Semantics of global view of choreographies[☆]

Emilio Tuosto [a,*], Roberto Guanciale[b,**]

[a] *University of Leicester, UK*
[b] *KTH, Sweden*

Abstract

We propose two abstract semantics of the global view of choreographies given in terms of partial orders. The first semantics is formalised as pomsets of communication events while the second one is based on hypergraphs of events. These semantics can accommodate different levels of abstractions. We discuss the adequacy of our models by considering their relation with communicating machines, that we use to formalise the local view. Our approach increases expressiveness and allows us to overcome some limitations that affect alternative semantics of global views. This will be illustrated by discussing some interesting examples. Finally, we show that the two semantics are equivalent and have different merits. More precisely, the semantics based on pomsets yields a more elegant presentation, but it is less suitable for implementation. The semantics based on hypergraphs instead is amenable to a straightforward implementation.

Keywords: Choreography, communicating finite-state machines, global graphs, hypergraphs, pomsets, semantics.

1. Introduction

Distributed applications are nowadays widespread. Rarely applications are stand-alone anymore: software is today conceived to dynamically interact with other applications. The combination of ubiquitous connectivity and the evolution of portable or wearable devices (such as smart phones or watches) practically changed the nature of software and it is also determining new approaches to software development [28]. Big vendors as well as small software companies, have to satisfy the appetite of users who want data and applications 'always handy'. Also, software is becoming more and more important as it increasingly deals with delicate societal and economical aspects. Distributed applications are used in many aspects of our lives, from handling commercial transactions

[☆]The authors are grateful to the reviewers for the helpful comments and to Ivan Lanese for useful comments.

This report extends the published version of our journal paper with full proofs.

[*]Principal Corresponding author
[**]Corresponding author

to social interaction, to providing e-health and e-government services. For such (and other) reasons, developers are required to carefully design their applications so that unintended behaviours do not happen at runtime.

The problem. It is widely accepted that distributed systems and applications are not easy to design, implement, verify, deploy, and maintain. A key issue to tackle is the coordination of distributed components. For this, two primary (and in our view, complementary) approaches have been identified: *orchestration* and *choreography*. We focus here on the latter. More precisely, we propose a new semantics of a model of choreographies for message-passing software. We argue that our semantics offer a framework that (*i*) generalises existing linguistic constructs of choreographies by removing some unnecessary constraints adopted elsewhere, (*ii*) allows architects to design the coordination of distributed applications without forcing them to consider low level details, and (*iii*) enables the possibility of tool support at earlier stages of the development.

Choreographies have been advocated as a suitable methodology for the design and analysis of distributed applications since, unlike models based on orchestration, they do not require an explicit coordinator. Roughly, a choreography describes how two or more distributed components coordinate with each other by exchanging messages. Among the possible interpretations of what choreographies are (see [2] for a discussion and references), we embrace the one suggested by W3C's [23]:

> Using the Web Services Choreography specification, a contract containing a global definition of the common ordering conditions and constraints under which messages are exchanged, is produced that describes, from a **global viewpoint** [...] observable behaviour [...]. Each party can then use the **global definition** to build and test solutions that conform to it. The global specification is in turn realised by combination of the resulting **local systems** [...]

This description conceptualises two views, a **global** and a **local** one, which enable the relations represented by the following diagram:

$$\text{Global view} \xrightarrow{\text{projection}} \text{Local view} \xleftarrow{\text{comply}} \text{Local systems} \tag{1}$$

where the operation of 'projection' produces the local view from the global one and the operation 'comply' verifies that the behaviour of each component adheres to the one of the corresponding local view.

For diagram (1) to make sense, precise semantics should be fixed for the global and the local views. The semantics of the latter is well understood: it directly emanates from the adopted communication model. In fact, the local view details how communications take place. For instance, in a channel-based communication model, the local view may specify what is the behaviour of each component in terms of its send/receive actions.

What is instead "the semantics of the global view"? We investigate such question here. And, after making it more precise, we propose a new semantic framework for global views and discuss its advantages over existing frameworks.

Before continuing, we remark that the relations among views and systems of choreographies are richer than those depicted in diagram (1). For instance, local views can also be 'compiled' into template code for the local components and the projection operation may have an "inverse" (cf. [26]). Those aspects are not in the scope of this paper though.

A view of global views. The W3C description above is intriguing, however it is not very enlightening to understand what a global view is; basically it says that a global view has to describe the observable behaviour from a global viewpoint...a bit too much circularity for a definition!

We will consider global views as high-level descriptions of systems abstracting away some aspects in order to offer a *holistic* understanding of the communication behaviour of distributed systems. (This is still vague, but will become precise in the forthcoming sections.) In a global view, components are not taken anymore in *isolation*. Rather they are *specified together*, while *forgetting some details*. For us, this will mean to describe the protocol of interaction of a system in a way that makes it explicit how messages are actually exchanged among components. For instance, in our example based on channels, the global view may abstract away from send/receive actions and use *interactions* as the unit of coordination [7].

The idea depicted in diagram (1) is beautiful. To our best knowledge, it has been firstly formally pursued in [19] (later refined in [21]) and then followed by others. The main reason that makes diagram (1) attractive is the interplay between global and local artefacts[1] as it fosters some of the best principles of computer science and of software engineering:

Separation of concerns The *intrinsic logic* of the distributed coordination is expressed in and analysed on global artefacts, while the local artefacts refine such logic at lower levels of abstraction.

Modular software development life-cycle The W3C description above yields a distinctive element of choreographies which makes them appealing (also to practitioners). Choreographies allow independent development: components can harmoniously interact when proven compliant to their corresponding local view. Global and local views yield the "blueprints" of systems as a whole and of each component, respectively.

Principled design A choreographic framework orbits around the following implication:

if *cond*(global artefact) **then** *behave*(*projection*(global artefact))

[1] We will use the term 'artefact' when referring to actual specifications embodying the global/local views. Such embodiments may assume various forms: types [21], programs [13], graphs and automata [26, 15], executable models [23, 1], etc. Typically, the literature uses the (overloaded) word 'model' to refer to this flora of embodiments. We prefer the word 'artefact' because it allows us to refer to different contexts and different abstraction levels without attaching yet another meaning to 'model'.

that is, proving that a correctness condition *cond* holds on an abstraction (the global artefacts) guarantees that the system is well behaved, provided that the local artefacts are "compiled" from the global ones via a *projection* operation that preserves behaviour.

