Context-free Synchronising Graphs

Pietro Cenciarelli and Alessandro Tiberi

University of Rome, "La Sapienza"
Department of Computer Science - Via Salaria 113, 00198 Roma.

Abstract. *Synchronising Graphs* is a system of parallel graph transformation designed for modeling process interaction in a network environment. Although notions of observational equivalence are abundant in the literature for process calculi, not so for graph rewriting, where system behaviour is typically context dependent. We propose a theory of *context-free* synchronising graphs and a novel notion of bisimulation equivalence which is shown to be a congruence with respect to graph composition and node restriction. This notion is used to provide a proof technique for the *hyperequivalence* of the Fusion calculus, through an encoding which is shown to be sound and complete. This builds a bridge between graph rewriting and process algebra. As a further application, we prove the correctness of a system component, called non-deterministic commuter, with respect to its specification. The result shows that our notion of equivalence is fine enough to discriminate between different degrees of parallelism in a network.

1 Introduction

Synchronising Graphs (SG) is a system of parallel graph transformation designed for modeling process interaction in a network environment. The system is inspired by [DM87], and it stems from the *Synchronized Hyperedge Replacement* of [FMT01], with which it has been compared in [CT05]. In the SG model, as in SHR, hyperedges represent agents, or software components, while nodes are thought of as communication channels, synchronisation points or, more generally, network communication infrastructure. The idea that hypergraphs may interact by synchronising action and co-action pairs at specific synchronisation points (the nodes) is quite intuitive, while the flexibility of the model in representing diverse network topologies and communication protocols makes SG a reasonable candidate as common semantic framework for interpreting different calculi. We followed this idea in [CTT05], where *Mobile Ambients* [CG00] and the *distributed* CCS of [RH01] (without restriction) were both modeled by using a common recursive synchronising graph architecture, called *ambient graphs*. In particular distributed CCS terms were shown to correspond to *flat* ambient graphs, while mobile ambients were trees.

Any common semantic framework for interpreting process calculi must come equipped with a notion of *observational equivalence*. Indeed such notions are often sought in the theory of concurrency so as to achieve a sort of compositionality in the semantics: considering as meaning of a process its abstract behaviour, which is traditionally identified by its bisimulation equivalence class, compositionality requires that, when equivalent processes are plugged into the same context, they produce equivalent results. This amounts to proving that bisimulation is a congruence. However, although such results are abundant in the literature for process calculi, not so for graph rewriting, where system behaviour is typically context dependent. Then, in order to bridge the gap between the two disciplines, we seek a characterisation of *context-free* synchronising graphs, and a suitable notion of bisimulation to capture their abstract behaviour. Such a notion has not yet emerged for related models such as [FMT01]. Indeed, some progress in this direction has been made in [KM01], which contains, to our knowledge, the only notion of observational equivalence proposed as yet in the framework of synchronised graph rewriting. However, the system of [KM01], we call it *SGR*, differs substantially from ours, both in the kind of synchronisation and in the model of mobility. In particular, in SGR graphs synchronise *à la Hoare* (an arbitrary number of them may synchronise with a unique action) while we use *Milner style* synchronisation, adopted by a large majority of process algebras. As for mobility, synchronising graphs, like for example processes in the Ambient calculus, can freely migrate within the network environment, while in SGR, where only *new* names can be communicated, processes cannot move to existing locations. Finally, the observational equivalence on SGR is, by the authors' admission, rather coarse. We meet their challenge for a finer notion, capable of distinguishing between different degrees of parallelism.

After presenting the general model of SG in section 2, we investigate the possible sources of context dependency in synchronised graph rewriting. In section 3 we focus on *context-free* SG, that is systems of synchronising graphs whose behaviour is specified by simple axioms called *productions*. The use of productions derives from [DM87] and it is built in the model of [FMT01]. A novel notion of bisimulation equivalence is introduced in section 4, and shown to be a congruence with respect to graph composition and node restriction. Two applications are considered for the proposed notion of equivalence. One is a (simple) proof of correctness of an implementation of a simple system called *nondeterministic commuter*. The second is the derivation of a proof technique for *hyperbisimulation* in the *Fusion calculus* [PV98].

Fusion is indeed a natural choice for linking the synchronising graphs with process calculi: because of its input/output symmetry it is closest to our model, while, because of its correspondence with π (which it contains as a proper subcalculus) it provides a gateway to the universe of process algebra. This is the topic (and the title) of section 5, where we prove an operational correspondence (preservation and reflection of computational steps) between Fusion and a specific theory of SG. Here we improve a previous result of [LM03], where operational correspondence with Fusion was obtained by an ad-hoc, simplified set of inference rules. Moreover, as mentioned above, we show that if the translations of two fusion processes P and Q are observationally equivalent graphs, then P and Q are *hyperequivalent*. We do not know whether the opposite direction (which would amount in *full abstraction* holds.

Notation. We often write function application without parentheses, that is fx instead of $f(x)$. We write $\boldsymbol{x}$ for a finite sequence $x_1, x_2, \dots x_n$. If $f \subseteq A \times B$ is a relation and $a \in A$, we write fa for the set $\{b \in B \mid (a, b) \in f\}$. The *domain* of f is the set $dom(f) = \{x \in A \mid \exists\, b \in B\,.\,(a, b) \in f\}$. If φ is an equivalence relation, we write $[x]_\varphi$ the equivalence class of an element x; or just $[x]$, when φ is understood.

Proofs of theorems and such are moved to the appendix.

2 Synchronising Graphs

Let $\mathcal{N}$ be a set of *nodes*, which we consider fixed throughout. A *graph* $G = (E, G, R)$ consists of a set E of *hyperedges* (or just *edges*), an attachment function $G : E \to \mathcal{N}^*$ and a set $R \subseteq |G|$ of nodes, called *restricted*, where

$$|G| = \{x \in \mathcal{N} \mid \exists\, e \in E \text{ s.t. } G\,e = x_1 \dots x_n \text{ and } x = x_i\}$$

is the set of nodes of the graph. When $G\,e = x_1 x_2 \dots x_n$ we call n the *arity* of e and say that the i-th *tentacle* of e is attached to x_i. We denote by $res(G)$ the set of restricted nodes of G, and by $fn(G)$ the set $|G| - res(G)$ of *free* nodes. We write $e(\boldsymbol{x})$ for an edge of a graph G such that $G\,e = \boldsymbol{x}$. Moreover, we let $\nu x\, G$ denote the graph $(E, G, R \cup \{x\})$ when $x \in |G|$, while $\nu x\, G = G$ otherwise. The *composite* of two graphs (E, G, R) and (D, F, S), written $G|F$, is the graph $(E \uplus D, G{+}F, R \uplus S)$, where $\uplus$ denotes disjoint union and $G{+}F$ is the attachment function mapping $e \in E$ to $G\,e$ and $d \in D$ to Fd.

Let $Act = \{a, b, \dots\} \cup \{\overline{a}, \overline{b}, \dots\}$ be a set of *actions*; we call $\overline{a}$ the *co-action* of a, and intend a by $\overline{\overline{a}}$. Let Act^+ denote the set $Act \times \mathcal{N}^*$. Given an element $(a, \boldsymbol{x})$ of Act^+, we call *objects* of a the components of $\boldsymbol{x}$. A *pre-transition* is a triple (G, Λ, H), written

$$G \xrightarrow{\Lambda} H$$

(or just Λ for short), where $\Lambda \subseteq \mathcal{N} \times Act^+$ is a relation, while G and H are graphs, called respectively the *source* and *destination* of Λ.

Notation. We write $(x, a, \boldsymbol{y})$ for an element $(x, (a, \boldsymbol{y}))$ of Λ, and (x, a) when $\boldsymbol{y}$ is the empty sequence. Given a pre-transition $G \xrightarrow{\Lambda} H$, we denote by $|\Lambda|$ the set $|G| \cup |H|$ and by $res(\Lambda)$ the set $res(G) \cup res(H)$. By $obj(\Lambda)$ we denote the set of objects $\{y \in \mathcal{N} : \exists\,(x, a, \boldsymbol{y}) \in \Lambda \text{ such that } y \in \boldsymbol{y}\}$. □

Intuitively, $(a, \boldsymbol{y}) \in \Lambda\, x$ expresses the occurrence of action a at node x. In SG the occurrence of both $(a, \boldsymbol{y})$ and $(\overline{a}, \boldsymbol{z})$ at x is called a *synchronisation*, and it corresponds to the *silent* action τ of most process calculi. Synchronising agents may exchange information. This is implemented in SG by unifying the lists $\boldsymbol{y}$ and $\boldsymbol{z}$ of objects, which are required to be of the same length. Only two agents at a time may synchronise at one node. Moreover, if an action occurs at a restricted node, then it *must* synchronise with a corresponding co-action, as we consider *observable* the unsynchronised actions. A restricted node may be "opened" by unifying it with an argument of an observable action, or with a node which is not restricted. These requirements are formalised as follows.

