$.*

[4] In [10] this problem is resolved by using labels of the form $\mathbf{out}_2$ that exposes a single name which plays the rôle of subject and object in the transition. We cannot exploit this mechanism for general polyadicity, because if $\tilde{x}$ are the names of a transition, we should consider a different label for each subset of names of $\tilde{x}$. Even though this could be a valid solution, it is not suitable as far as verification purposes are concerned. Indeed, it would require a different transition for each subset of the set names carried by the label. This makes the number of states of the underlying automata exponential.

γ	$\mathbf{1}$	$(\tilde{z})u\,\tilde{x}$				$(\tilde{z})\overline{u}\,\tilde{x}$				$\{\tilde{x} = \tilde{z}\}$					
$l \in \mathrm{L}^f$	$\mathtt{tau}$	$\mathtt{in}\, b_1 \cdots b_m$				$\mathtt{out}\, b_1 \cdots b_m$				$\mathtt{fus}_m$					
$\lambda \in \mathrm{L}^f[l]$	$-$	u	x_1	$\cdots$	x_m	u	x_1	$\cdots$	x_m	x_1	$\cdots$	x_m	z_1	$\cdots$	z_m
$\kappa(n) \in \mathrm{L}^f[l]$	$-$	v_n	v_{n+1}	$\cdots$	v_{n+m}	v_n	v_{n+1}	$\cdots$	v_{n+m}	v_n	$\cdots$	v_{n+m+1}	v_{n+m+2}	$\cdots$	v_{2n+m}

Table 3. Labels for fusion calculus pHDN ($m = |\tilde{x}|$)

6 Mapping Fusion Calculus to pHDN

The mapping of fusion calculus agents to pHDN is similar to the mapping of constrained π-calculus agents in the open semantics [10]; like there, the closure under substitutions is handled by matching the condition part of the label in bisimulation, and the negative transitions of the pHDN are used to "mark" possibly redundant transitions. Differently from [10], we do not need neither distinctions nor to split a symbolic transition $\xmapsto{M,\gamma}$ into two transitions at HD-automata level (one for condition M and one for action γ). Our polyadic extension of HDN allows conditions and actions to be combined in the same transition.

Hereafter, we consider only *representative transitions*, defined as follows:

Definition 23. *A transition $P \xmapsto{M,\gamma} Q$ is a* representative transitions *iff, assuming a total order on $\mathcal{N}$, either γ is a fusion or $\gamma \in \{(z_1 \dots z_m)\overline{u}\,\tilde{x}, (z_1 \dots z_m)u\,\tilde{x}\}$ and, for all $i = 1, \dots, m$, $z_i = min\{\mathcal{N} \setminus \mathrm{fn}(P) \setminus \{z_j : 1 \leq j < i\}\}$.*

In the following, we assume a function norm(_) such that, for any fusion calculus agent A, norm(A) yields the pair (B, ρ) where

– B is the representative element of the class of agents differing from A by bijective substitutions,
– $\rho : \mathrm{fn}(A) \to \mathrm{fn}(B)$ is a bijective substitution such that $A\rho = B$.

Basically, norm(_) renames agents exploiting the total order assumed on $\mathcal{N}$. To fix the terminology, we assume that $v_1 < v_2 \cdots < v_n \cdots$, where, for each i, v_i is a name in $\mathcal{N}$. For instance, let us again consider the agent of Section 4, $P = (y)(\overline{u}\,vy\,.\,B(u, y) + [u = v]\overline{v}\,vy\,.\,B(y, y))$; norm$(P) = (q_0, \sigma_0)$, where $q_0 = (y)(\overline{v_1}\,v_2 y\,.\,B(v_1, y) + [v_1 = v_2]\overline{v_2}\,v_2 y\,.\,B(y, y))$ and $\sigma_0 = \{(u, v_1), (v, v_2)\}$.

Table 3 gives the correspondence between labels of the fusion calculus and pHDN labels.

The first row reports the labels of fusion calculus, for each of these, the second row establishes the correspondence between the label of the HD-automata. The set of labels for pHDN of fusion calculus agents is

$$L^f = \{\mathtt{tau}\} \cup \{\alpha\, b_1 \cdots b_m : \alpha \in \{\mathtt{in}, \mathtt{out}\} \wedge \forall i = 1, \dots, m : b_i \in \{1, 0\}\} \cup \{\mathtt{fus}_m : m > 0\}.$$

Label $\mathtt{tau}$ is used for identity fusions; $\mathtt{in}\, b_1 \cdots b_m$ and $\mathtt{out}\, b_1 \cdots b_m$ correspond to input and output of m names, say $x_1 \cdots x_m$; each x_i is free or bound depending whether $b_i = 1$ or $b_i = 0$; $\mathtt{fus}_m$ corresponds to a fusion action $\{x_1 = z_1, \dots, x_m = z_m\}$.

The correspondence between the names of the occurring in the fusion calculus labels and those of the HD-automata are given the last two rows. Such correspondence is defined in terms of two embeddings κ and λ; the former depends on a parameter n which is the number (incremented by 1) of names of the condition part of transitions not included in the action part names of the fusion calculus agents. It is used to let the names (of canonical representatives) of pHDN transitions be different from the names used in the conditions; $\kappa(n)$ is not required to be injective. Basically, λ and κ define the embedding of names of the transitions of agents into the (normalised) names of the transitions of the HD-automata.