Therefore, providing suitable semantics for global artefacts is worthwhile: it gives precise algorithms and establishes precise relations between specifications of distributed systems (the global artefacts) and their refinements (the local artefacts). It is worth remarking that the languages we adopt here for global and local views are specification languages. In particular, we abstract away from concrete programming mechanisms (e.g., adopting non-deterministic constructs), avoiding to specify local computations (that is computations that do not require the interactions of components), and abstract away from data, in fact the term "message" here has to be understood as "data type" rather than actual value[2].

We have previously proposed in [18] a hypergraph semantics of choreographies, which is apt to be implemented (we are currently embedding the hypergraph semantics in ChorGram [25], a toolchain for choreographic modelling). The new pomset semantics proposed here is more abstract (and, we believe, more elegant) than the hypergraph semantics. In particular, the use of pomsets simplifies the notion of projection and generalises well-sequencedness and well-branchedness. Besides, defining the pomsets semantics and showing its equivalence with the hypergraph-based semantics, in this paper we have revised and simplified the presentation in [18].

Contributions Our main technical contributions are two semantics of global views of choreographies, modelled as global graphs [15]. Our first semantic framework relies on *pomsets* [29, 16], which provide a simple and elegant theoretical presentation of concurrency and distribution. We note that pomsets have been adopted also in [22] as a semantic framework of message-sequence charts. Our second semantic framework relies on a model of hypergraphs. This framework is more suitable for developing analysis tools. As we show in the paper, the two semantic models are equivalent, in the sense that they yield the same causality relations of choreographies. To the best of our knowledge, our semantics generalise existing approaches; we demonstrate this by giving examples of global views usually discarded in other approaches.

We note that in [14] (grammars of) hypergraphs have been used as a formal operational semantics of distributed computations. Here we use hypergraphs as the semantic codomain in a denotational style.

Outline Section 2 highlights the advantages of abstract semantics of global views. The syntax of our language of global artefacts is in Section 3 with some examples used in later sections to explain some of our constructions. Section 4 presents the abstract semantics of global artefacts by capturing their causal dependencies using pomsets. A first technical advantage of our semantics is provided by the definition of *well-branched* choices, explained through some illustrative

[2]The term "message" could be misleading, but we adopt it because it is widespread in the literature.

examples. Our semantics is used to identify all licit traces of a choreography, so to precisely characterise the behaviour expected by the specification. Section 5 first recalls the communicating finite state machines (that are used to formalise the local behaviours) and then defines the projection of global artefacts on communicating machines. The main technical results establish that well-branched choreographies yield deadlock-free systems (Theorem 1) whose behaviour is included in the behaviours specified by the global view (Theorem 2). Section 6 presents an alternative semantics based on hypergraphs and introduces the notion of *reflection*, crucial for our model. Theorem 3 establishes equivalence between the two semantics (detailed proofs are in Appendix B). Section 7 draws some conclusions.

2. Why going abstract?

We propose a denotational semantics of global views of choreographies based on partial orders. Our new semantics is more abstract than existing ones as it is not based on traces and it makes minimal assumptions on how messages are exchanged at lower levels. Conceptually this is easy to achieve. We fix a specification language of global artefacts and we interpret a specification as a set of "minimal and natural" causal dependencies among the messages. We then define when a global artefact is sound, namely when its causal dependencies are consistent so that they are amenable to be executed distributively by some local artefacts, regardless of the underlying message passing semantics. We illustrate the advantages of our approach by adopting a rather liberal language of global artefacts inspired by global graphs [15]. We then show the relation of such language on a local view featuring local artefacts as communicating machines [5].

As discussed in Section 1, many authors have adopted the idea in diagram (1) and several semantics of (models of) the global view have been introduced. We distinguish two broad classes.[3]

The first class consists of the approaches where the semantics of the global view is obtained by composing the semantics of the local view. For instance, this class includes the seminal work on global types [20]. The idea is to give an explicit semantics of local artefacts and define the semantics of global artefacts in terms of the semantics of their projections. In the case of global types, the projection yields local types, that are process algebras equipped with an operational semantics. This approach is ubiquitous in the literature based on behavioural types and it has also been adopted in [26] where global artefacts are global graphs [15] and local artefacts are communicating finite-state machines [5].

In the other class, the semantics of global view is defined explicitly. A first such semantics appeared in [6] where a language of global views syntactically equivalent to ours has been considered (the semantics is however different from ours since in [6] synchrony is assumed both at the global and local level). Also, the global types of [20] are refined and equipped with an operational semantics in [21].

[3] We mention a tiny portion of the literature in way of example; no claim of exhaustiveness.

In [9] an operational interleaving semantics is defined for global specifications while in [3] a trace-based semantics is given. In both cases, the idea is to "split" the interactions in the global view into its constituent send/receive actions. A slightly more abstract interleaving operational semantics of global types is given in [8] where interleaving is attained by swapping independent interactions, rather than manipulating traces of send/receive actions. A denotational trace-based model of global views is given in [30] (unlike ours, this semantics is based on synchronous communication and respects the sequential compositions of projections on well-formed global views (Theorem 2 in [30]). In this category we also put approaches like [12, 13] where global artefacts become *global programs* with an operational semantics. In this area authors have proposed to mitigate the limitations usually present in languages for global views using operational approaches [6, 24, 13, 8].

The classes above contain perfectly reasonable approaches. After all, from a theoretical perspective, we just need a semantics for the global view; whatever "fits" with the semantics of the local view would do. We believe that an abstract semantics of global view like the one we propose here makes it easier reason about global artefacts and to develop the algorithmic and tool support for such reasoning.[4] Trace-based (denotational or operational) semantics of global views are less suitable in this respect because they require the designer to consider the dynamics of the interactions in terms of (an approximation of) their execution. In other words, existing semantics of global views are "too concrete" and require to manipulate traces in order to capture relevant aspects of the design. For instance, many approaches take traces up-to permutations of non causally-related events to "mimick" asynchrony at the global level. Another drawback present in some of the approaches based on traces is that they make the semantics of the global view a *dependent variable* of the semantics of the local one brings in the following issues. For instance, to guarantee some properties it is sometimes necessary to adopt (syntactic) restrictions imposed by the underlying communication infrastructure. In several cases these restrictions limit the expressiveness of the language at hand (for instance, languages featuring the parallel composition of global artefacts do not allow components involved in more than one parallel thread). Another example of such dependency is the introduction in the semantics of the global views of low-level elements (e.g., to guarantee order-preserving asynchronous outputs).