An *action set* is a relation $\Lambda \subseteq \mathcal{N} \times Act^{+}$ such that, for all nodes x, Λx has *at most* two elements and, when so, it is of the form $\{(a, \boldsymbol{y}), (\overline{a}, \boldsymbol{z})\}$, where the lengths of vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ coincide. Given an action set Λ, we denote by $\stackrel{\Lambda}{=}$ the smallest equivalence relation on nodes such that, if $(x, a, y_1 y_2 \dots y_n)$ and $(x, \overline{a}, z_1 z_2 \dots z_n)$ are in Λ, then $y_i \stackrel{\Lambda}{=} z_i$, for $i = 1 \dots n$. Arguments of unsynchronised actions are called *dangling*. More precisely, we call dangling in Λ the elements of the set

$$dng\,(\Lambda) = \{y \in obj\,(\Lambda) \,:\, \Lambda\, x = \{(a, y_1 \dots y_n)\} \text{ and } y \stackrel{\Lambda}{=} y_i, \text{ for some } x\}.$$

Definition 1 *A* transition *is a pre-transition $G \xrightarrow{\Lambda} H$ such that:*

1. *Λ is an action set such that $\mathrm{dom}\,(\Lambda) \cup obj\,(\Lambda) \subseteq |G|$;*
2. *if a node x is restricted in G then $|\Lambda\, x| \neq 1$;*
3. *if $x \in |H|$, then $x \in fn\,(H)$ if and only if $x \in fn\,(G) \cup dng\,(\Lambda)$.*
4. *$H = \rho H$ for some unifier ρ of Λ such that $\rho\, x \in fn\,(G)$ for all $x \in fn\,(G)$.*

By condition 1, the pre-transition $e(x) \xrightarrow{x,a,y} d(y)$ is not a transition, because $y \notin |e(x)|$. A consequence of 1 and 3 is that all free nodes in the destination of a transition must occur in the source. Hence, while $e(x) \xrightarrow{\emptyset} \nu y\, d(y)$ is a legal transition, $e(x) \xrightarrow{\emptyset} d(y)$ is not. Condition 4 enforces fusions. It also grants a privilege to the free nodes when they are fused with the bound, which allows $\nu y\, e(x\, y) \xrightarrow[x,a,y]{x,\overline{a},x} d(x)$ and forbids $\nu y\, e(x\, y) \xrightarrow[x,a,y]{x,\overline{a},x} d(y)$. This restriction is not essential in fact for the theory of synchronising graphs but, as we shall see, it simplifies the meta-theory without loss of generality.

In SG, synchronisation is subject to a non-interference condition: two transitions can be synchronised provided they are disjoint and they share no restricted nodes. Formally, $G \xrightarrow{\Lambda} H$ and $F \xrightarrow{\Theta} K$ are said to be *non-interfering*, written $\Lambda \# \Theta$, when:

- $\Lambda \cap \Theta = \emptyset$, and moreover
- $res\,(\Lambda) \cap |\Theta| = res\,(\Theta) \cap |\Lambda| = \emptyset$.

It is an easy check that the only nodes two non-interfering transitions may have in common are the free nodes in their sources. By $F \# G$ we mean that the identity transitions $F \xrightarrow{\emptyset} F$ and $G \xrightarrow{\emptyset} G$ do not interfere.

As mentioned above, graph synchronisation involves unification, like in the Fusion calculus. Let $f : \mathcal{N} \to \mathcal{N}$ be a function on nodes and let (E, G, R) be a graph; we write fG the graph (E, fG, fR) obtained by substituting all nodes x in G with fx. More precisely, for all $e \in E$, if $G\,e = x_1 \ldots x_n$ then $(fG)\,e = fx_1 \ldots fx_n$. A function $f : \mathcal{N} \to \mathcal{N}$ is said to *agree* with an equivalence relation φ on $\mathcal{N}$ if, as a set of pairs, it is a subset of φ, that is if $(x, fx) \in \varphi$, for all nodes $x \in \mathcal{N}$. A *unifier* of φ is a function ρ which agrees with φ and such that $|\,\rho[x]_\varphi| = 1$ for all x. By a slight abuse, we say that a function agrees with (or unifies) an action set Λ to mean that it agrees with (unifies) the relation $\stackrel{\Lambda}{=}$.

The rules of the system of synchronising graphs are:

$$[\,\text{sync}\,] \quad \frac{G \xrightarrow{\Lambda} H \qquad F \xrightarrow{\Theta} K}{G|F \xrightarrow{\Lambda \cup \Theta} \rho(H|K)} \quad (\text{if } \Lambda \# \Theta \text{ and } \rho \text{ unifies } \Lambda \cup \Theta)$$

$$[\,\text{open}\,] \quad \frac{G \xrightarrow{\Lambda} H}{\nu x\, G \xrightarrow{\Lambda} H} \quad (\text{if } x \in dng\,(\Lambda))$$

$$[\,\text{res}\,] \quad \frac{G \xrightarrow{\Lambda} H}{\nu x\, G \xrightarrow{\Lambda} \nu x\, H} \quad (\text{if } x \notin dng\,(\Lambda))$$

A non-deterministic commuter (extended example). Consider a system consisting of a certain number of input and output *sockets*. The system, which we shall call *non-deterministic commuter*, acts by connecting client processes possibly attached to an input socket, non-deterministically with one of the output sockets, where server processes may be attached. Connections are established one at a time. Figure 1 depicts a commuter **C** with three input and two output sockets. A client process **r** is being connected with a server **q**.

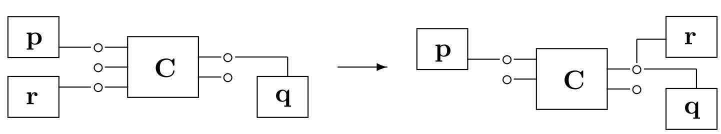


Fig. 1. A non-deterministic commuter

Non-deterministic commuters can be engineered in SG by assembling simple components of the form $in\,(x\,u)$ and $out\,(u\,y)$, representing input and output sockets respectively. Clients are meant to be attached to the node x, while servers are attached to y. The node u represents an internal communication channel of the system. Then, ignoring the two unused sockets, the initial state of the commuter **C** is represented by the graph $\nu u\; in\,(x\,u) \mid in\,(z\,u) \mid out\,(u\,y)$. The system's behaviour is specified by the following two transitions, where a and $\overline{a}$ represent the input and output actions respectively.

$$
\begin{aligned}
in\,(x\,u) &\xrightarrow{u,a,x} \emptyset \\
out\,(u\,y) &\xrightarrow{u,\overline{a},y} out\,(u\,y)
\end{aligned}
$$

To be precise, these are to be considered as axiom *schemes* for deriving the whole system's transition. In particular, in lack of labels on edges, as there are for example in [FMT01], we must assume one axiom of the first kind for each input socket, and one of the second for each output. In the present example, we further assume that any edge can perform a passive (empty) transition to itself. Then, ignoring $\mathbf{p}$ and its socket, the transition of figure 1 is obtained by the derivation of figure 2. □

$$
\dfrac{in\,(z\,u) \xrightarrow{u,a,z} \emptyset \quad out\,(u\,y) \xrightarrow{u,\overline{a},y} out\,(u\,y)}{in\,(z\,u) \mid out\,(u\,y) \xrightarrow{\substack{u,\overline{a},y\\u,a,z}} out\,(u\,y)}\ [\,\mathsf{sync}\,]
$$

$$
\dfrac{in\,(z\,u) \mid out\,(u\,y) \xrightarrow{\substack{u,\overline{a},y\\u,a,z}} out\,(u\,y)}{\nu u\; in\,(z\,u) \mid out\,(u\,y) \xrightarrow{\substack{u,\overline{a},y\\u,a,z}} \nu u\; out\,(u\,y)}\ [\,\mathsf{res}\,]
$$

$$
\dfrac{r(z) \mid q(y) \xrightarrow{\emptyset} r(z) \mid q(y) \qquad \nu u\; in\,(z\,u) \mid out\,(u\,y) \xrightarrow{\substack{u,\overline{a},y\\u,a,z}} \nu u\; out\,(u\,y)}{r(z) \mid q(y) \mid \nu u\; in\,(z\,u) \mid out\,(u\,y) \xrightarrow{\substack{u,\overline{a},y\\u,a,z}} r(y) \mid q(y) \mid \nu u\; out\,(u\,y)}\ [\,\mathsf{sync}\,]
$$

Fig. 2. Deriving a commuter transition

The rest of this section is devoted to proving two meta-theoretical properties of the system of synchronising graphs which are crucial for the theory of observational equivalence developed in the next section.

Notation. Let $G \xrightarrow{\Lambda} H$ and $F \xrightarrow{\Theta} K$ be transitions; we denote by $\Lambda * \Theta$ the set of transitions of the form $G|F \xrightarrow{\Lambda\cup\Theta} \rho(H|K)$ obtained by synchronising Λ and Θ. Clearly, $\Lambda * \Theta$ is empty when Λ and Θ interfere. The expression $(\Lambda * \Theta) * \Phi$ stands for $\bigcup_{\Xi\in\Lambda*\Theta}(\Xi * \Phi)$. Similarly, we let $\nu x\,\Lambda$ be the transition which results from restricting Λ on x by an application of $[\,\mathrm{res}\,]$ or $[\,\mathrm{open}\,]$. The expression $\nu x\,(\Lambda * \Theta)$ denotes the set of transitions of the form $\nu x\,\Xi$ with $\Xi \in \Lambda * \Theta$.

Lemma 1 *Let Λ and Θ be transitions and let x occur unrestricted in the source of Λ. Then, $(\nu x\,\Lambda) * \Theta \subseteq \nu x\,(\Lambda * \Theta)$.*

Note that the opposite inclusion does not hold in general. For example, if Λ is $e(x\,y) \xrightarrow{x,a,y} h(y)$ and Θ is $\nu z\,d(x\,z) \xrightarrow{x,\overline{a},z} k(z)$, then $\nu y\,(\Lambda * \Theta)$ includes

$\nu yz\, e(x\,y)|d(x\,z) \xrightarrow[x,a,y]{x,\overline{a},z} \nu y\, h(y)|k(y)$, while $(\nu y\, \Lambda) * \Theta$ is empty because the result of applying [sync] to $\nu y\, \Lambda$ and Θ violates condition 3 of definition 1.