Definition 24. *The pHDN $\mathcal{A}^f(A)$ corresponding to the fusion calculus calculus agent A is the smallest pHDN that satisfies the following rules:*

1. Initial state and embedding*: if* $\mathrm{norm}(A) = (A', \sigma')$ *then*
   – $A' \in Q$ *is the initial state, and* $\mathsf{Q}[A'] = \mathrm{fn}(A')$*;*
   – σ' *is the initial embedding;*
2. Positive transitions*: if* $A \in Q$*,* $A \xmapsto{M,\gamma} A'$ *is a representative transition,* $\mathrm{norm}(A') = (A'', \sigma)$*,* N *is a normalised condition, and* $N \Leftrightarrow M\rho$ *for some bijective substitution* ρ*, then*
   – $A'' \in Q$ *and* $\mathsf{Q}[A''] = \mathrm{fn}(A'')$*;*
   – *there is some* $t \in T$ *such that* $\mathsf{T}[t] = \mathrm{fn}(A) \cup \mathrm{n}(M) \cup \mathrm{bn}(\gamma)$*;*
   – $s(t) = A$*,* $d(t) = A''$*,* $\mathsf{s}[t] = \mathrm{id}_{\mathrm{fn}(A)}$ *and* $\mathsf{d}[t] = \sigma$*;*
   – $o(t) = \langle N, l\rangle$ *and* $\mathsf{o}[t] = \rho \cup \kappa(n)$ *where* $n = 1 + |\mathrm{n}(N) \setminus \mathrm{n}(\gamma)|$*, while* l *and* κ *are related to* γ *as defined in Table 3;*
3. Negative transitions*: if* $A \in Q$*,* $A \xmapsto{M,\gamma} A'$ *and* $A \xmapsto{M',\gamma'} A''$ *are representative transition such that* $M \not\Longleftarrow\!\!\Longrightarrow M'$ *and* $\gamma = \gamma'\sigma_M$*; let* $\mathrm{norm}(A''\sigma_M) = (A''', \sigma)$*,* N *be a normalised condition such that* $N \Leftrightarrow M\rho$ *for some bijective substitution* ρ*, then*
   – $A''' \in Q$ *and* $\mathsf{Q}[A''] = \mathrm{fn}(A''')$*;*
   – *there is some* $t \in \overline{T}$ *such that* $\overline{\mathsf{T}}[t] = \mathrm{fn}(A) \cup \mathrm{n}(M) \cup \mathrm{bn}(\gamma)$*;*
   – $s(t) = A$*,* $d(t) = A'''$*,* $\mathsf{s}[t] = \mathrm{id}_{\mathrm{fn}(A)}$ *and* $\mathsf{d}[t] = \sigma_M\sigma$*;*
   – $o(t) = \langle N, l\rangle$ *and* $\mathsf{o}[t] = \rho \cup \kappa(n)$ *where* $n = 1 + |\mathrm{n}(N) \setminus \mathrm{n}(\gamma)|$*, while* l *and* κ *are related to* γ *as defined in Table 3.*

We use the agent P given in Section 4 to comment on Definition 24; we build the automaton following the steps of Definition 24.

1. Initial state and initial embedding are given by $\mathrm{norm}(P) = (q_0, \sigma_0)$ previously detailed.
2. There are two possible representative transitions out of q_0, namely $q_0 \xmapsto{v_1=v_2,(v_3)\overline{v_2}\,v_2v_3} B(v_3, v_3)$ and $q_0 \xmapsto{\emptyset,(v_3)\overline{v_1}\,v_2v_3} B(v_1, v_3)$, so if $(q_1, \sigma_1) = \mathrm{norm}(B(v_3, v_3))$ and $(q_2, \sigma_2) = \mathrm{norm}(B(v_1, v_3))$ then $\sigma_1(v_3) = v_1$ and $\sigma_2(v_3) = v_2$ as shown in Figure 1. As prescribed by Definition 24, we add two positive transition, t and t''. Transition t goes from q_0 to q_1; its names are $\mathrm{fn}(q_0) \cup \mathrm{n}(v_1 = v_2) \cup \mathrm{bn}((v_3)\overline{v_2}\,v_2v_3) = \{v_1, v_2, v_3\}$, moreover, $\mathsf{s}[t] = \mathrm{id}_{\mathrm{fn}(q_0)}$ and $\mathsf{d}[t] = \sigma_1$. Finally, the observation of t is obtained through Table 3. Indeed, $o(t) = \langle v_1 = v_2, \mathtt{out}\,001\rangle$; since v_1 and v_2 appear in the action part of the label, $\kappa(1) = \{(v_1, v_1), (v_1, v_2), (v_2, v_3)\}$, as reported in Table 3, which in Figure 1 corresponds to the dashed arrows from names of t to names of the observation.
3. To complete the automaton, it remains to add the negative transitions. Let us observe that, by definition, the condition of t strictly implies the condition of t'', i.e. $u = v \not\Longleftarrow\!\!\Longrightarrow \emptyset$; moreover, $\overline{u}\,vy\sigma_{u=v} = \overline{v}\,vy$ and $\mathrm{norm}(B(u, v)\sigma_{u=v}) = (q_2, id)$. Thus, we add negative transition t' to the automaton, where the name embeddings are computed as done above for positive transitions t and t''.

7 Relating pHDN-bisimulation and Hyperequivalence

This section shows the relationship between hyperequivalence and pHDN-bisimulation formally stated in Theorem 29: The HD-automata corresponding to bisimilar fusion calculus agents are pHDN-bisimilar and, conversely, we can recover an agent from a state of an automaton once a "global" identity is assigned to the local names of the state.