To sum up, we suggest that an abstract semantics of global views yields a more suitable framework to foster the interplay between global and local views described in Section 1.

[4] We are currently working on the definition and implementation of algorithms to establish equivalences of global views based on the abstract semantics presented in this paper.

3. Global views as graphs

Let $\mathcal{P}$ be a set of *participants* (ranged over by A, B, etc.), $\mathcal{M}$ a set of *messages* (ranged over by m, x, etc.), and $\mathcal{K}$ a set of *control points* (ranged over by i, j, etc.). We take $\mathcal{P}$, $\mathcal{M}$, and $\mathcal{K}$ pairwise disjoint. The participants of a choreography exchange messages to coordinate with each other. In the global view, this is modelled with *interactions*[5] $\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$, representing the fact that participant A sends message m to participant B, which is expected to receive m. A *global choreography* (g-choreography for short) is a term G derived by the following grammar (recursion/iteration is omitted for simplicity as discussed in Section 7)

$$\mathsf{G} \;::=\; \mathbf{0} \;\big|\; \mathsf{i}\colon \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B} \;\big|\; \mathsf{G};\mathsf{G}' \;\big|\; \mathsf{i}{:}(\mathsf{G}|\mathsf{G}') \;\big|\; \mathsf{i}{:}(\mathsf{G}+\mathsf{G}') \tag{2}$$

A g-choreography can be empty, a simple interaction, the sequential or parallel composition of g-choreographies, or the choice between two g-choreographies. We implicitly assume $\mathsf{A} \neq \mathsf{B}$ in interactions $\mathsf{i}\colon \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$. In (2), *control points*, denoted by i, tag nodes of g-choreographies that correspond to events of interest. For instance, the control point of an interaction identifies its output and input events, the one of a fork (resp. choice) identifies when a g-choreography splits in more threads (resp. branches). Note that the empty g-choreography is not tagged with a control point because it does not have any event; likewise for sequential composition $\mathsf{G};\mathsf{G}'$ whose events are already completely determined by the control points of G and G'. We assume that in a g-choreography G any two control points occurring in different positions are different, e.g., we cannot write $\mathsf{i}{:}(\mathsf{j}\colon \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B} | \mathsf{i}\colon \mathsf{C} \xrightarrow{\mathsf{y}} \mathsf{D})$. Let $\mathcal{G}$ be the set of g-choreographies and, for $\mathsf{G} \in \mathcal{G}$, let $\mathsf{cp}(\mathsf{G})$ denote the set of control points in G. Control points are a technical device and could be avoided.[6] The concrete representation of control point is immaterial; throughout the paper, for all $\mathsf{G} \in \mathcal{G}$, we will assume that $\mathsf{cp}(\mathsf{G})$ are strictly positive integers. Also, we may omit control points when immaterial, e.g., writing $\mathsf{G}+\mathsf{G}'$ instead of $\mathsf{i}{:}(\mathsf{G}+\mathsf{G}')$.

The syntax in (2) captures the structure of a visual language of acyclic[7] directed graphs. In fact, each g-choreography G can be represented as a rooted graph with a single "enter" (resp. "exit") node; namely, G has a distinguished *source* (resp. *sink*) node that can reach (resp. be reached by) any other node in G. Before commenting on our visual notation in Fig. 1, we remark that each fork or branch gate with control point i in our pictures will have a corresponding join and merge gate with control point $-\mathsf{i}$; note that negative control points will appear only in the visual notation of a global graph and not in its textual representation. This will also be used in Section 6.

In Fig. 1, dotted edges connect nodes $\bullet$ to boxed G to identify source and sink nodes of G. More precisely, a dotted edge from $\bullet$ to a boxed G means

[5] We depart from the usual notation $\mathsf{A} \to \mathsf{B} : \mathsf{m}$ to a have a more lightweight syntax.

[6] At the cost of adding technical complexity, one can automatically assign a unique identifier to such control points.

[7] Cycles are not considered for simplicity and can be easily added.

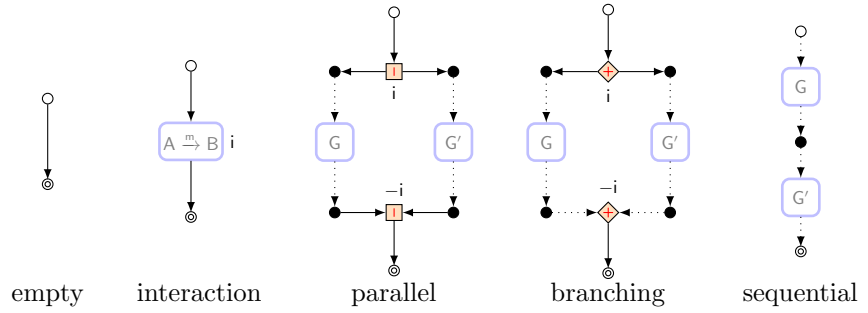


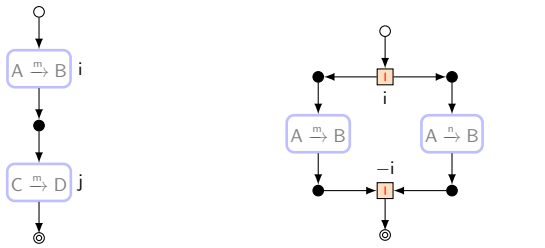
Figure 1: Our graphs: ○ is the source node, ◎ the sink one; other nodes are drawn as •

that • is the source of G; similarly, a dotted edge from a boxed G to • means that • is the sink of G. For instance, the dotted edges entering and leaving the g-choreography G in the sequential composition in Fig. 1 respectively identify the source and sink nodes of G while the other dotted edges identifies the source and sink nodes of G′; in other words, the sequential composition of G and G′ is obtained by coalescing the sink of G with the source of G′. Figs. 2a and 2b give an example of this construction for sequential and parallel composition respectively (in the latter figure we omitted the control points of interactions for readability). Fig. 2a shows why there is no need to assign a control point to sequential composition: there is no interesting event at the coalescing • node. Indeed, the pattern $\to \bullet \to$ in a graph does not yield any important event to trace and in the following we will often replace it with a simple edge $\to$). Note that the graph[8] $\mathsf{G}_{(2b)}$ in Fig. 2b represents a choreography where A sends B messages m and n in any order. Akin to BPMN [17] diagrams, our graphs yield a visual description of the distributed coordination of communicating components. In this respect, control points mark the nodes of the graph where communication and distributed work flow activities may happen. This is similar to BPMN where tasks (corresponding to our communication activities) and control gates (corresponding to our fork/join and branch/merge gates) have a special standing in the graphical notation.