Lemma 2 *Synchronisation is associative:* $(\Lambda * \Theta) * \Xi = \Lambda * (\Theta * \Xi)$.

3 Context-free theories

Notation. In this section we abandon the brute force notion of node substitution in a graph G adopted in the previous sections, and denote by hG the graph obtained by applying a substitution h to the *free* nodes of G, while bound nodes are suitably renamed so as to avoid capture. This simplifies the statement of results such as lemma 4 while remaining consistent with the theory developed so far. In particular, note that the new interpretation of $\rho(H|K)$ in [sync] does not alter the set of derivable transitions. □

A *theory* of synchronising graphs is a set of transitions which is closed under the inference rules. The theory *generated* by a set $\mathcal{T}$ of transitions, called *axioms*, is the smallest theory including $\mathcal{T}$. One of the aims of the present paper is to characterise the theories of synchronising graphs in which the behaviour of a graph is not affected by the context. The following examples will clarify this concept.

Example. In the theory generated by a unique axiom $e|d \xrightarrow{\emptyset} \emptyset$, the two processes e and d, considered in isolation, have the same behaviour: none of them can move. However, if set in the context $[\_]|d$, the two processes exhibit quite different behaviour, as $e|d$ can move while $d|d$ cannot.

Example ($\star$). In the theory generated by a unique axiom $\nu x\, e(x) \xrightarrow{\emptyset} \emptyset$, the process $e(x)$ cannot move, thus exhibiting the same catatonic behaviour as the empty process $\emptyset$. However, when set in a context $\nu x\, [\_]$ where x is restricted, $\nu x\, e(x)$ can move while $\nu x\, \emptyset = \emptyset$ cannot.

Example. In the theory generated by the four axioms $h(x\,y) \xrightarrow[x,a,y]{x,\overline{a},x} \emptyset$, $d \xrightarrow{\emptyset} d$, $e(x\,y) \xrightarrow{\emptyset} e(x\,y)$ and $e(x\,x) \xrightarrow{x,a} \emptyset$, $e(x\,y)$ behaves just like the process d, cycling forever over itself. However, when put in parallel with $h(x\,y)$, $e(x\,y)$ yields a trace which $h(x\,y)\,|\,d$ does not have: $h(x\,y)|e(x\,y) \xrightarrow[x,a,y]{x,\overline{a},x} e(x\,x) \xrightarrow{x,a} \emptyset$.

Example. In the theory generated by the three axioms $h(x\,y) \xrightarrow[x,a,y]{x,\overline{a},x} \emptyset$, $d \xrightarrow{\emptyset} d$ and $e(x) \xrightarrow{\emptyset} e(x)$, the processes $e(x)$ and d have the same behaviour. However, when put in parallel with $h(x\,y)$, $e(x)$ yields a transition to a catatonic state, namely $e(y)$, which $h(x\,y)\,|\,d$ cannot reach. □

The four examples above capture in fact the only possible sources of context dependency in a theory of synchronising graphs. This is shown in the present section by providing a notion of bisimulation equivalence on graphs and then proving that, in any theory generated from axiom sets which exclude the four scenarios just described, the proposed equivalence is a congruence with respect to restriction and parallel composition.

An *instance* of a transition $G \xrightarrow{\Lambda} H$ is a transition of the form $hG \xrightarrow{h\Lambda} \rho\, hH$ where $h : \mathcal{N} \to \mathcal{N}$ is a node substitution and ρ is a unifier of $h\Lambda$. A *production* is a transition whose source consists of a single edge $e(\boldsymbol{x})$, where all components of $\boldsymbol{x}$ are distinct and none of them is restricted. A theory of synchronising graphs is called *context-free* when it is generated by all the instances of a given set of productions. Note that, the constraints productions are asked to satisfy prevent the first three examples of context dependency to occur, while the use of *all* their instances for generating the theory accounts for the fourth example.

Parallel and sequential composition have useful meta-theoretical properties in context-free theories. Let Λ be any transition of a composite graph $G|F$; there exists a Y-shaped derivation of Λ where the actions of G and those of F are synchronised separately in each branch of the Y. More precisely:

Theorem 1 *Let Λ be a transition in a context-free theory, and let $G|F$ be its source. Then, Λ is an element of a set $\nu\boldsymbol{x}\,(\Theta * \Xi)$, where G has exactly the same edges as the source of Θ and F as the source of Ξ.*

Similarly, synchronisations occurring in parallel can be serialised, provided the axioms of the theory are suitably simple. We call *simple* a transition Λ such that $|dom\,(\Lambda)| \leq 1$. A simple transition of the form $G \xrightarrow{\emptyset} G$ is called an *identity.*

Theorem 2 *Let $G \xrightarrow{\Lambda \cup \Theta} H$ be a transition in a context-free theory generated from simple axioms including the identities, and let $\mathrm{dom}\,(\Lambda) \cap \mathrm{dom}\,(\Theta)$ be empty. Then $G \xrightarrow{\Lambda \cup \Theta} H$ factorises as $G \xrightarrow{\Lambda} F \xrightarrow{\rho\,\Theta} H$, where ρ is a unifier of Λ.*

Corollary 1 *Any transition in a context-free theory with simple axioms and identities factorises as a sequence of simple transitions.*

4 Observational equivalence

We call *parameters* the elements of the set $\mathcal{P} = \mathcal{N} \times \mathit{Act} \times \mathit{Nat}$. Intuitively, a parameter (x, a, i) is an abstraction over the i-th argument y_i of an action $(x, a, \boldsymbol{y}\,)$. We call *observations* the elements of the set $\mathcal{O} = \mathcal{N} \cup \mathcal{P}$. Given an action set Λ, the relation $\stackrel{\Lambda}{=}$ extends to a relation $\stackrel{\Lambda}{=}_o$ on observations as follows:

$$\stackrel{\Lambda}{=}_o \subseteq \mathcal{O} \times \mathcal{O}$$

is the smallest equivalence relation such that $(x, a, i) \stackrel{\Lambda}{=}_o (x, \overline{a}, i)$ and moreover $(y, b, j) \stackrel{\Lambda}{=}_o z$ if $(y, b, z_1 \dots z_j \dots z_n) \in \Lambda$ and $z = z_j$.

Not all of $\stackrel{\Lambda}{=}_o$ is to be observed. The set $obs\,(\Lambda)$ of *observables* of a transition $G \xrightarrow{\Lambda} H$ consists of its observable objects, the set of which we denote by $|\Lambda|_o$, together with the parameters of the unsynchronised actions:

$$|\Lambda|_o = \{x \in fn\,(G) \,:\, x \text{ is dangling or } x \stackrel{\Lambda}{=} y \neq x \text{ for some } y \in fn\,(G)\};$$
$$obs\,(\Lambda) = |\Lambda|_o \cup \{(x, a, i) \in \mathcal{P} \,:\, \Lambda\, x = \{(a, \boldsymbol{y})\} \text{ and } 0 \leq i \leq |\boldsymbol{y}|\}.$$

Note that x is not observable in $e(x) \xrightarrow[x,a,x]{x,\overline{a},x} e(x)$ because, although it is free in $e(x)$, "self-fusion" has no bearing on the interacting environment.

The observable part *of the relation* $\stackrel{\Lambda}{=}_o$, written $\stackrel{\Lambda}{\simeq}$, is the equivalence relation obtained by restricting $\stackrel{\Lambda}{=}_o$ to $obs\,(\Lambda)$:

$$p \stackrel{\Lambda}{\simeq} q \text{ if and only if } p \stackrel{\Lambda}{=}_o q \text{ and } p, q \in obs\,(\Lambda).$$

Definition 2 *Two transitions* $G \xrightarrow{\Lambda} H$ *and* $F \xrightarrow{\Theta} K$ *are called* equivalent *when:*

- $\stackrel{\Lambda}{\simeq}$ *and* $\stackrel{\Theta}{\simeq}$ *are the same relation, and*
- $\{x \in fn\,(G) \,:\, |\Lambda\, x| = 2\} = \{y \in fn\,(F) \,:\, |\Theta\, y| = 2\}$.

Alpha equivalent graphs, that is graphs which are identical up to renaming of bound nodes, do have equivalent transitions. In particular, $\nu x\, e(x\, y) \xrightarrow{y,a,x} d(x)$ and $\nu z\, e(z\, y) \xrightarrow{y,a,z} d(z)$ are equivalent. However, if we want the two sources to be observationally equivalent, we must relate x and z in the target graphs.

Let Λ and Θ be equivalent transitions as above; we say that $x \in |\Lambda|$ and $y \in |\Theta|$ are *observationally related* if either $x = y \in fn\,(G) \cap fn\,(F)$ or there exists $o \in obs\,(\Lambda)$ such that $x \stackrel{\Lambda}{=}_o o \stackrel{\Theta}{=}_o y$. Note that the relation is symmetrical because, by the first clause in definition 2, $obs\,(\Lambda) = obs\,(\Theta)$ when Λ and Θ are equivalent. Note also that, as shown by the above transitions of alpha-equivalent graphs, observational relation is not restricted to observable nodes, being it defined by $\stackrel{\Lambda}{=}_o$ and $\stackrel{\Theta}{=}_o$ rather that by $\stackrel{\Lambda}{\simeq}$ and $\stackrel{\Theta}{\simeq}$.