Definition 25. *Given a pHDN automaton* $\mathcal{A} = \langle \mathsf{Q}, \mathsf{L}, \mathsf{T}, \overline{\mathsf{T}}, \mathsf{s}, \mathsf{d}, \mathsf{o}, q_0, \sigma_0\rangle$*, a* global state of $\mathcal{A}$ *is a pair* (q, σ) *where* q *is a state of* $\mathcal{A}$ *and* $\sigma : \mathcal{N} \longleftrightarrow \mathsf{Q}[q]$*.* $\mathsf{G}_{\mathcal{A}}$ *is the named set of global states of* $\mathcal{A}$*, where* $\mathsf{G}_{\mathcal{A}}[(q, \sigma)] = \sigma^{-1}(\mathsf{Q}[q])$*.*

A global transition of $\mathcal{A}$ *is a pair* (t, ρ) *where* t *is a transition of* $\mathcal{A}$ *and* $\rho : \mathcal{N} \multimap \mathsf{T}[t]$ *(*$\rho : \mathcal{N} \multimap \overline{\mathsf{T}}[t]$ *if* t *is a negative transition) is s.t.* $(n, x), (n, x') \in \rho \iff \mathsf{o}[t](x) = \mathsf{o}[t](x')$*.* $\mathsf{U}_{\mathcal{A}}$ *is the named set of global transition of* $\mathcal{A}$ *where* $\mathsf{U}_{\mathcal{A}}[(t, \rho)] = \rho^{-1}(\mathsf{T}[t])$*. Moreover, we let* $\mathsf{U}_{\mathcal{A}}[(t, \rho)]_{\mathrm{old}} = \rho(\mathsf{T}[t]_{\mathrm{old}})$ *(and similarly for* $\mathsf{U}_{\mathcal{A}}[(t, \rho)]_{\mathrm{new}}$*).*

If $t : q \xrightarrow{l} q'$ *then* $(t, \rho) : (q, \rho; \mathsf{s}[t]) \xrightarrow{l,\lambda} (q', \rho; \mathsf{d}[t])$ *is s.t.* $\lambda^{-1} = \rho \cup \mathsf{o}[t]$ *and* λ *and* l *are related as in Table 3 (and similarly for negative transitions).*

A global state (q, σ) represents the image of state q of a pHDN automaton once a global identity σ has been fixed for the names of q (and analogously for global transitions).

Theorem 29 relies on the fact that pHDN-bisimulation is equivalent to bisimulation on global states and global transitions that is defined as follows:

Definition 26. *Let* $\mathcal{R}$ *be a binary relation over global states of a pHDN-automaton* $\mathcal{A}$*. A positive global transition* $t : q \xrightarrow{l} q'$ *is* redundant for $\mathcal{R}$ *if there exists a global negative transition* $t' : q \overset{l}{\rightsquigarrow} q''$ *s.t.* $q' \mathrel{\mathcal{R}} q''$*.*

A symmetric binary relation $\mathcal{R}$ *over global states of* $\mathcal{A}$ *is a* global bisimulation *if, whenever* $q_1 \mathrel{\mathcal{R}} q_2$*, for each non-redundant transition* $t_1 : q_1 \xrightarrow{l} q'$ *for* $\mathcal{R}$*, there exists a global transition* $t_2 : q_2 \xrightarrow{l} q'_2$ *s.t.* $\mathsf{U}_1[t_1]_{\text{new}} \cap \mathsf{G}_2[t_2] = \emptyset$ *and* $q'_1 \mathrel{\mathcal{R}} q'_2$*.*

Theorem 27. *Two pHDN-automata are pHDN-bisimilar iff they are globally bisimilar.*

The proof of Theorem 27 is given in the appendix. The intuition is that for the "only if" part is to prove that $\mathcal{R}' \stackrel{\text{def}}{=} \{ \langle q_1, \delta, q_2 \rangle : (q_1, \sigma_1)\mathcal{R}(q_2, \sigma_2), \text{ where } \delta = \sigma_1^{-1}; \sigma_2\}$ is a pHDN-bisimulation, if $\mathcal{R}$ is a global bisimulation for $\mathcal{A}_1$ and $\mathcal{A}_2$; while for the "if" part of the theorem we can show that $\mathcal{R}' \stackrel{\text{def}}{=} \{ \langle q_1, \delta, q_2 \rangle : (q_1, \sigma_1) \mathcal{R}(q_2, \sigma_2), \text{ where } \delta = \sigma_1^{-1}; \sigma_2\}$ is a pHDN-bisimulation, provided that $\mathcal{R}$ is a global bisimulation for $\mathcal{A}_1$ and $\mathcal{A}_2$.

Lemma 28. *Given fusion calculus agent* P *and a global state* (q, σ) *of* $\mathcal{A}^f(P)$*, then:*

1. *if* $q\sigma \xmapsto{M,\gamma} q''$ *is a transition in* $\mathcal{A}^f(P)$*, then there is a global transition* $(t, \rho) : (q, \sigma) \xrightarrow{l,\lambda} (q', \sigma')$ *of* $\mathcal{A}^f(P)$ *s.t.* $q'' = q'\sigma'$ *and* l *and* λ *are related as in Table 3;*
2. *vice versa, if* $(t, \rho) : (q, \sigma) \xrightarrow{l,\lambda} (q', \sigma')$ *is a global transition of* $\mathcal{A}^f(P)$*, then there is a transition* $q\sigma \xmapsto{M,\gamma} q''$ *s.t.* $q'' = q'\sigma'$ *and* l *and* λ *are related as in Table 3.*