Our graphs resemble the ones in [15, 26] the only differences being that

- by construction, forking and branching control points i have a corresponding join and merge control point −i;
- there is a unique sink control point with a unique incoming edge (as in [15, 26], there is also a unique source control point with a unique outgoing edge).

[8] We indexed our examples with the numbering of the figure they are in; therefore, we will hereafter avoid cross-referencing the figures.



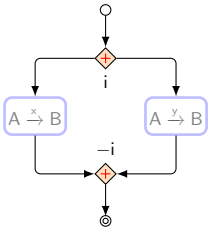
(a) $\mathsf{G}_{(2a)}$ (b) $\mathsf{G}_{(2b)} = \mathsf{i}{:}(\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B} | \mathsf{A} \xrightarrow{\mathsf{n}} \mathsf{B})$

Figure 2: Examples of choreographies

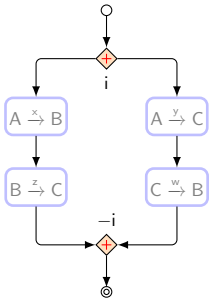
We now consider a few examples to give an intuition of g-choreographies and of the problems arising when defining their semantics. Indeed, not all g-choreographies of our language have a defined semantics. In particular, sequential and non-deterministic composition of choreographies can lead to choreographies that are impossible to realise. We will also use these examples through the paper to highlights some aspects of our constructions.

The problem of giving semantics to the sequential composition can be illustrated with the graph $\mathsf{G}_{(2a)}$. The graph specifies that the interaction between A and B must be followed (in some sense) by an interaction between C and D. Intuitively, this is not directly realisable in a distributed setting. In fact, neither the participant C nor D are informed about the termination of the interaction $\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$, making it impossible to respect the order of the interactions without further communication between A or B and C or D. In Section 4.2 we introduce the notion of *well-sequencedness* (cf. Definition 6) to identify sequential composition of choreographies that are meaningful. Typically, this problem is addressed by imposing syntactic constraints on sequential composition so to obtain the intended causal order on output and input events. Our notion of well-sequencedness is given instead at the semantic level so to accommodate more cases. Intuitively, we will require that $\mathsf{G}; \mathsf{G}'$ is defined when the causal order it generates allows each participant to ascertain when its execution in G is completed and the one in G' can start.

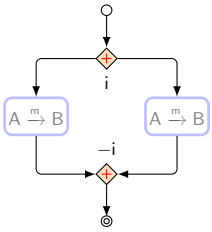
We now turn our attention to non-deterministic g-choreographies. The graph $\mathsf{G}_{(3a)}$ contains a sound choice: participant A decides which message (between x and y) is delivered to B. A further example of correct choreography is $\mathsf{G}_{(3b)}$. Here, participant A decides which branch is taken and informs either B or C, which in turn notifies the third participant. Intuitively, participant B can wait for a message coming from A or C and identify the selected branch accordingly. Let us first consider two correct non-deterministic g-choreographies that are only partially accommodated in the literature. The first interesting case is $\mathsf{G}_{(3c)}$. Intuitively, no difference can be observed between $\mathsf{G}_{(3c)}$ and $\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$, thus every participant complying with $\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$ is also complying with $\mathsf{G}_{(3c)}$. Another



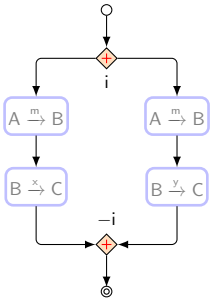
(a) $\mathsf{G}_{(3a)} = \mathsf{i}:(\underbrace{\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}}_{\mathsf{G}_1} + \underbrace{\mathsf{A} \xrightarrow{\mathsf{y}} \mathsf{B}}_{\mathsf{G}_2})$



(b) $\mathsf{G}_{(3b)} = \underbrace{\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{z}} \mathsf{C}}_{\mathsf{G}_1} + \underbrace{\mathsf{A} \xrightarrow{\mathsf{y}} \mathsf{C}; \mathsf{C} \xrightarrow{\mathsf{w}} \mathsf{B}}_{\mathsf{G}_2}$



(c) $\mathsf{G}_{(3c)} = \mathsf{i}:(\underbrace{\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}}_{\mathsf{G}_1} + \underbrace{\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}}_{\mathsf{G}_1})$

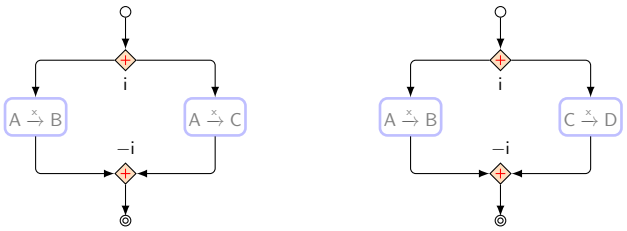


(d) $\mathsf{G}_{(3d)} = \underbrace{\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{x}} \mathsf{C}}_{\mathsf{G}_1} + \underbrace{\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{y}} \mathsf{C}}_{\mathsf{G}_2}$

Figure 3: Correct non-deterministic g-choreographies

example of correct choreography is the graph $\mathsf{G}_{(3d)}$, where the two branches have the same first interaction. Here A has the same behaviour in both G_1 and G_2, B decides which message is delivered to C, and C is informed about the choice taken by B via the reception of different messages. These simple cases are sometimes tackled in the literature by *merging* the branches of a choice. However, these type of (partial) operations have been defined in literature by constraining the parallelism in the branches of choices [7]. Note also that in our framework in $\mathsf{G}; \mathsf{G}'$ and in $\mathsf{G} + \mathsf{G}'$ both G and G' can be parallel choreographies, matching the level of generality of the choreographic languages in [6, 7] and in [24], however, our notion of well-branchedness is more general than the corresponding notions there.