Definition 3 *A* simulation *is a binary relation* $\mathcal{S}$ *on graphs such that* $G\, \mathcal{S}\, F$ *implies that for all transitions* $G \xrightarrow{\Lambda} H$ *there exists a transition* $F \xrightarrow{\Theta} K$ *such that:*

- Λ *and* Θ *are equivalent, and*
- $hH\, \mathcal{S}\, kK$*, where* $h : |H| \to \mathcal{N}$ *and* $k : |K| \to \mathcal{N}$ *are injective node substitutions such that* $h\, x = k\, y$ *if and only if* x *and* y *are observationally related.*

A graph G is *simulated* by a graph F, written $G \prec F$, if there exists a simulation $\mathcal{S}$ such that $G\, \mathcal{S}\, F$. Note that $\prec$ is a simulation itself.

Example. $\nu zw\, e(x\,y\,z\,w) \xrightarrow[y,a,w]{x,b,z} f(z)$ and $\nu zw\, d(x\,y\,z\,w) \xrightarrow[y,a,z]{x,b,w} f(z)$ are equivalent transitions. They are, however, not part of a bisimulation, unless $f(z)$ and $f(w)$ simulate each other. And this may only happen when f is a process that cannot act.

Lemma 3 *Let $h : \mathcal{N} \to \mathcal{N}$ be a node substitution and let $hG \xrightarrow{\Psi} H$ be derivable from a set of productions. Then, Ψ is of the form $hG \xrightarrow{h\Lambda} \nu\mathbf{x}\,\rho(hK)$, where $G \xrightarrow{\Lambda} K$ is a derivable transition and ρ is a unifier of $h\Lambda$.*

Lemma 4 *Simulation is preserved by node substitution: for all $h : \mathcal{N} \to \mathcal{N}$, $G \prec F$ implies $hG \prec hF$.*

Indeed the lemma may fail for non-context-free theories, as for instance that of example $(\star)$. Note also that the lemma would have to be rephrased were we adopting the brute-force notion of node substitution of section 2.

Corollary 2 *Simulation is transitive.*

Next we show that simulation is preserved by node restriction and parallel composition. The result is proven by a standard technique which consists in showing that a suitably defined relation $\mathcal{R}$ is a simulation. In particular, writing $\stackrel{\alpha}{\equiv}$ for alpha-equivalence, let

$$\mathcal{S} = \{(\nu\boldsymbol{x}\,(G|U), \nu\boldsymbol{x}\,(F|U)) \,:\, G \,\#\, U,\ F \,\#\, U \text{ and } G \prec F\} \text{ and}$$
$$G\,\mathcal{R}\,F \text{ if and only if } G \stackrel{\alpha}{\equiv} G' \text{ and } G'\mathcal{S}F' \text{ and } F' \stackrel{\alpha}{\equiv} F.$$

Theorem 3 *$\mathcal{R}$ is a simulation.*

A *bisimulation* is a symmetric simulation. Two graphs G and F are called *bisimulation equivalent*, written $G \sim F$, when they are related by a bisimulation. This is our notion of observational equivalence on context-free synchronising graphs.

Theorem 4 *Bisimulation is a congruence.*

The result is obtained as for theorem 3: by replacing $\prec$ with $\sim$ in the definition of $\mathcal{S}$, we get a relation $\tilde{\mathcal{R}}$ similar to $\mathcal{R}$. The proof that $\tilde{\mathcal{R}}$ is a bisimulation goes through as for $\mathcal{R}$. Then, for a non-interfering U, $G \sim F$ is shown to imply $G|U \sim F|U$ by choosing the empty vector for $\boldsymbol{x}$ in the definition of $\tilde{\mathcal{R}}$. And similarly for restriction.

A non-deterministic commuter (continued). The internal communication channel of a non-deterministic commuter as described in section 2 can be implemented by a local network without affecting the observable behaviour of the system. We build such internal infrastructure by means of simple components, called *connectors*, of the form $c(u_1u_2v)$. Connectors echo the information received from u_1 (call it the *input* node) over u_2 (the *output* node) using a *service* node v for the matching. Once v has served its purpose, a new service node is created. In symbols:

$$c(u_1u_2v) \xrightarrow{\substack{u_1,\overline{a},v\\ u_2,a,v}} \nu w\, c(u_1u_2w).$$

Figure 3 shows a new representation of the commuter of figure 1 where the internal channel is implemented by a net of four connectors. (We draw labeled boxes for edges and bullets for nodes. The latter are solid when restricted, and clear otherwise. Tentacles are represented by lines connecting edges with nodes.) With this implementation, the transition of figure 2 is simulated by an equivalent transition $G \xrightarrow{\Lambda} H$, where (grouping all indexed names into vectors):

$$\begin{aligned}
G &= \ r(z) \,|\, q(y) \,|\, \nu\, \boldsymbol{u}\, \boldsymbol{v}\; in\,(z\,u_1) \,|\, c(u_1u_2v_1) \,|\, c(u_2u_3v_2) \,|\, c(u_3u_4v_3) \,|\, out\,(u_4y)\\
H &= \ r(y) \,|\, q(y) \,|\, \nu\, \boldsymbol{u}\, \boldsymbol{w}\; c(u_1u_2w_1) \,|\, c(u_2u_3w_2) \,|\, c(u_3u_4w_3) \,|\, out\,(u_4y)\\
\Lambda &= \{\, (u_1,a,z), (u_1,\overline{a},v_1), (u_2,a,v_1), (u_2,\overline{a},v_2),\\
&\qquad (u_3,a,v_2), (u_3,\overline{a},v_3), (u_4,a,v_3), (u_4,\overline{a},y)\,\}.
\end{aligned}$$

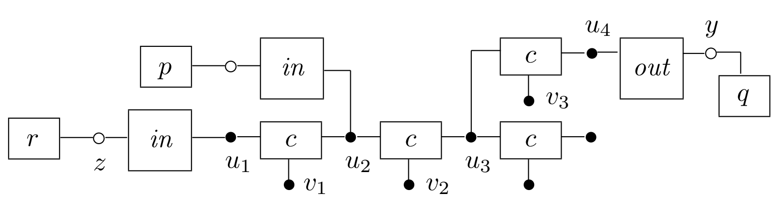


Fig. 3. a non-deterministic commuter with internal structure

In general, a graph made of sockets and connectors behaves like a non-deterministic commuter when it is a tree (that is, connected and acyclic) in which output sockets are attached by their first tentacle, input sockets by their second, no connector is attached by its service node, and moreover there exists a node, called *pivot*, splitting the graph into two (possibly disconnected) subgraphs, one including all the input and the other all the output sockets. In figure 3, nodes u_2, u_3 and u_4 are all pivotal. Of course, in the absence of a pivot, the internal infrastructure may allow for parallel connections, which are not contemplated in the specification of section 2.

Proposition 1 *Any graph G satisfying the conditions above is bisimulation equivalent to the non-deterministic commuter obtained by deleting all the connectors from G and attaching all sockets to the pivot node.*

5 A gateway to process algebra

Our rule of synchronisation is reminiscent of the communication law of the *Fusion Calculus* [PV98] (see appendix). Fusion contains the π-calculus as a proper sub-calculus and the encoding of the latter in the former is straightforward. Linking to Fusion is therefore a natural gateway for us to the universe of process algebra. Here we exhibit a context-free theory of synchronising graphs which can be viewed, in a precise sense, as a parallel, syntax-free version of the Fusion calculus. Beside establishing an operational correspondence with the calculus, and proving it sound and complete, we show that the bisimulation of section 4 can be conveniently used for proving observational equivalence (*hyperequivalence*) of Fusion processes.

Of the Fusion calculus we shall not consider summation, which can be treated straightforwardly when restricted to *guarded* sums, while it complicates translation in its general form. Similarly, we do not consider fusion prefixes, such as $\{\boldsymbol{x} = \boldsymbol{y}\}.P$, which can be regarded as derived forms (corresponding semantically to $(z)\,(\overline{z}\,\boldsymbol{x} \mid z\,\boldsymbol{y}.P)$). Without loss of generality, we also make the simplifying assumptions that the bound names of a sub-term are always new.

As in SG, input and output are completely symmetrical in Fusion. The name vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ that are passed as arguments of symmetrical communication actions $x\boldsymbol{y}$ and $\overline{x}\boldsymbol{z}$ are matched to define an equivalence relation, denoted by $\{\boldsymbol{y} = \boldsymbol{z}\}$. More precisely $\{\boldsymbol{y} = \boldsymbol{z}\}$ is the smallest equivalence relation on nodes including all pairs (y_i, z_i), with $1 \leq i \leq |\boldsymbol{y}| = |\boldsymbol{z}|$. This relation is used as label to the transition corresponding to communication. Members of the same equivalence class are fused by applying a *scope* operator, written $(\boldsymbol{x})P$ which corresponds to restriction in SG. This is shown by the rules [COM] and [SCOPE]. We denote by $bn(P)$ the bound names of P. Note that *scope* is the only binding operator in fusion, whereas in the π-calculus there are two (input and restriction).

Our encoding makes use of syntactic entities called *templates*. A template θ is a term of Fusion in which all *free* names are replaced by *distinct* progressive indices, starting from 1, as in $1\,2.3 \mid (w)\,\overline{4}\,w.5$. The *arity* of θ, written $|\theta|$, is its highest index: 5 in the above example, 0 in the *inaction* process $\emptyset$. Given a vector $\boldsymbol{x} = x_1 x_2 \dots x_n$ of names, with $n = |\theta|$, we write $\theta\langle \boldsymbol{x} \rangle$ the term of Fusion obtained by replacing each index i with x_i in θ. For example, if θ is the above template, $\theta\langle x\,y\,z\,x\,z \rangle$ is the term $x\,y.z \mid (w)\,\overline{x}\,w.z$.