Proof of 1: By Lemma 12(2), we have (without loss of generality) a representative transition $q \xmapsto{M',\gamma'} q'''$ where $M \Leftrightarrow M'\sigma$, $\gamma = \gamma'\sigma$, $q'' = q'''\sigma_M\sigma_\gamma$. Then, letting $(q', \sigma'') = \text{norm}(q''')$, by Definition 24, $t : q \xmapsto{N,l} q'$ is a transition of $\mathcal{A}^f(P)$, where, for a bijective substitution θ,

$$N \Leftrightarrow M'\theta, \qquad \mathsf{o}[t] = \theta \cup \kappa(|\text{n}(N)|), \qquad \mathsf{d}[t] = \sigma'', \qquad \mathsf{s}[t] = id_{\text{fn}(q)},$$

and l and κ are related as in Table 3. If we let $\sigma' = \sigma''^{-1}; \sigma_{M'}; \sigma_\gamma$ and consider an embedding ρ s.t. $\rho|_{\text{dom}(\sigma)} = \sigma$ and $\rho^{-1}|_{\text{n}(N)} = \theta^{-1}$ then it is trivial to see that $(t, \rho) : (q, \sigma) \xrightarrow{l,\lambda} (q', \sigma')$ is a global transition for $\mathcal{A}^f(P)$. Finally, since by construction of σ' and $q' = q'''\sigma''$, we have $q'\sigma' = q'''\sigma''; \sigma''^{-1}; \sigma_{M'}; \sigma_\gamma = q'''\sigma_{M'}; \sigma_\gamma = q''$ (where the last equality holds by Lemma 12(2)).

Proof of 2: Let $(t, \rho) : (q, \sigma) \xrightarrow{l,\lambda} (q', \sigma')$ be a global transition for $\mathcal{A}^f(P)$ then, by construction, there is a transition $t : q \xmapsto{M,l} q'$ in $\mathcal{A}^f(P)$ (where M, l and λ are related as in Table 3); hence, by Definition 24, there is a (representative) transition s.t., for a bijective substitution θ, $q \xmapsto{N,\gamma} q'''$ where $N \Leftrightarrow M\theta$, $(q', \sigma'') = \text{norm}(q''')$ and $\sigma'' = \mathsf{d}[t]$. Then, by Lemma 12(1), $q\sigma \xmapsto{M,\gamma} q'''\sigma$, where $q'' = q'''\sigma\sigma_M\sigma_\gamma\sigma$ and proceeding as in the first part, we have that $q'\sigma' = q''$. □

Theorem 29. $A_1 \simeq A_2$ *iff* $\mathcal{A}^f(A_1) \sim \mathcal{A}^f(A_2)$*, where* A_1 *and* A_2 *are two fusion calculus agents.*

Proof. The proof shows that $A_1 \simeq A_2$ iff the corresponding automaton are global bisimilar. This, by Theorem 27, proves the result.

By Lemma 28(1) we have that the relation

$$\mathcal{R} = \{\langle (q_1, \sigma_1), (q_2, \sigma_2) : (q_i, \sigma_i) \text{ global state of } \mathcal{A}^f(A_i), i \in \{1, 2\} \ \wedge \ q_1\sigma_1 \simeq q_2\sigma_2 \rangle\} \qquad (1)$$

is a global bisimulation and, by construction of $\mathcal{A}^f(A_1)$ and $\mathcal{A}^f(A_2)$, if (q_{i_0}, σ_{i_0}) is the initial state of $\mathcal{A}^f(A_i)$ ($i \in \{1, 2\}$), then

$$q_{1_0}\sigma_{1_0} = A_1 \simeq A_2 = q_{2_0}\sigma_{2_0},$$

hence $\langle (q_{1_0}, \sigma_{1_0}), (q_{2_0}, \sigma_{2_0}) \rangle \in \mathcal{R}$, which, by Definition 22, means that $\mathcal{A}^f(A_1)$ is bisimilar to $\mathcal{A}^f(A_1)$.

The converse is similarly proved by using Lemma 28(2). Indeed, if $\mathcal{R}$ is global bisimulation which contains $\langle (q_{1_0}, \sigma_{1_0}), (q_{2_0}, \sigma_{2_0}) \rangle$ then the relation

$$\mathcal{S} \stackrel{\text{def}}{=} \{(q_1\sigma_1, q_2\sigma_2) : (q_1, \sigma_1)\mathcal{R}(q_2, \sigma_2)\}$$

is a bisimulation containing (A_1, A_2). □

8 Conclusion

We developed a new symbolic semantics for the fusion calculus where *canonical* substitutions are applied eagerly, lessening the computational work in bisimulation checking, and which fits well in the HD framework with negative transitions. We made a conservative extension (pHDN) of history dependent automata to handle *polyadicity*. This extension is applicable also to π-calculus semantics. For the open semantics of π, it also reduces the number of states and transitions in the corresponding HD-automata by avoiding the split into condition and action transitions. To simplify the presentation (avoiding the distinctions of open semantics of π), and to show that the substitutive effects of the fusion calculus do not present a problem for pHDN, we gave a mapping of fusion calculus into pHDN, and proved its adequacy. Similar mappings for the closely related π_F-calculus [4] or the χ-calculus [3] should be straightforward.

As future work, we will apply our results in a partition refinement-based equivalence checking tool within the PWEB, a web-based verification environment under development in the PROFUNDIS FET-GC project. In addition, we intend to give an explicit co-algebraic formulation of pHDN where an effective minimization algorithm can be guaranteed by the existence of a terminal object, like for the π-calculus with early semantics [1]. We also intend to extend HD-automata with a substitution operation on the states in order to model full congruences for late and early π semantics.

References

[1] G. Ferrari, U. Montanari, and M. Pistore. Minimizing transition systems for name passing calculi: A co-algebraic formulation. In M. Nielsen and U. Engberg, editors, *FOSSACS 2002*, volume 2303 of *LNCS*, pages 129–143. Springer, 2002.

[2] G. Ferrari, U. Montanari, R. Raggi, and E. Tuosto. From co-algebraic specifications to implementation: The Mihda toolkit. In *International Symposium on Formal Methods for Components and Objects*, LNCS. Springer-Verlag, 2003. To appear.

[3] Y. Fu. A proof-theoretical approach to communication. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proceedings of ICALP '97*, volume 1256 of *LNCS*, pages 325–335. Springer, 1997.