The following examples discuss and anticipate some of the problems related to non-deterministic composition of g-choreographies. Choreography $\mathsf{G}_{(4a)}$ provides an example of unsound choices. Participant A decides to which participant the message x is delivered. The participant that is not selected by A has no way to identify if/when the choice has been made. This will force that participant to wait indefinitely for the message x. Another example of incorrect choreography



(a) $\mathsf{G}_{(4a)} = \mathsf{i}{:}(\underbrace{\mathsf{A} \xrightarrow{x} \mathsf{B}}_{\mathsf{G}_1} + \underbrace{\mathsf{A} \xrightarrow{x} \mathsf{C}}_{\mathsf{G}_2})$ (b) $\mathsf{G}_{(4b)} = \mathsf{i}{:}(\underbrace{\mathsf{A} \xrightarrow{x} \mathsf{B}}_{\mathsf{G}_1} + \underbrace{\mathsf{C} \xrightarrow{x} \mathsf{D}}_{\mathsf{G}_2})$

Figure 4: Incorrect non-deterministic choices

is $\mathsf{G}_{(4b)}$, since it requires both A and C to commit to a distributed choice without any communication among them.

The choreography $\mathsf{G}_{(5)}$ illustrates a complex choreography with nested choices. Here, the participant making all the choices is A. However, B and C receive different information about these choices. The behaviour of B is uniform in all branches of G_1 (where it always receives l from A) as well as in all those of G_2 (where it always receives r from A). The behaviour of C is more complex. The first message sent by A is either h or y_1. If h is received, C discovers that the left branch of either G_1 or G_2 has been selected: in both cases all the tasks of C in the choreography have been completed. If instead y_1 is received, C discovers that the right branch of either G_1 or G_2 has been selected: in all cases C waits for the reception of the message y_2 from A and uses the subsequent message (z_1, z_2 or z_3) to identify the choice made by A.

In Section 4.2 we introduce the notion of *well-branchedness* to identify non-deterministic composition of choreographies that are meaningful.

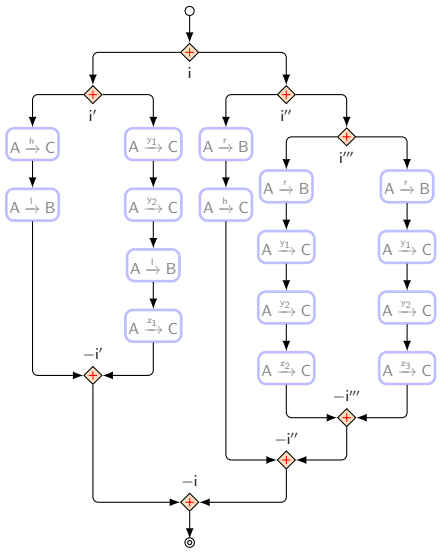
4. Pomset-based semantics of choreographies

The basic idea to provide a semantics to g-choreographies is to use collections of pomsets in order to account for the casual dependencies among communications that are introduced by different alternatives of choices.

4.1. Preliminaries: pomsets for global graphs

In our framework, interactions are based on actions that *happen on channels*, which we identify by the names of the participants involved in the communication. Formally, a channel is an element of the set $\mathcal{C} = \mathcal{P}^2 \setminus \{(\mathsf{A}, \mathsf{A}) \mid \mathsf{A} \in \mathcal{P}\}$ and we abbreviate $(\mathsf{A}, \mathsf{B}) \in \mathcal{C}$ as AB. The set of *labels* $\mathcal{L}$ is defined by

$$\mathcal{L} = \mathcal{L}^! \cup \mathcal{L}^? \qquad \text{where} \qquad \mathcal{L}^! = \mathcal{C} \times \{!\} \times \mathcal{M} \qquad \text{and} \qquad \mathcal{L}^? = \mathcal{C} \times \{?\} \times \mathcal{M}$$



$$\mathsf{G}_{(5)} = \mathsf{i}{:}(\mathsf{G}_1 + \mathsf{G}_2)$$

$$\mathsf{G}_1 = \mathsf{i}' : \left(\begin{array}{l} \mathsf{A} \xrightarrow{\mathsf{h}} \mathsf{C}; \mathsf{A} \xrightarrow{\mathsf{l}} \mathsf{B} \\ + \\ \mathsf{A} \xrightarrow{\mathsf{y}_1} \mathsf{C}; \mathsf{A} \xrightarrow{\mathsf{y}_2} \mathsf{C}; \\ \mathsf{A} \xrightarrow{\mathsf{l}} \mathsf{B}; \mathsf{A} \xrightarrow{\mathsf{z}_1} \mathsf{C} \end{array} \right)$$

$$\mathsf{G}_2 = \mathsf{i}'' : \left(\begin{array}{l} \mathsf{A} \xrightarrow{\mathsf{r}} \mathsf{B}; \mathsf{A} \xrightarrow{\mathsf{h}} \mathsf{C} \\ + \\ \mathsf{i}''' : (\mathsf{G}_{2a} + \mathsf{G}_{2b}) \end{array} \right)$$

$$\mathsf{G}_{2a} = \begin{array}{l} \mathsf{A} \xrightarrow{\mathsf{r}} \mathsf{B}; \mathsf{A} \xrightarrow{\mathsf{y}_1} \mathsf{C}; \\ \mathsf{A} \xrightarrow{\mathsf{y}_2} \mathsf{C}; \mathsf{A} \xrightarrow{\mathsf{z}_2} \mathsf{C} \end{array}$$

$$\mathsf{G}_{2b} = \begin{array}{l} \mathsf{A} \xrightarrow{\mathsf{r}} \mathsf{B}; \mathsf{A} \xrightarrow{\mathsf{y}_1} \mathsf{C}; \\ \mathsf{A} \xrightarrow{\mathsf{y}_2} \mathsf{C}; \mathsf{A} \xrightarrow{\mathsf{z}_2} \mathsf{C} \end{array}$$

Figure 5: A complex choreography

The elements of $\mathcal{L}^!$ and $\mathcal{L}^?$, outputs and inputs, respectively represent *sending* and *receiving* actions; we shorten $(\mathsf{AB}, !, \mathsf{m})$ as $\mathsf{AB!m}$ and $(\mathsf{AB}, ?, \mathsf{m})$ as $\mathsf{AB?m}$ and let l, l', ... range over $\mathcal{L}$. The *subject* of an action is defined by

$$\mathsf{sbj}(\mathsf{AB!m}) = \mathsf{A} \quad \text{(the sender)} \qquad \text{and} \qquad \mathsf{sbj}(\mathsf{AB?m}) = \mathsf{B} \quad \text{(the receiver)}$$

In the following, we write $l \in \mathsf{G}$ when there is an interaction $\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$ in the g-choreography G such that $l \in \{\mathsf{AB!m}, \mathsf{AB?m}\}$, and accordingly $\tilde{l} \subseteq \mathsf{G}$ means that $l \in \mathsf{G}$ for all $l \in \tilde{l}$.