The *theory of Fusion*, SG_F, is a context-free theory of synchronising graphs featuring the names of Fusion as nodes and indexed (prefix) templates as edges. An edge $\theta_n(\boldsymbol{x}\,)$ is to represent an instance of the term $\theta\langle \boldsymbol{x}\,\rangle$. Let the metavariable a range over the set $\{x, \overline{x} \dots\}$ of names and co-names. Processes translate into graphs by a function $[\![\_]\!]$ defined as follows:

$$[\![\emptyset]\!] = \emptyset$$

$$[\![a\,\boldsymbol{y}.P]\!] = \theta_n(\boldsymbol{x}), \text{ where } n \text{ is } \textit{new} \text{ and } \theta\langle \boldsymbol{x} \rangle = a\,\boldsymbol{y}.P$$

$$[\![P \mid Q]\!] = [\![P]\!] \mid [\![Q]\!]$$

$$[\![(x)\,P]\!] = \nu x\,[\![P]\!]$$

Note that indices are used to distinguish different occurrences of the same term, as in $[\![x.y \,|\, x.y]\!] = \theta_7(x,y) \,|\, \theta_4(x,y)$, where θ is the template 1.2. This explains the requirement for a new n in the second clause.

We model Fusion by letting $Act = \{\imath, \bar{\imath}\}$. Then, elements of an action set are either of the form $(x, \imath, \boldsymbol{y})$, which we write $x\,\boldsymbol{y}$, or of the form $(x, \bar{\imath}, \boldsymbol{y})$, which we write $\overline{x}\,\boldsymbol{y}$. The axioms of SG_F are generated by all productions of the form:

$$[\![a\,\boldsymbol{y}.P]\!] \xrightarrow{a\,\boldsymbol{y}} [\![P]\!].$$

Transitions are labeled in the Fusion Calculus by actions γ of two kinds: possibly *bound* input/outputs, as in $(\boldsymbol{x})\,a\,\boldsymbol{y}$, and *fusion* actions, represented by equivalence relations on names (see the appendix). If φ is such a relation and X is a set of names, we let $\{(y,z) \in \varphi \,|\, y \notin X \text{ and } z \notin X\} \cup \{(x,x) \,|\, x \in X\}$ be denoted by $\varphi \restriction X$.

Lemma 5 *Let $[\![P]\!] \xrightarrow{\Lambda} G$, be a simple, nonempty transition derivable in SG_F.*

- *If $\Lambda = \{a\,\boldsymbol{y}\}$ then $P \xrightarrow{(\boldsymbol{x})\,a\,\boldsymbol{y}} Q$, where $[\![Q]\!] = G$ and $\boldsymbol{x}$ includes all names in $\boldsymbol{y}$ that are bound in P;*
- *if Λ is a synchronisation then $P \xrightarrow{\varphi} Q$, where φ is $\stackrel{\Lambda}{=}\restriction bn(P)$ and $\sigma[\![Q]\!] = G$ for a unifier σ of φ*

Note that, when transitions correspond to fusions, synchronising graphs do apply the fusion to the right hand side of the transition, while Fusion only does it to bound names. This justifies the application of σ to $[\![Q]\!]$ in the second clause of the theorem. In general, the notion of simulation given in [PV98] justifies defining a *computation* $P \stackrel{\boldsymbol{\gamma}}{\Longrightarrow} Q$ in Fusion to be a sequence $\boldsymbol{\gamma} = \gamma_1 \ldots \gamma_n$ of transitions $P \xrightarrow{\gamma_1} Q_1$, $\sigma_1 Q_1 \xrightarrow{\gamma_2} Q_2$, $\sigma_2 Q_2 \xrightarrow{\gamma_2} Q_3$,... where $Q_n = Q$ and, for all i, σ_i is a unifier of γ_i. The above lemma, in conjunction with corollary 1, yields the following soundness result.

Theorem 5 *Let $[\![P]\!] \xrightarrow{\Lambda} G$ be in SG_F. There exists a computation $P \stackrel{\boldsymbol{\gamma}}{\Longrightarrow} Q$ such that $G = \sigma[\![Q]\!]$, where σ unifies the last fusion in $\boldsymbol{\gamma}$, Λ includes all fusions in $\boldsymbol{\gamma}$, and $\Lambda\,x = \{(\imath, \boldsymbol{y})\}$ if and only if $(\boldsymbol{z})\,x\,\boldsymbol{y} \in \boldsymbol{\gamma}$ (and similarly for $\bar{\imath}$).*

The following completeness result shows that translation preserves computational steps.

Theorem 6 *Let $P \xrightarrow{\gamma} Q$ be a transition in the Fusion Calculus.*

- *If $\gamma = (\boldsymbol{x})\,a\,\boldsymbol{y}$, then $[\![P]\!] \xrightarrow{a\,\boldsymbol{y}} [\![Q]\!]$ is derivable in SG_F;*
- *if γ is a fusion, then $[\![P]\!] \xrightarrow{\Lambda} \rho[\![Q]\!]$ is derivable in SG_F, where Λ is simple, ρ unifies γ, and γ is $\stackrel{\Lambda}{=}\restriction bn(P)$.*

Note that α-equivalent terms are identified in Fusion, and hence judgments of the form $[\![P]\!] \xrightarrow{\Lambda} [\![Q]\!]$ hold in theorem 6 *up to α-equivalence*. This means that there exist P' and Q' that are α-equivalent respectively to P and Q and such that $[\![P']\!] \xrightarrow{\Lambda} [\![Q']\!]$ is derivable.

As in the π-calculus, the naive definition of bisimulation relating processes in Fusion [PV98, Def.7] does not yield observational equivalence: bisimilar processes may be distinguished by plugging them into context which fuse some of their names. Then, *hyperbisimulation* is introduced in [PV98, Def.8], that is a simulation closed under name substitution. Hyperbisimulation equivalence, which we write $\stackrel{h}{\sim}$, is shown to be preserved by the operations of the calculus, but it may be hard to prove, as it involves quantification over all possible substitutions. In the rest of this section we introduce a proof technique for hyperbisimulation which consists in applying our translation to the Fusion processes and then checking that the resulting graphs are bisimulation equivalent in a system where parallelism has been suitably restricted. No quantification over substitutions involved.

A *single action transition* is a transition of the form $\{(x, a, \boldsymbol{y})\}$ or of the form $\{(x, a, y_1 \dots y_n), (w, \bar{a}, z_1 \dots z_n)\}$, with x possibly different from w. A *hypobisimulation*, is a relation on synchronising graphs defined just like bisimulation in section 4, but restricting quantification over Λ and Θ in definition 3 to single action transitions. Two graphs G and F are hypobisimulation equivalent, or *hypoequivalent* for short, written $G \dot{\sim} F$, when they are related by a hypobisimulation. Hypoequivalence satisfies the following properties: it is preserved by node substitution, and moreover it is a congruence. We do not include the proofs of these results as they are replicas of the ones in section 4.

In general, neither does hypoequivalence imply bisimulation equivalence nor the other way around. Moreover, two graphs of the form $[\![P]\!]$ and $[\![Q]\!]$ may be bisimulation equivalent even if P and Q are not hyperequivalent. Below we show that the same cannot happen if $[\![P]\!]$ and $[\![Q]\!]$ are hypoequivalent.

Theorem 7 *Let P and Q be processes in Fusion. $[\![P]\!] \dot{\sim} [\![Q]\!]$ implies $P \stackrel{h}{\sim} Q$.*

6 Conclusions

Synchronised graph rewriting has been proposed as a unifying semantic framework for process calculi [HM01,FMT01,LM03,CTT05]. To fulfill this project, graphs must be endowed with abstract notion of behaviour. In this paper we do so by introducing a notion of bisimulation equivalence for a system of context-free synchronising graphs, and by proving it a congruence with respect to parallel composition and node restriction. As an application, the Fusion calculus is encoded in SG and an operational correspondence is proven between the terms of the calculus and their translations. Our bisimulation yields a proof technique for hyperbisimulation, which captures observational equivalence in Fusion. Hyperbisimulation is defined by a quantification over all substitution. By our technique, we can avoid this difficulty by just observing suitable truly-concurrent transitions which processes cannot perform in Fusion but their translations are capable of performing in SG. We called *hypoequivalence* the relation obtained on graphs by such observations. Hypoequivalence implies hyperequivalence, but do non know whether the opposite holds. Were it the case, it would amount to the first full-abstraction result relating process calculi and graph rewriting.

Bisimulation equivalence is also useful for proving the correctness of system implementations, or (dually) of optimisation steps. We developed a simple application where the specification of a simple component, called non-deterministic commuter, is shown to be equivalent to an implementation in which the internal communication channel is replaced by a local net. The equivalence relies on the particular "sand clock" shape of the net. Differently shaped nets would make the equivalence fail, e.g. by introducing a degree of parallelism which non-deterministic commuters do not have. This responds to the challenge proposed in [KM01] for an equivalence capable of determining the degree of parallelism in a network.

References

[CG00] L. Cardelli and A.D. Gordon. Mobile Ambients. *Theoretical Computer Science*, 1(240):177–213, 2000.

[CT05] P. Cenciarelli, , and A. Tiberi. Rational Unification in 28 Characters. *Electronic Notes in Theoretical Computer Science*, 127-5:3–20, 2005.

[CTT05] P. Cenciarelli, I. Talamo, and A. Tiberi. Ambient Graph Rewriting. *Electronic Notes in Theoretical Computer Science*, 117:335–351, 2005.

[DM87] Pierpaolo Degano and Ugo Montanari. A model for distributed systems based on graph rewriting. *Journal of the ACM (JACM)*, 34:411–449, 1987.