[4] P. Gardner and L. Wischik. Explicit fusions. In Nielsen and Rovan [8], pages 373–382.

[5] C. Laneve and B. Victor. Solos in concert. In J. Wiederman, P. van Emde Boas, and M. Nielsen, editors, *Proceedings of ICALP '99*, volume 1644 of *LNCS*, pages 513–523. Springer, July 1999.

[6] U. Montanari and M. Pistore. History dependent automata. Technical report, Computer Science Department, Università di Pisa, 1998. TR-11-98.

[7] U. Montanari and M. Pistore. pi-calculus, structured coalgebras, and minimal HD-automata. In Nielsen and Rovan [8].

[8] M. Nielsen and B. Rovan, editors. *Mathematical Foundations of Computer Science, 25th International Symposium, MFCS 2000 (Bratislava, Slovakia)*, volume 1893 of *LNCS*. Springer, 2000.

[9] J. Parrow and B. Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. In *Proceedings of LICS '98*, pages 176–185. IEEE, Computer Society Press, July 1998.

[10] M. Pistore. *History Dependent Automata*. PhD thesis, Dipartimento di Informatica, Università di Pisa, Mar. 1999.

[11] M. Pistore and D. Sangiorgi. A partition refinement algorithm for the π-calculus. *Information and Computation*, 164(2):264–321, 2001. An extended abstract appeared in *Proceedings of CAV'96*, LNCS 1102.

[12] D. Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33:69–97, 1996. Earlier version published as Report ECS-LFCS-93-270, University of Edinburgh. An extended abstract appeared in the *Proceedings of CONCUR '93*, LNCS 715.

[13] P. Taylor. *Practical Foundations of Mathematics*. Cambridge University Press, 1999.

A Proofs

A.1 Proofs from Section 3

Proof of Lemma 11:

1. See [12].
2. Follows from the following lemmas of Sangiorgi [12]:
   (a) if $M \Rightarrow N$ then $M\sigma \Rightarrow N\sigma$
   (b) if $\sigma \Rightarrow \sigma'$ then ρ exists s.t. $\sigma = \sigma'\rho$
3. Follows from the Substitution Lemma in Taylor [13].

Proof of Lemma 12: We show the case for S-PAR; the others are similar.

1. $P \equiv P_1 \mid P_2 \xmapsto{M,\gamma} P_1' \mid P_2' \equiv Q$ where $P_1 \xmapsto{M,\gamma} P_1'$ and $P_2' \equiv P_2\sigma_M\sigma_\gamma$.
   Then by induction $P_1\sigma \xmapsto{M\sigma,\gamma\sigma} P_1'\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma}$ and by S-PAR $P_1\sigma \mid P_2\sigma \xmapsto{M\sigma,\gamma\sigma} P_1'\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma} \mid P_2\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma}$.
   Let $P' = P_1'\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma}$,

$$
\begin{array}{ll}
P' \mid P_2\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma} = & \\
\quad = P' \mid P_2\sigma_M\sigma_\gamma\rho & \text{(by Lemma 11(1), for some } \rho \text{ s.t. } \sigma\sigma_{M\sigma}\sigma_{\gamma\sigma} = \sigma_M) \\
\quad = P' \mid P_2\sigma_M\sigma_M\sigma_\gamma\sigma_\gamma\rho & \text{(idempotence)} \\
\quad = P' \mid P_2\sigma_M\sigma_M\sigma_{\gamma\sigma_M}\sigma_\gamma\rho & (\gamma = \gamma\sigma_M \text{ by Lemma 9)} \\
\quad = P' \mid P_2\sigma_M\sigma_\gamma\sigma_M\sigma_\gamma\rho & \text{(by Lemma 11(3))} \\
\quad = P' \mid P_2'\sigma_M\sigma_\gamma\rho & (P_2' = P_2\sigma_M\sigma_\gamma) \\
\quad = P_1'\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma} \mid P_2'\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma} & \text{(using } \rho, \sigma \text{ from first step above)} \\
\quad = (P_1' \mid P_2')\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma} & \text{(which is what is required)}
\end{array}
$$

2. Let $P \equiv (P_1 \mid P_2)\sigma \xmapsto{N,\gamma'} P_1' \mid P_2' \equiv Q$ where $P_1\sigma \xmapsto{N,\gamma'} P_1'$ and $P_2' = P_2\sigma\sigma_N\sigma_\gamma'$
   Then by induction hypothesis we have $P_1 \xmapsto{M,\gamma} P_1''$ with $P_1' = P_1''\sigma\sigma_N\sigma_{\gamma'}$, $\gamma\sigma = \gamma'$ and $M\sigma \Leftrightarrow N$ and by S-PAR $P_1 \mid P_2 \xmapsto{M,\gamma} P_1'' \mid P_2'' \equiv P_3$ with $P_2'' \equiv P_2\sigma_M\sigma_\gamma$. We only need to show that $Q = P_3\sigma\sigma_N\sigma_{\gamma'}$.