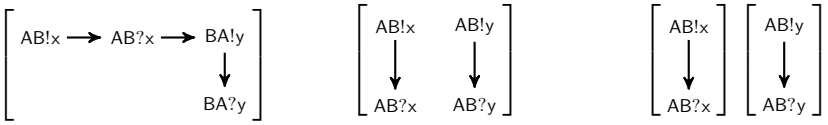
We reuse the formalisation of partially-ordered multi-set of [16] later used by [22] to give semantics to message-sequence charts.

Definition 1. *A labelled partially-ordered set (lposet) is a triple $(\mathcal{E}, \leq, \lambda)$, with $\mathcal{E}$ a set of events, $\leq \subseteq \mathcal{E} \times \mathcal{E}$ a reflexive, anti-symmetric, and transitive relation on $\mathcal{E}$, and $\lambda : \mathcal{E} \to \mathcal{L}$ a labelling function.*

For $e \neq e'$, $\lambda(e) = \lambda(e')$ means that e and e' model different occurrences of the same action. Intuitively, $\leq$ represents causality; for $e \neq e'$, if $e \leq e'$ and both events occur then e' is caused by e. In the following, we use $\boldsymbol{\varepsilon}$ to denote the empty lposet and $e \to e'$ to denote that $e \leq e'$.

Definition 2. *Two lposets $(\mathcal{E}, \leq, \lambda)$ and $(\mathcal{E}', \leq', \lambda')$ are isomorphic iff there exists a bijection $\phi : \mathcal{E} \to \mathcal{E}'$ such that $e \leq e'$ iff $\phi(e) \leq' \phi(e')$ and $\lambda = \lambda' \circ \phi$.*

Definition 3. *A partially-ordered multi-set (of actions), pomset for short, is an isomorphism class of lposets.*



(a) $r_{(6a)}$ (b) $r_{(6b)}$ (c) $\mathcal{R}_{(6c)}$

Figure 6: Examples of pomsets

The main benefit of using pomset in place of lposets is that we can abstract from the mathematical structure used to define the events $\mathcal{E}$, freeing us from introducing canonical representations of lposets. In the following, $[\mathcal{E}, \leq, \lambda]$ denotes the isomorphism class of $(\mathcal{E}, \leq, \lambda)$, we let $r, r', \ldots$ (resp. $\mathcal{R}, \mathcal{R}', \ldots$) range over pomsets (resp. sets of pomsets), and we assume that any r contains at least one lposet which will possibly be referred to as $(\mathcal{E}_r, \leq_r, \lambda_r)$. Examples of pomsets are depicted in Fig. 6, where events are not explicitly shown as their labels univocally determine them. Intuitively, $r_{(6a)}$ establishes a total causal order from the left-most to the bottom right-most event; $r_{(6b)}$ represents the pomset of the parallel execution of two threads. In Fig. 6, $\mathcal{R}_{(6c)}$ represents a set of two pomsets, where each pomset defines the causal order of events that corresponds to a choice made by the participant A; in one case A sends message x, in the other case A sends message y.

In the following, given a natural number n, $\mathbf{n}$ represents the singleton $\{n\}$. Also, we use $X \uplus Y$ to represent the disjoint union of two sets X and Y: $X \uplus Y = (X \times \mathbf{1}) \cup (Y \times \mathbf{2})$. Finally, given a function f on X, we define $f \otimes \mathbf{n} = \{(x, n) \mapsto f(x) \mid x \in X\}$ as the function extending f to $X \times \mathbf{n}$; analogously, for a relation $R \subseteq X \times Y$, we let $R \otimes \mathbf{n} = \{((x, n), (y, n)) \mid (x, y) \in R\}$ be the relation extending R to $(X \times \mathbf{n}) \times (Y \times \mathbf{n})$.

We now define two important constructions to compose pomsets in parallel and sequentially.

Definition 4. *Let $r = [\mathcal{E}, \leq, \lambda]$ and $r' = [\mathcal{E}', \leq', \lambda']$ be two pomsets. The* parallel composition of r and r' *is:*

$$\mathsf{par}(r, r') \quad = \quad [\mathcal{E} \uplus \mathcal{E}',\ (\leq \otimes\ \mathbf{1}) \cup (\leq' \otimes\ \mathbf{2}),\ (\lambda \otimes\ \mathbf{1}) \cup (\lambda' \otimes\ \mathbf{2})]$$

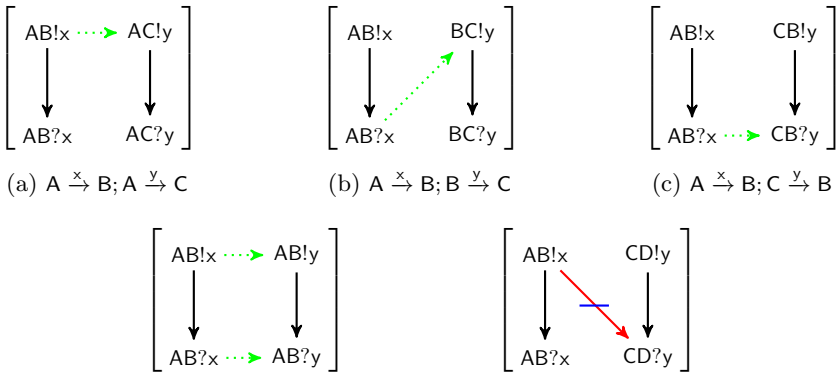
For a pomset r and a participant $\mathsf{A} \in \mathcal{P}$, let $\mathcal{E}_{r,\mathsf{A}} = \{e \in \mathcal{E}_r \mid \mathit{sbj}(\lambda_r(e)) = \mathsf{A}\}$ be the set of events of A in $\mathcal{E}_r$. The sequential composition of r and r' *is:*