[FMT01] G. Ferrari, U. Montanari, and E. Tuosto. A LTS semantics of ambients via graph synchronization with mobility. *Proc.ITCS 01, Springer LNCS 2202*, 2001.

[HM01] Hirsch, Dan and Montanari, Ugo. Synchronized hyperedge replacement with name mobility: A graphical calculus for name mobility. In Larsen, Kim and Nielsen, Mogens, editors, *12th International Conference in Concurrency Theory (CONCUR 2001)*, volume 2154 of *Lecture Notes in Computer Science*, pages 121–136, Aalborg, Denmark, August 2001. Springer Verlag.

[KM01] Barbara König and Ugo Montanari. Observational equivalence for synchronized graph rewriting with mobility. In *TACS: 4th International Conference on Theoretical Aspects of Computer Software*, 2001.

[LM03] I. Lanese and U. Montanari. A graphical fusion calculus. In *Proc. of COMETA'03*, 2003.

[PV98] J. Parrow and B. Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. *Proc. LICS'98*, 1998.

[RH01] J. Riely and M. Hennessy. Distributed Processes and Location Failures. *Theoretical Computer Science*, 266:693–735, 2001.

Appendices

Proof of Lemma 1.

A transition Ξ in $(\nu x\, \Lambda) * \Theta$ is of the form $(\nu x\, G)|F \xrightarrow{\Lambda\cup\Theta} \rho(H'\,|K)$, where $H' = H$ if $x \in dng(\Lambda)$ and $H' = \nu x\, H$ otherwise. Note first that, since $\nu x\, \Lambda \# \Theta$ implies $\Lambda \# \Theta$, the transition $G|F \xrightarrow{\Lambda\cup\Theta} \rho(H|K)$ is in $\Lambda * \Theta$. Let $x \in dng(\Lambda)$. Since Ξ is a transition, $x \notin \rho(H|K)$ by condition 3 of definition 1. Then, independently of whether [open] or [res] is applied to $G|F \xrightarrow{\Lambda\cup\Theta} \rho(H|K)$, we obtain $(\nu x\, G)|F = \nu x\, (G|F) \xrightarrow{\Lambda\cup\Theta} \rho(H'\,|K)$ in $\nu x\, (\Lambda * \Theta)$ as required. Otherwise, let $x \notin dng(\Lambda)$. If $x \notin |H|$, then $H' = \nu x\, H = H$. If $x \in dng(\Lambda\cup\Theta)$ then Ξ is the result of applying [open] to $G|F \xrightarrow{\Lambda\cup\Theta} \rho(H|K)$. On the other hand, since Ξ satisfies condition 3, $x \notin dng(\Lambda \cup \Theta)$ implies $x \notin \rho(H|K)$, and hence Ξ is obtained again from $\Lambda \cup \Theta$ by [res]. Finally, if $x \in |H|$, the synchronisation of $\nu x\, G \xrightarrow{\Lambda} \nu x\, H$ with Θ has no effect on x. Hence $\rho(\nu x\, H|K) = \nu x\, \rho(H|K)$ and moreover, again by condition 3, $x \notin dng(\Lambda \cup \Theta)$. Then Ξ is the result of applying [res] to $(\Lambda \cup \Theta) \in \Lambda * \Theta$ as above.

Proof of Lemma 2.

We show one inclusion; the other is proven likewise.

Let $F|G \xrightarrow{\Lambda\cup\Theta} \rho(H|K)$ be in $\Lambda * \Theta$ and let $F|G|I \xrightarrow{\Lambda\cup\Theta\cup\Xi} \sigma(\rho(H|K)|J)$ be in $(\Lambda * \Theta) * \Xi$. It is easy to check that $(\Lambda \cup \Theta) \# \Xi$ if and only if $\Lambda \# \Xi$ and $\Theta \# \Xi$. Hence, by synchronising Ξ with Θ, and the result with Λ, we obtain a transition $F|G|I \xrightarrow{\Lambda\cup\Theta\cup\Xi} \sigma(H|\pi(K|J))$ in $\Lambda \# (\Theta \# \Xi)$. The result follows by noticing that $\sigma\rho = \sigma\pi = \sigma$ and hence:

$$\sigma(\rho(H|K)|J) = \sigma(H|K|J) = \sigma(H|\pi(K|J)).$$

Proof of Theorem 1.

Observe that, if x does not occur or it is restricted in the source of a transition Λ, then $\nu x\, \Lambda = \Lambda$. Hence, the hypothesis of lemma 1 can be assumed to hold for any application of [open] and [res] in a derivation tree. All such applications can therefore be moved toward the root of the tree. Moreover, since the source of productions are single-edged, all applications of [sync] can be reshuffled by lemma 2 so as to separate the actions of G from those of F.

Proof of Lemma 3.

By induction on the depth of the derivation tree. If Ψ is an axiom, the statement holds with Λ obtained as an instance of the same production as Ψ. This is because each tentacle in the source of a production is attached to a distinct node. As for the inductive steps, we only show the case where Ψ is the result of a synchronisation. The others are similar.

Let Ψ be a transition $h(G_1|G_2) \xrightarrow{\Psi_1 \cup \Psi_2} \rho(H_1|H_2)$ obtained by synchronising $hG_1 \xrightarrow{\Psi_1} H_1$ and $hG_2 \xrightarrow{\Psi_2} H_2$. By inductive hypothesis Ψ_1 and Ψ_2 are respectively of the form $hG_1 \xrightarrow{h\Lambda_1} \nu\mathbf{x}_1\rho_1(hK_1)$, and $hG_2 \xrightarrow{h\Lambda_2} \nu\mathbf{x}_2\rho_2(hK_2)$, and the transitions $G_1 \xrightarrow{\Lambda_1} K_1$ and $G_2 \xrightarrow{\Lambda_2} K_2$ are derivable. Since ρ_1 unifies Ψ_1, while ρ unifies $\Psi_1 \cup \Psi_2$, we have $\rho \circ \rho_1 = \rho$, and similarly for ρ_2. Moreover, $\mathbf{x}_1$ and $\mathbf{x}_2$ are not affected by the synchronisation of Ψ_1 and Ψ_2, and hence by ρ. Therefore, writing $\mathbf{x}$ for $\mathbf{x}_1\mathbf{x}_2$, we have:

$$\rho(H_1|H_2) = \rho(\nu\mathbf{x}_1\rho_1 hK_1|\nu\mathbf{x}_2\rho_2 hK_2) = \nu\mathbf{x}\,\rho(\rho_1 hK_1|\rho_2 hK_2) = \nu\mathbf{x}\,\rho\, h(K_1|K_2).$$

Let $G_1|\,G_2 \xrightarrow{\Lambda_1 \cup \Lambda_2} \sigma(K_1|K_2)$ result from synchronising Λ_1 and Λ_2, and let x be a node of $K_1|K_2$. Since $\sigma x \in [x]_{\Lambda_1 \cup \Lambda_2}$, we have $h\sigma x \in h[x]_{\Lambda_1 \cup \Lambda_2} \subseteq [hx]_{h(\Lambda_1 \cup \Lambda_2)}$. Hence, $\rho\, h\sigma x = \rho\, hx$. Then, $\rho\, h\sigma(K_1|K_2) = \rho\, h(K_1|K_2)$ as required.

Proof of Lemma 4.

Define R to be the relation $\{(\nu\mathbf{x}.\rho G, \nu\mathbf{x}.\rho F) \mid G \prec F\}$ where ρ is a fusion affecting only free nodes. Let R^α be its $\alpha-$closure. We show that R^α is a simulation.

First we prove that R is a simulation up to alpha-equivalence. More precisely, we show that, if $(a, b) \in R$ and $a \xrightarrow{\Lambda} c$, then $b \xrightarrow{\Theta} d$ with $\Lambda \sim \Theta$ and $(hc', kd') \in R$, where h, k are two injective substitutions as in the definition of simulation and c', d' are $\alpha-$variants of c and d.

Consider a pair $(\nu\mathbf{x}.\rho G, \nu\mathbf{x}.\rho F)$ in R and a transition $\nu\mathbf{x}.\rho G \xrightarrow{\Psi} H'$. By lemma 3 we know that Ψ can be obtained from $G \xrightarrow{\Lambda} H$ and hence it must be of the following form: $\nu\mathbf{x}.\rho G \xrightarrow{\rho\Lambda} \nu\mathbf{u}.\sigma\rho H$ where σ is a unifier of $\rho\Lambda$ and σ is the identity on $\mathbf{u}$. It is easy to check that $\nu\mathbf{x}.\rho F$ can make a transition $\nu\mathbf{x}.\rho F \xrightarrow{\rho\Theta} \nu\mathbf{v}.\sigma'\rho K$ such that $\rho\Lambda \sim \rho\Theta$ and that for some h, k satisfying the condition in the definition of simulation it holds that $hH < kK$. Thus we have to show that $(h'\nu\mathbf{u}.\sigma\rho H,\, k'\nu\mathbf{v}.\sigma'\rho K)$ is in R^α for a suitable choice of h', k'.

Observe first that σ (and σ') can be written as the composition of three fusions, affecting different kinds of nodes. That is $\sigma = \sigma_f \cdot \sigma_b \cdot \sigma_{bf}$ where σ_f affects nodes free in $\nu\mathbf{x}.\rho G$, σ_b nodes bound in both $\nu\mathbf{x}.\rho G$ and in $\nu\mathbf{u}.\sigma\rho H$ and σ_{bf} nodes bound in $\nu\mathbf{x}.\rho G$ but free in $\nu\mathbf{u}.\sigma\rho H$ (similarly for σ'). We can also assume σ_f and σ'_f to be the same function since $\rho\Lambda \sim \rho\Theta$.