$$
\begin{array}{lll}
Q & = P_1' \mid P_2' & \\
 & = P_1' \mid P_2\sigma\sigma_N\sigma_{\gamma'} & \\
 & = P_1' \mid P_2\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma} & (\gamma' = \gamma\sigma \text{ and } N \equiv M\sigma) \\
 & = P_1' \mid P_2\sigma_M\sigma_\gamma\rho & \text{(by Lemma 11(1), for some } \rho \text{ s.t. } \sigma\sigma_{M\sigma}\sigma_{\gamma\sigma} = \sigma_M) \\
 & = P_1' \mid P_2\sigma_M\sigma_\gamma\sigma_M\sigma_\gamma\rho & (\gamma = \gamma\sigma_M\text{, Lemma 11(3) and idempotence)} \\
 & = P_1' \mid P_2''\sigma_M\sigma_\gamma\rho & (P_2'' = P_2\sigma_M\sigma_\gamma) \\
 & = P_1' \mid P_2''\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma} & \text{(using } \rho, \sigma \text{ from earlier step)} \\
 & = P_1''\sigma\sigma_N\sigma_{\gamma'} \mid P_2''\sigma\sigma_N\sigma_{\gamma'} & (\gamma' = \gamma\sigma \text{ and } N \Leftrightarrow M\sigma) \\
 & = (P_1'' \mid P_2'')\sigma\sigma_N\sigma_{\gamma'} = P_3\sigma\sigma_N\sigma_{\gamma'} &
\end{array}
$$

Proof of Lemma 13: We show that

$$\mathcal{S} = \bigcup_\sigma \{(P\sigma, Q\sigma) : P \simeq Q\}$$

is a symbolic hyperbisimulation. By Lemma 12(2) an action of $P\sigma$ can be written as $P\sigma \xmapsto{M\sigma,\gamma\sigma} P''$ for some M, γ and P' s.t. $P \xmapsto{M,\gamma} P'$ and $P'' = P'\sigma\sigma_{M\sigma}\sigma_{\gamma\sigma}$. Note that by Lemma 9, $\gamma\sigma = \gamma\sigma\sigma_{M\sigma}$. Since

$P \simeq Q$, we have $Q \overset{N,\delta}{\longmapsto} Q'$ with

$$M \Rightarrow N \tag{2}$$
$$\gamma = \delta\sigma_M \tag{3}$$
$$P' \simeq Q'\sigma_M \tag{4}$$

Using Lemma 12(1), we have $Q\sigma \overset{N\sigma,\delta\sigma}{\longmapsto} Q''\sigma\sigma_{N\sigma\delta\sigma}$. We only need to show the following three equations in order to show this is a matching action of $P\sigma$.

$$M\sigma \Rightarrow N\sigma \tag{5}$$
$$\gamma\sigma = \delta\sigma\sigma_{M\sigma} \tag{6}$$
$$P'\sigma \ \mathcal{S} \ Q'\sigma\sigma_{M\sigma} \tag{7}$$

Equation (5) follows from (2) and item (b) in the proof of Lemma 11(2).

$$\begin{array}{rrcll} & \gamma & = & \delta\sigma_M & \text{(from (3))} \\ \Leftrightarrow & \gamma\sigma_M & = & \delta\sigma_M & (\gamma = \gamma\sigma_M \text{ by Lemma 9}) \\ \Rightarrow & \gamma\sigma_M\rho & = & \delta\sigma_M\rho & \text{(for some } \rho \text{ s.t. } \sigma\sigma_{M\sigma}\sigma_{\gamma\sigma} = \sigma_M\rho \text{ by Lemma 11(1))} \\ \Leftrightarrow & \gamma\sigma\sigma_{M\sigma} & = & \delta\sigma\sigma_{M\sigma} & \text{(by Lemma 11(1))} \\ \Leftrightarrow & \gamma\sigma & = & \delta\sigma\sigma_{M\sigma} & (\text{since } \gamma\sigma = \gamma\sigma\sigma_{M\sigma}) \end{array}$$

Equation (7) follows from (3) by similar reasoning.

Proof of Lemma 14:

1. The proof is by transition induction. We show the case for S-COM, the others are similar.

$$\frac{P_1 \overset{M_1,u\tilde{x}}{\longmapsto} P_1',\ \ P_2 \overset{M_2,\overline{v}\tilde{y}}{\longmapsto} P_2',\ \ |\tilde{x}| = |\tilde{y}|,\ \ L = M_1M_2[u=v],\ \varphi = \{\tilde{x} = \tilde{y}\}\sigma_L}{P_1 \mid P_2 \overset{L,\varphi}{\longmapsto} (P_1' \mid P_2')\sigma_L\sigma_\varphi}$$

   From the induction hypothesis we have

$$P_1\sigma_{M_1} \xrightarrow{(\overline{u}\tilde{x})\sigma_{M_1}} P_1'' \qquad P_2\sigma_{M_2} \xrightarrow{(v\tilde{y})\sigma_{M_2}} P_2''$$

   with $P_1'' = P_1'\sigma_{\overline{u}\tilde{x}} = P_1'$ and $P_2'' = P_2'\sigma_{u\tilde{y}} = P_2'$.
   Using Lemma 2, σ_L can be decomposed into $\sigma_{M_1}\rho_1$ and into $\sigma_{M_2}\rho_2$ for some ρ_1, ρ_2. Using these and Lemma 6

$$\frac{P_1\sigma_L \xrightarrow{(\overline{a}y)\sigma_L} P_1'\sigma_L \qquad P_2\sigma_L \xrightarrow{(\overline{a}y)\sigma_L} P_2'\sigma_L}{(P_1 \mid P_2)\sigma_L \xrightarrow{\varphi} (P_1'\sigma_L \mid P_2'\sigma_L) = (P_1' \mid P_2')\sigma_L}$$

2. This proof is also by transition induction. We give the proof for the case of S-PAR.

$$\text{PAR}\ \frac{P_1\sigma_M \overset{\gamma}{\longmapsto} P_1'}{(P_1 \mid P_2)\sigma_M \overset{\gamma}{\longmapsto} P_1' \mid P_2\sigma_M}$$

   From the induction hypothesis we have;

$$\text{S-PAR}\ \frac{P_1 \overset{N,\gamma'}{\longmapsto} P_1''}{P_1 \mid P_2 \overset{N,\gamma'}{\longmapsto} P_1'' \mid P_2\sigma_N\sigma_{\gamma'}}$$

(Notice that $P_1'' = P_1''\sigma_N\sigma'_\gamma$ by Lemma 9.) But

$$\begin{aligned}(P_1'' \mid P_2\sigma_N\sigma_{\gamma'})\sigma_M &= (P_1''\sigma_N\sigma_{\gamma'} \mid P_2\sigma_N\sigma_{\gamma'})\sigma_M\\ &= (P_1'' \mid P_2)\sigma_N\sigma_{\gamma'}\sigma_M\\ &= (P_1'' \mid P_2)\sigma_N\sigma_M\sigma_{\gamma'\sigma_M}\\ &= (P_1' \mid P_2)\sigma_M\sigma_{\gamma'\sigma_M}\\ &= (P_1' \mid P_2)\sigma_M\sigma_\gamma\end{aligned}$$

The proof of the rest cases are similar.