$$\mathsf{seq}(r, r') \quad = \quad [\mathcal{E} \uplus \mathcal{E}', \leq_{seq}, (\lambda \otimes\ \mathbf{1}) \cup (\lambda' \otimes\ \mathbf{2})]$$

where

$$\leq_{seq} \quad = \quad \left((\leq \otimes\ \mathbf{1}) \cup (\leq' \otimes\ \mathbf{2}) \cup \bigcup_{\mathsf{A} \in \mathcal{P}} ((\mathcal{E}_{r,\mathsf{A}} \times \mathbf{1}) \times (\mathcal{E}_{r',\mathsf{A}} \times \mathbf{2})) \right)^{\star}$$

and $\star$ is the reflexive-transitive closure.



(a) $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{A} \xrightarrow{\mathsf{y}} \mathsf{C}$ (b) $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{y}} \mathsf{C}$ (c) $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{C} \xrightarrow{\mathsf{y}} \mathsf{B}$

(d) $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{A} \xrightarrow{\mathsf{y}} \mathsf{B}$ (e) $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{C} \xrightarrow{\mathsf{y}} \mathsf{D}$

Figure 7: Examples of sequential composition

Both parallel and sequential composition preserve the causal dependencies of its constituents $\leq$ and $\leq'$. However, while there is no new dependency introduced by the parallel composition, the sequential composition of two pomsets adds to those in $\leq$ and $\leq'$ the dependencies among events in r and r' with the same subject. Basically, a causal relation is induced whenever a participant performing a communication in r also performs a communication in r'. Fig. 7 depicts the sequential compositions of two pomsets, say r and r'. The former pomset corresponds to the interaction $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}$, while the second ranges over the interactions

$$\mathsf{A} \xrightarrow{\mathsf{y}} \mathsf{C}, \qquad \mathsf{B} \xrightarrow{\mathsf{y}} \mathsf{C}, \qquad \mathsf{C} \xrightarrow{\mathsf{y}} \mathsf{B}, \qquad \mathsf{A} \xrightarrow{\mathsf{y}} \mathsf{B}, \qquad \text{and} \qquad \mathsf{C} \xrightarrow{\mathsf{y}} \mathsf{D}$$

Simple arrows represent the dependencies induced by the subjects and dotted arrows represent dependencies induced by the sequential composition (the meaning of stroken arrows will be explained in Section 4.2).

4.2. Semantics of choreographies

The semantics of a choice-free g-choreography $\mathsf{G} \in \mathcal{G}$ (i.e. a choreography that does not contain $\_ + \_$ terms) can be expressed using a partial order, which represents the causal dependencies of the communication actions specified by G. Choices are a bit more tricky. Intuitively, the semantics of $\mathsf{G} + \mathsf{G}'$ consists of two (sets of) partial orders, one representing the causal dependencies of the communication actions of G and the other of those of G'. Therefore, the semantics

of a g-choreography is a family of pomsets defined as

$$[\![\mathbf{0}]\!] = \{\varepsilon\}$$

$$[\![\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}]\!] = \{[\big(\{e_1, e_2\}, \{(e_1, e_1), (e_2, e_2), (e_1, e_2)\}, \lambda\big)]\} \text{ where } \lambda : \begin{cases} e_1 \mapsto \mathsf{AB!m} \\ e_2 \mapsto \mathsf{AB?m} \end{cases}$$

$$[\![\mathsf{G}|\mathsf{G}']\!] = \begin{cases} \{\mathsf{par}(r, r') \mid (r, r') \in [\![\mathsf{G}]\!] \times [\![\mathsf{G}']\!]\} & \text{if } wf(\mathsf{G}, \mathsf{G}') \\ \bot & \text{otherwise} \end{cases}$$

$$[\![\mathsf{G}; \mathsf{G}']\!] = \begin{cases} \{\mathsf{seq}(r, r') \mid (r, r') \in [\![\mathsf{G}]\!] \times [\![\mathsf{G}']\!]\} & \text{if } ws(\mathsf{G}, \mathsf{G}') \\ \bot & \text{otherwise} \end{cases}$$

$$[\![\mathsf{G} + \mathsf{G}']\!] = \begin{cases} [\![\mathsf{G}]\!] \cup [\![\mathsf{G}']\!] & \text{if } wb(\mathsf{G}, \mathsf{G}') \\ \bot & \text{otherwise} \end{cases}$$

where $wf(\mathsf{G}, \mathsf{G}')$, $ws(\mathsf{G}, \mathsf{G}')$, and $wb(\mathsf{G}, \mathsf{G}')$ in the last three clauses check that parallel, sequential, and non-deterministic composition of choreographies are meaningful. Before defining those predicates we discuss the other cases. The semantics of the empty g-choreography $\mathbf{0}$ and of interaction $\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$ are straightforward; for the latter, the send part $\mathsf{AB!m}$ of the interaction must precede its receive part $\mathsf{AB?m}$.

For the parallel composition $\mathsf{G}|\mathsf{G}'$ we take the union of the dependencies of every possible execution, thus allowing the concurrent occurrence of the events of each thread. We also require that input events of G and G' are disjoint. Formally,

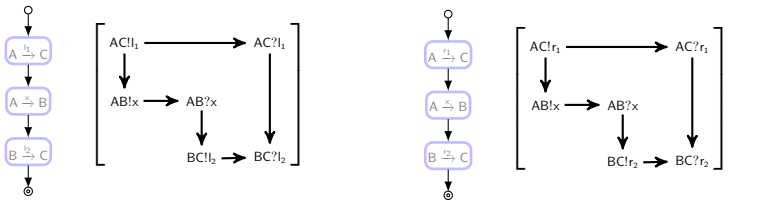
Definition 5. *Pomsets* $r = [\mathcal{E}, \leq, \lambda]$ *and* $r' = [\mathcal{E}', \leq', \lambda']$ *are* well-forked *if*

$$\lambda(\mathcal{E}) \cap \lambda'(\mathcal{E}') \cap \mathcal{L}^? = \emptyset$$