Now we show that there are α variants of $\nu\mathbf{u}.\sigma\rho H$ and $\nu\mathbf{v}.\sigma'\rho K$ which can be rewritten as $\nu\mathbf{y}.\sigma_f \cdot \bar{\sigma}_b \cdot \sigma_{bf} \cdot \rho' \cdot h_b H$ and $\nu\mathbf{y}.\sigma_f \cdot \bar{\sigma}_b \cdot \sigma'_{bf} \cdot \rho' \cdot k_b K$, where h_b, k_b are h, k restricted to bound nodes of the graphs. Consider a class $[u_i]$ containing the image under ρ of a node y_i free in G. Because of the equivalence of Λ and Θ there must be a class $[v_j]$ observationally related to $[u_i]$ containing $\rho(y_i)$ and $h[u_i] = k[v_j]$. Extend ρ to a function ρ', defined on $h[u_i]$ so that $\rho'(h[u_i]) = \rho[u_i]$ and define $\bar{\sigma}_b(\rho[u_i]) = \rho(y_i)$. On the other hand, if $[u_i]$ contains only nodes already bound in G, these nodes can not be affected by ρ. Moreover in Θ there must be a class $[v_j]$ that is observationally related to $[u_i]$ and $h[u_i] = k[v_j]$. Choose a representative z and define $\bar{\sigma}_b(k[u_i]) = \bar{\sigma}_b(h[v_j]) = z$.

Next we define two injective substitutions h' and k' and a function $\bar{\sigma}_{bf}$ such that $h' \cdot \sigma_{bf} \cdot \rho = \bar{\sigma}_{bf} \cdot \rho' \cdot h$ and $k' \cdot \sigma'_{bf} \cdot \rho = \bar{\sigma}_{bf} \cdot \rho' \cdot k$. Observe that for each class $[x_i]$ that is affected by σ_{bf}, there is an observationally related class $[z_i]$ affected by σ'_{bf}, hence these classes can be equated by h' and k'. Consider $[x_i]$ containing the image under ρ of a node y free in G and let $[z_j]$ be the class observationally related to it. Notice that $[z_j]$ must contain $\rho(y)$. Define $h' \cdot \sigma_{bf}[x_i] = \rho(y) = k' \cdot \sigma'_{bf}[z_j]$. Notice that $h' \cdot \sigma_{bf}$ and $k' \cdot \sigma'_{bf}$ on these classes can be written as a $\bar{\sigma}_{bf}$ defined on $\rho' h[x_i] = \rho' k[z_j]$. Consider now the classes $[x_i]$, $[z_i]$ which are observationally related, and contain no free node of G (and hence of F). The fusion ρ must be the identity on these classes. Define h', k' so that $h' \cdot \sigma_{bf}[x_i] = h \cdot \sigma_{bf}[x_i]$ and $k' \cdot \sigma'_{bf}[z_i] = h' \cdot \sigma_{bf}[x_i]$. The fusion $\bar{\sigma}_{bf}$ defined above can be extended so that on these classes $h' \cdot \sigma_{bf} = \bar{\sigma}_{bf} \cdot h$ and $k' \cdot \sigma'_{bf} = \bar{\sigma}_{bf} \cdot k$. Thus, by applying h' and k', we obtain two graphs of the following form:

$$\nu \mathbf{y}.\sigma_f \cdot \bar{\sigma}_b \cdot \bar{\sigma}_{bf} \cdot \rho' \cdot hH$$
$$\nu \mathbf{y}.\sigma_f \cdot \bar{\sigma}_b \cdot \bar{\sigma}_{bf} \cdot \rho' \cdot kK$$

which by definition are in R. This concludes the proof that R is a simulation up to alpha-equivalence.

Consider a pair $(a', b') \in R^\alpha$; if $a' \xrightarrow{\Lambda'} c''$ then $b' \xrightarrow{\Theta'} d''$ with $\Lambda' \sim \Theta'$, because there are two equivalent transitions, $a \xrightarrow{\Lambda} c$ equivalent to Λ', and $b \xrightarrow{\Theta} d$ equivalent to Θ'. Moreover, since $(hc', kd') \in R$ and c'' and d'' are almost α equivalent to c' and d' (they can differ on nodes that were restricted but have been opened), we can define h' and k' in agreement with the definition of simulation so that $h'c''$ is $\alpha-$equivalent to hc' and $k'd''$ to kd', hence $(h'c'', k'd'')$ is in R^α. Hence, R^α is a simulation.

Proof of Corollary 2.

Let $G \prec F \prec I$ and let $G \xrightarrow{\Lambda} H$ be a transition. There exist $F \xrightarrow{\Theta} K$ and $I \xrightarrow{\Xi} J$ such that Λ, Θ and Ξ are equivalent and moreover $hH \prec kK$ for suitable h and k. Since $F \prec I$, so are kF and kI by lemma 4. By lemma 3, and since k is injective, there exists a transition $kF \xrightarrow{k\Theta} kK$ which is simulated by $kI \xrightarrow{k\Xi} kJ$, with $h'kK \prec k'kJ$ for suitable h' and k'. Then, again by lemma 4:

$$h'hH \prec h'kK \prec k'kJ.$$

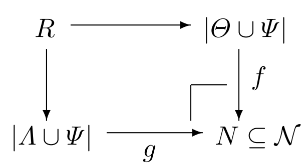
Proof of Theorem 3.

Let G, F and U be graphs such that $G\#U$, $F\#U$ and $G \prec F$. For simplicity we consider the transitions of $G|U$ and $F|U$, rather than $\nu \boldsymbol{x}\, G|U$ and $\nu \boldsymbol{x}\, F|U$. The general proof is only slightly more complicated. So, let $G|U \xrightarrow{\Phi} W$ be a transition. We start by exhibiting an equivalent transition $F|U \to Z$. Define the *focus* in a derivation of a transition Ξ to be Ξ itself if it is an axiom or the conclusion of a [sync]; otherwise the focus is that of the sub-derivation of the premise of Ξ. By Lemma 1, there exists a derivation of Φ in which all the actions of G are separated from those of U. Let $G_0|U_0 \xrightarrow{\Lambda\cup\Psi} \rho(H_0 \,|\, V_0)$ be its focus, with $G_0 \xrightarrow{\Lambda} H_0$ and $U_0 \xrightarrow{\Psi} V_0$ as premises, where $G = \nu\boldsymbol{x_0}\, G_0$ and $U = \nu\boldsymbol{u_0}\, U_0$. Since, by assumption, $\boldsymbol{x_0} \cap |U| = \emptyset$, there are no unsynchronised actions on $\boldsymbol{x_0}$ in Λ (such actions would otherwise occur unsynchronised in Φ as well), and a transition $G \xrightarrow{\Lambda} H$ can therefore be derived from $G_0 \xrightarrow{\Lambda} H_0$. Since $G \prec F$, there exists a transition $F \xrightarrow{\Theta} K$ which is equivalent to Λ, and such that $h\,H \prec k\,K$ for h and k as in definition 3. By lemmas 1 and 2, there exists a derivation of Θ where all synchronisations are applied first. Let $F_0 \xrightarrow{\Theta} H_0$ be its focus, with $F = \nu\boldsymbol{y_0}\, F_0$, and let $F_0|U_0 \xrightarrow{\Theta\cup\Psi} \sigma(K_0 \,|\, V_0)$ be derived by synchronising $U_0 \xrightarrow{\Psi} V_0$ with $F_0 \xrightarrow{\Theta} H_0$. By restricting $\Theta \cup \Psi$ on $\boldsymbol{y_0}$ and $\boldsymbol{u_0}$, a transition $F|U \xrightarrow{\Theta\cup\Psi} Z$ is obtained, which is easily shown to be equivalent to Φ. Then we exhibit two functions $h' : |W| \to \mathcal{N}$ and $k' : |Z| \to \mathcal{N}$ such that $h'W\,\mathcal{R}\,k'Z$ as required by Definition 3.

Let $H = \nu\boldsymbol{x_1}\, H_0$ and $V = \nu\boldsymbol{w_1}\, V_0$, with $\boldsymbol{x_1} \subseteq \boldsymbol{x_0}$ and $\boldsymbol{w_1} \subseteq \boldsymbol{u_0}$. Since the nodes in $\boldsymbol{x_1}$ and $\boldsymbol{w_1}$ are not affected by the synchronisation of Λ and Ψ, $\nu\boldsymbol{x_1}\,\rho H_0 = \rho H$ and $\nu\boldsymbol{w_1}\,\rho V_0 = \rho V$. Similarly, for σ. Hence, W and Z are respectively of the form $\nu\boldsymbol{x_2}\,\rho(H|V)$ and $\nu\boldsymbol{y_2}\,\sigma(K|V)$, where $\boldsymbol{x_2} \subseteq \boldsymbol{x_0}\boldsymbol{u_0}$ are the dangling nodes of Λ and Ψ which are fused by the synchronisation, and similarly for $\boldsymbol{y_2} \subseteq \boldsymbol{y_0}\boldsymbol{u_0}$. Summarising: we look for suitable functions h' and k' such that

$$h'\,\nu\boldsymbol{x_2}\,\rho(H|V)\;\mathcal{R}\;k'\,\nu\boldsymbol{y_2}\,\sigma(K|V). \tag{1}$$