Proof of Theorem 15:
($\Rightarrow$): We show that

$$\mathcal{S} = \{(P, Q) : P \sim Q\}$$

is a symbolic hyperbisimulation.

Suppose $P \xmapsto{M,\gamma} P'$. By Lemma 14(1), we have $P\sigma_M \xrightarrow{\gamma} P''$ such that $P' = P''\sigma_\gamma$. From $P \sim Q$ and Definition 7, we have $P\sigma_M \sim Q\sigma_M$, so $Q\sigma_M \xrightarrow{\gamma} Q''$ with $P''\sigma_\gamma \sim Q''\sigma_\gamma$. By Lemma 14(2), $Q \xmapsto{N,\delta} Q'$ with $M \Rightarrow N$, $\delta\sigma_M = \gamma$ and $Q'\sigma_M = Q''\sigma_\gamma$.

We have $P''\sigma_\gamma \sim Q''\sigma_\gamma$, so $P''\sigma_\gamma\sigma_M \sim Q''\sigma_\gamma\sigma_M$ by Definition 7. $P''\sigma_\gamma\sigma_M = P''\sigma_M\sigma_{\gamma\sigma_M} = P''\sigma_M\sigma_\gamma = P'$, and $Q''\sigma_\gamma\sigma_M = Q'\sigma_M\sigma_M = Q'\sigma_M$.

So finally $(P', Q'\sigma_M) \in \mathcal{S}$ which closes up the bisimulation.

($\Leftarrow$): We already have closure under substitution (Lemma 13), and only need to show that

$$\mathcal{S} = \{(P, Q) : P \simeq Q\}$$

is a fusion bisimulation. Suppose $P \xrightarrow{\alpha} P'$. By Lemma 14(2) we have $P \xmapsto{\emptyset,\alpha} P'$. Since $P \simeq Q$, we have $Q \xmapsto{\emptyset,\alpha} Q' \simeq P'$. From Lemma 14(1), $Q \xrightarrow{\alpha} Q'$, which closes up the bisimulation. □

A.2 Proofs from Section 7

Proof of Theorem 27: We prove the "only if" part by showing that

$$\mathcal{R}' \stackrel{\text{def}}{=} \{\langle (q_1, \sigma_1), (q_2, \sigma_2)\rangle | \langle q_1, \delta, q_2\rangle \in \mathcal{R},\ \text{where}\ \delta = \sigma_1^{-1}; \sigma_2\}$$

is a global bisimulation, provided that $\mathcal{R}$ is a pHDN-bisimulation.

Assume that $(q_1, \sigma_1)\mathcal{R}'(q_2, \sigma_2)$ and $(t_1, \rho_1) : (q_1, \sigma_1) \xrightarrow{l,\lambda} (q_1', \sigma_1')$ is a non-redundant global transition s.t. $\rho_1^{-1}(\mathsf{T}_1[t_1]) \cap \sigma_2^{-1}(\mathsf{Q}_2[q_2]) = \emptyset$. We must find a transition $(t_2, \rho_1) : (q_2, \sigma_2) \xrightarrow{l,\lambda} (q_2', \sigma_2')$ with $(q_1', \sigma_1')\mathcal{R}'(q_2', \sigma_2')$.

We have that $t_1 : q_1 \xrightarrow{l} q_1'$ (by Definition 25) and

$$\sigma_1 = \rho_1; \mathsf{s}_1[t_1], \qquad \lambda^{-1} = \rho_1; \mathsf{o}_1[t_1], \qquad \sigma_1 = \rho_1; \mathsf{d}_1[t_1]. \tag{8}$$

Since $\mathcal{R}$ is a pHDN-bisimulation, there is a transition $t_2 : q_2 \xrightarrow{l} q_2'$, some $\xi : \mathsf{T}_1[t_1]_{\text{new}} \rightharpoonup \mathsf{T}_2[t_2]_{\text{new}}$ and some $\zeta : \mathsf{T}_1[t_1] \rightharpoonup \mathsf{T}_2[t_2]$ s.t., for a $\delta' \subseteq \mathsf{d}_1[t_1]^{-1}; \zeta; \mathsf{d}_2[t_2]$, the following equalities hold

$$\zeta = (\mathsf{s}_1[t_1]; \delta; \mathsf{s}_2[t_2]^{-1}) \cup \xi, \qquad \mathsf{o}_1[t_1] = \zeta; \mathsf{o}_2[t_2], \qquad \langle q_1', \delta', q_2'\rangle \in \mathcal{R}. \tag{9}$$

We define $\rho_2 \stackrel{\text{def}}{=} (\sigma_2; \mathsf{s}_2[t_2]) \cup \rho_1; \xi$; it is trivial to note that $\rho_2 : \mathcal{N} \twoheadrightarrow \mathsf{T}_2[t_2]$. Hence, we have that $\zeta = \rho_1^{-1}; \rho_2$; and by (9),

$$\zeta = (\overbrace{\mathsf{s}_1[t_1]; \sigma_1^{-1}}^{\rho_1^{-1}}; \sigma_2; \mathsf{s}_2[t_2]^{-1}) \cup \xi = \rho_1; (\overbrace{(\sigma_2; \mathsf{s}_2[t_2] \cup \rho_1; \xi)}^{\rho_2}). \tag{10}$$

Moreover, by observing that $\sigma_2 \subseteq \rho_2; \mathsf{s}_2[t_2]$ (by definition of ρ_2), we conclude that $\sigma_2 = \rho_2; \mathsf{s}_2[t_2]$ since its inverse is a function.

By (9), $\mathsf{o}_1[t_1] = \zeta; \mathsf{o}_2[t_2]$, so by (10) we have $\rho_1; \mathsf{o}_1[t_1] = \lambda^{-1} = \rho_2; \mathsf{o}_2[t_2]$. This, and the definition of global transition, proves that $(t_2, \rho_2) : (q_2, \sigma_2) \xrightarrow{l,\lambda} (q_2', \sigma_2')$ is a global transition, where $\sigma_2' \stackrel{\text{def}}{=} \rho_2; \mathsf{d}_2[t_2]$. Finally,

$$\delta' = \mathsf{d}_1[t_1]^{-1}; \zeta; \mathsf{d}_2[t_2] = \mathsf{d}_1[t_1]^{-1}; \rho_1^{-1}; \rho_2; \mathsf{d}_2[t_2] = \sigma_1'^{-1}; \sigma_2'.$$

By definition of $\mathcal{R}'$, $\langle q_1', \delta', q_2' \rangle \in \mathcal{R}$ implies that $(q_1', \sigma_1')\mathcal{R}'(q_2', \sigma_2')$; this concludes the proof of the only "if part" (the proof of the symmetric clause of the bisimulation is similar).

Now we proof the "if" part of the theorem by showing that

$$\mathcal{R}' \stackrel{\text{def}}{=} \{\langle q_1, \delta, q_2 \rangle | (q_1, \sigma_1)\mathcal{R}(q_2, \sigma_2), \text{ where } \delta = \sigma_1^{-1}; \sigma_2\}$$

is a pHDN-bisimulation, if $\mathcal{R}$ is a global bisimulation for $\mathcal{A}_1$ and $\mathcal{A}_2$.

Suppose $\langle q_1, \delta, q_2 \rangle \in \mathcal{R}'$ and let $t_1 : q_1 \xrightarrow{\lambda} q_1'$ be a non redundant transition for $\mathcal{R}$. If we let $\rho_1 : \mathcal{N} \twoheadrightarrow \mathsf{T}_1[t_1]$ be an embedding s.t.

$$\rho_1|_{\mathsf{T}_1[t_1]_{\text{old}}} = \sigma_1; \mathsf{s}_1[t_1]^{-1}, \qquad \rho_1|_{\mathsf{T}_1[t_1]_{\text{new}} \setminus \text{dom}(\sigma_1 \cup \sigma_2)} : \mathcal{N} \twoheadrightarrow \mathsf{T}_1[t_1]_{\text{new}}$$

we have $\sigma_1 = \rho_1; \mathsf{s}_1[t_1]$ and $\rho_1^{-1}(\mathsf{T}_1[t_1]_{\text{new}}) \cap \sigma_2^{-1}(\mathsf{Q}_2[q_2]) = \emptyset$. By definition, there is a global transition $(t_1, \rho_1) : (q_1, \sigma_1) \xrightarrow{l,\lambda} (q_1', \sigma_1')$ s.t.

$$(q_1', \sigma_1')\mathcal{R}(q_2', \sigma_2'), \quad \sigma_2 = \rho_2; \mathsf{s}_2[t_2] \quad \rho_1; \mathsf{o}_1[t_1] = \lambda^{-1} = \rho_2; \mathsf{o}_2[t_2] \quad \sigma_2' = \rho_2; \mathsf{d}_2[t_2]. \tag{11}$$

Hence, there is a transition $t_2 : q_2 \xrightarrow{l} q_2'$ and, assuming $\zeta = \rho_1^{-1}; \rho_2 : \mathsf{T}_1[t_1] \hookrightarrow \mathsf{T}_2[t_2]$, we can define $\xi = \zeta|_{\mathsf{T}_1[t_1]_{\text{new}}}$. Therefore,

$$\begin{aligned}
\mathsf{o}_1[t_1] &= \rho_1^{-1}; \rho_2; \mathsf{o}_2[t_2] = \zeta; \mathsf{o}_2[t_2] && \text{(by (11))} \\
\zeta &= ((\rho_1|_{\mathsf{T}_1[t_1]_{\text{old}}})^{-1}; \rho_2|_{\mathsf{T}_2[t_2]_{\text{old}}}) \cup \xi \\
&= (\mathsf{s}_1[t_1]; \sigma_1^{-1}; \sigma_2; \mathsf{s}_2[t_2]^{-1}) \cup \xi \\
&= (\mathsf{s}_1[t_1]; \delta; \mathsf{s}_2[t_2]^{-1}) \cup \xi.
\end{aligned}$$

Finally, $(q_1', \sigma_1')\mathcal{R}(q_2', \sigma_2')$ implies that $\langle q_1, \delta', q_2 \rangle \in \mathcal{R}'$, provided that $\sigma_1'^{-1}; \sigma_2'$, thus $\delta' = \mathsf{d}_1[t_1]; \rho_1^{-1}; \rho_2; \mathsf{d}_2[t_2] = \mathsf{d}_1[t_1]; \zeta; \mathsf{d}_2[t_2]$. This concludes the proof, since the symmetric clause of bisimulation can be dealt with in a similar way. □