Calling *interface* of a transition Ξ the set of its unsynchronised parameters $|\Xi|_\iota = \{(x, a, i) \in \mathcal{P} \,:\, \Xi x = \{(a, \boldsymbol{y})\}$ and $0 \leq i \leq |\boldsymbol{y}|\}$, we let $x\,R\,y$ hold on $|\Lambda\cup\Psi| \times |\Theta\cup\Psi|$ precisely when x and y are related either observationally or by the interface of Ψ, that is: $x \overset{\Lambda\cup\Psi}{=} o \overset{\Theta\cup\Psi}{=} y$ for some $o \in |\Psi|_\iota$. The projections of R form a pushout diagram



in which N is a set of *fresh* nodes, and g and f are such that $g\,x = f\,y$ if and only if $x\,R\,y$. Let $\xi : \mathit{fn}\,(hH) \cup \mathit{fn}\,(kK) \to \mathcal{N}$ be the node renaming function mapping x to $g(h^{-1}x)$ if $x \in \mathit{fn}\,(hH)$, or else to $f(k^{-1}x)$ if $x \in \mathit{fn}\,(kK)$. This is a good definition because, if $x \in \mathit{fn}\,(hH) \cup \mathit{fn}\,(kK)$, then $g(h^{-1}x) = f(k^{-1}x)$. It is easy to check that $g(\rho H) = \xi(hH)$ and $f(\sigma K) = \xi(kK)$. Since $hH \prec kK$, by Lemma 4:

$$g(\rho H) \prec f(\sigma K). \tag{2}$$

Moreover, let x be a free node of V. Either x is free in U, in which case x (as an object of $|\mathit{\Lambda}\cup\mathit{\Psi}|$) is observationally related with itself (as an object of $|\mathit{\Theta}\cup\mathit{\Psi}|$) or there exists a parameter o in the interface of $\mathit{\Psi}$ such that $x \stackrel{\mathit{\Lambda}\cup\mathit{\Psi}}{=} o \stackrel{\mathit{\Theta}\cup\mathit{\Psi}}{=} x$. In both cases $\rho x \,\mathcal{R}\, \sigma x$, and therefore

$$g(\rho V) = f(\sigma V). \tag{3}$$

Finally, there exists a vector $\boldsymbol{v}$ such that

$$\nu\boldsymbol{v}\, g\rho(H|V) = \nu(g\boldsymbol{x_2})\, g\rho(H|V) \text{ and} \tag{4}$$

$$\nu\boldsymbol{v}\, f\sigma(K|V) = \nu(f\boldsymbol{y_2})\, f\sigma(K|V). \tag{5}$$

In fact, any node $x \in \boldsymbol{x_2}$ must be either dangling in $\mathit{\Lambda}$ or in $\mathit{\Psi}$. Since $\mathit{\Lambda}$ and $\mathit{\Theta}$ are equivalent, there must exist $y \in |\mathit{\Theta}|$ such that $x \stackrel{\mathit{\Lambda}\cup\mathit{\Psi}}{=} o \stackrel{\mathit{\Theta}\cup\mathit{\Psi}}{=} y$ for some $o \in |\mathit{\Psi}|_\iota$. Hence, $x\, R\, y$ holds. Then, either $[y]_{\mathit{\Theta}\cup\mathit{\Psi}} \cap |\sigma(K|V)| = \emptyset$, in which case $gx \notin f\sigma(K|V)$, or else there exists $y' \in \boldsymbol{y_2}$ such that $gx = fy'$. In either cases $\nu(gx)\, \nu(f\boldsymbol{y_2}) f\sigma(K|V) = \nu(f\boldsymbol{y_2})\, f\sigma(K|V)$. The dual argument applies when $x \in \boldsymbol{y_2}$. So, we obtain the equations (4) and (5) by taking $\boldsymbol{v}$ to be $g\boldsymbol{x_2} \cup f\boldsymbol{y_2}$.

Now, define h' to be the restriction of g to the free nodes of $\nu\boldsymbol{x_2}\, \rho(H|V)$ and k' the restriction of f to the free nodes of $\nu\boldsymbol{y_2}\, \sigma(K|V)$. Since ρ and σ are unifiers, both h' and k' are injective and moreover, since they do not affect the nodes in $\boldsymbol{x_2}$ and $\boldsymbol{y_2}$, they are such that $h'x = k'y$ if and only if x and y are observationally related, as required. Then, noticing that $h'\nu\boldsymbol{x_2}\, \rho(H|V)$ and $\nu(g\boldsymbol{x_2})\, g\rho(H|V)$ are alpha-equivalent, and so are $k'\nu\boldsymbol{y_2}\, \sigma(K|V)$ and $\nu(f\boldsymbol{x_2})\, f\sigma(K|V)$, we can finally prove the relation (1):

$$\begin{aligned}
h'\nu\boldsymbol{x_2}\, \rho(H|V) \stackrel{\alpha}{=} \nu(g\boldsymbol{x_2})\, g\rho(H|V) = \nu\boldsymbol{v}\, g\rho(H|V) \qquad & \text{by (4)} \\
\nu\boldsymbol{v}\, g\rho(H|V)\; \mathcal{S}\; \nu\boldsymbol{v} f\sigma(K|V) \qquad & \text{by (2) and (3)} \\
\nu\boldsymbol{v} f\sigma(K|V) = \nu(f\boldsymbol{y_2})\, f\sigma(K|V) \stackrel{\alpha}{=} k'\nu\boldsymbol{y_2}\, \sigma(K|V) \qquad & \text{by (5).}
\end{aligned}$$

Proof of Proposition 1.

Writing C_G the commuter obtained from a graph G of sockets and connectors as above, the set of all pairs of the form (G, C_G) is a bisimulation. In fact, any transition of C_G must involve the synchronisation of one (because the infrastructure is acyclic) and only one (because a pivot exists) input/output pair of sockets. By an easy check, the echoing actions $(u_1, \overline{a}, v)$ and (u_2, a, v) performed by the connectors preseve the equivalence of transitions.

Proof of Theorem 7.

We show that $\mathcal{S} = \{(P, Q) \,|\, [\![P]\!] \mathrel{\dot\sim} [\![Q]\!]\,\}$ is a simulation. Then, the results follows because $\mathcal{S}$ is clearly symmetric and closed under substitution (because so is $\dot\sim$). Spelling out the definition of simulation [PV98], this means to prove that, for all $P \xrightarrow{\gamma} P'$ such that $bn(\gamma) \cap fn(Q) = \emptyset$, there exists $Q \xrightarrow{\gamma} Q'$ such that $\sigma Q'$ simulates $\sigma P'$ for a substitutive effect σ of γ. Consider a generic transition of $P \xrightarrow{\gamma} P'$ such that $bn(\gamma) \cap fn(Q) = \emptyset$. Since $\dot\sim$ is closed under α equivalence and arbitrary fusions, and because of theorem 6, we know that there exists an α-variant of $[\![P]\!]$ which can go into $[\![\sigma P']\!]$ with an action set Λ that corresponds exactly to γ, where σ is a substitutive effect of γ. Since $[\![P]\!] \mathrel{\dot\sim} [\![Q]\!]$ and since $bn(\gamma) \cap fn(Q) = \emptyset$, we also know that there is an α-variant of $[\![Q]\!]$ which can make a transition to $[\![\sigma Q']\!]$ with an action set Θ that corresponds to γ and is equivalent to Λ, where σ is as above. Moreover, we also know that two injective renamings h, k exist such that $h[\![\sigma P']\!] \mathrel{\dot\sim} k[\![\sigma Q']\!]$, but since the two actions sets used above are identical, and since $\dot\sim$ is closed under fusions, we can assume h, k to be identities. Hence Q simulates P as required.

The fusion Calculus

Here we sketch the operational semantics of the Fusion Calculus (see [PV98] for more detail). Transitions are labeled by actions of two kinds: input/output and fusion. Input/output actions can be *bound*, as in $(\boldsymbol{x})\, a\, \boldsymbol{y}$, thus recording that the action opens the scope of $\boldsymbol{x}$. Fusions occur as a result of matching the names that are passed during communication, and they are represented in labels by equivalence relations. In particular, $\{\boldsymbol{x} = \boldsymbol{y}\}$ denotes the smallest equivalence relation φ such that $(x_i, y_i) \in \varphi$. Fusions are applied *only* when they occur under the scope of some bound name x. In such a case, if a *free* name y exists such that $(x, y) \in \varphi$, then y substitutes x in the right hand side of the transition, and the relation labeling the transition is restricted as follows: $\varphi \restriction x = \varphi \cap (\mathcal{N} - \{x\})^2 \cup \{(x, x)\}$.

[PREF] $\dfrac{-}{\alpha.P \xrightarrow{\alpha} P}$ [PAR] $\dfrac{P \xrightarrow{\alpha} P'}{P \,|\, Q \xrightarrow{\alpha} P' \,|\, Q}$

[COM] $\dfrac{P \xrightarrow{x\boldsymbol{y}} P' \quad Q \xrightarrow{\overline{x}\boldsymbol{z}} Q'}{P \,|\, Q \xrightarrow{\{\boldsymbol{y}=\boldsymbol{z}\}} P' \,|\, Q'}$ [SCOPE] $\dfrac{P \xrightarrow{\varphi} Q \quad x\,\varphi\, y \quad x \neq y}{(x)P \xrightarrow{\varphi\restriction x} [y/x]Q}$

[PASS] $\dfrac{P \xrightarrow{\alpha} Q \quad x \notin n(\alpha)}{(x)P \xrightarrow{\alpha} (x)Q}$ [OPEN] $\dfrac{P \xrightarrow{(\boldsymbol{y})a\boldsymbol{z}} Q \quad x \in \boldsymbol{z} - \boldsymbol{y} \quad a \notin \{x, \overline{x}\}}{(x)P \xrightarrow{(x\boldsymbol{y})a\boldsymbol{z}} Q}$