We write $wf(r, r')$ *when* r *and* r' *are well-forked and, for* $\mathsf{G}, \mathsf{G}' \in \mathcal{G}$, $wf(\mathsf{G}, \mathsf{G}')$ *when* $[\![\mathsf{G}]\!] \neq \bot \wedge [\![\mathsf{G}']\!] \neq \bot \wedge \forall r \in [\![\mathsf{G}]\!], r' \in [\![\mathsf{G}']\!] : wf(r, r')$

Parallel composition of the pomsets of G_1 and G_2 preserves the order relations of parallel threads. Additionally, well-forkedness ensures that the actions corresponding to the events in one thread cannot be confused with those in other threads. When well-forkedness does not hold, the pomsets induced by G_1 and G_2 would yield an order on shared inputs that would be too strict. This is illustrated by considering the the choreography $\mathsf{G}_{(8)} = \mathsf{G}_{(8a)}|\mathsf{G}_{(8b)}$, where $\mathsf{G}_{(8a)}$ and $\mathsf{G}_{(8b)}$ are given in Fig. 8 together with their semantics. In the parallel composition, the interaction $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}$ is shared between the two threads. The semantics of $\mathsf{G}_{(8)}$ should be the parallel composition of $[\![\mathsf{G}_{(8a)}]\!]$ and $[\![\mathsf{G}_{(8b)}]\!]$, without any further dependencies among the events of the two pomsets. Notice that, independently of the interleaving of the constituent threads, the event $\mathsf{AC!l}_1$ must always precede the event $\mathsf{BC!r}_2$. However, this property cannot be enforced because an implementation of $\mathsf{G}_{(8)}$ can proceed as follows

1. the left thread of A executes $\mathsf{AC!l}_1$ and $\mathsf{AB!x}$



(a) $\mathsf{G}_{(8a)} = \mathsf{A} \xrightarrow{\mathsf{l}_1} \mathsf{C}; \mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{l}_2} \mathsf{C}$ (b) $\mathsf{G}_{(8b)} = \mathsf{A} \xrightarrow{\mathsf{r}_1} \mathsf{C}; \mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{r}_2} \mathsf{C}$

Figure 8: A non well-forked choreography

2. the right thread of B executes $\mathsf{AB?x}$, "stealing" the message x generated by the left thread of A and meant for the left thread of B
3. the right thread of B executes $\mathsf{BC!r_2}$.

Informally, the participant B cannot ascertain if the reception of $\mathsf{AB?m}$ belongs to the first or second thread, therefore he cannot decide which message between $\mathsf{l_2}$ and $\mathsf{r_2}$ should be delivered.

The semantics of sequential composition $\mathsf{G}; \mathsf{G}'$ establishes happens-before relations as computed by $\mathsf{seq}(r, r')$ provided that they cover the dependencies between the output events of G and the input events of G'. Formally,

Definition 6. *Pomsets* $r = [\mathcal{E}, \leq, \lambda]$ *and* $r' = [\mathcal{E}', \leq', \lambda']$ *are* well-sequenced *if*

$$\leq_{\mathsf{seq}(r,r')} \quad \supseteq \quad (\{e \in \mathcal{E} \mid \lambda(e) \in \mathcal{L}^!\} \times \mathbf{1}) \times (\{e \in \mathcal{E}' \mid \lambda'(e) \in \mathcal{L}^?\} \times \mathbf{2})$$

We write $ws(r, r')$ *when* r *and* r' *are well-sequenced and, for* $\mathsf{G}, \mathsf{G}' \in \mathcal{G}$, $ws(\mathsf{G}, \mathsf{G}')$ *when* $[\![\mathsf{G}]\!] \neq \bot \wedge [\![\mathsf{G}']\!] \neq \bot \wedge \forall r \in [\![\mathsf{G}]\!], r' \in [\![\mathsf{G}']\!] : ws(r, r')$.

Well-sequencedness ensures the soundness of sequential composition; when it does not hold, there is a participant A in G' that cannot ascertain when to start its execution in G'. All examples in Fig. 7 are sound, bar the one in Fig. 7e, where the stroken edge depicts the missing causal dependency.

The semantics of a choice $\mathsf{G} + \mathsf{G}'$ is defined provided that the *well-branched* condition holds. The definition of well-branchedness relies on the notions of *active* and *passive* participants (defined below) that single out those participants that (internally) select which branch to execute and those participants that have to "understand" what choice was made according to the messages they receive, instead of internally choosing a branch.

Definition 7. *Two g-choreographies* G *and* G' *are* well-branched *when*

(i) there is at most one active *participant in* $\mathsf{G} + \mathsf{G}'$ *and*

(ii) all the other participants of $\mathsf{G} + \mathsf{G}'$ *are* passive.

We write $wb(\mathsf{G}, \mathsf{G}')$ *to denote that* G *and* G' *are well-branched.*

As said earlier, in the semantics of $\mathsf{G} + \mathsf{G}'$ no additional dependencies are introduced other than the ones induced by either G or G'. In fact, either the actions of the first branch or the actions of the second one will be performed. The behaviour of two branches can be the same up to a *point of divergence* where some participants start to behave differently on the two branches. It is at points of divergence where we discriminate if a participant is active or passive.

As usual, $\_|_X$ denotes the restriction of a function to a subset X of its domain. In order to identify a point of divergence we need to consider the part of a pomset involving a participant. In the following, we fix a participant $\mathsf{A} \in \mathcal{P}$.

Definition 8. *Let $r = [\mathcal{E}, \leq, \lambda]$ be a pomset. The pomset*

$$r\lfloor_{\mathsf{A}} \quad = \quad [\mathcal{E}_{r,\mathsf{A}},\ \leq \cap\ (\mathcal{E}_{r,\mathsf{A}} \times \mathcal{E}_{r,\mathsf{A}}),\ \lambda|_{\mathcal{E}_{r,\mathsf{A}}}]$$

is the pomset of A in r*; $\_\lfloor_{\mathsf{A}}$ extends to sets of pomsets element-wise.*

A way of defining points of divergence of two branches is in terms of the events that differ in the two branches after "a common prefix", where a prefix of a pomset r is a pomset r' on a subset of the events of r that preserves the order and labelling of r; formally (following [22])

Definition 9. *A pomset $r' = [\mathcal{E}', \leq', \lambda']$ is a* sub-pomset *of pomset $r = [\mathcal{E}, \leq, \lambda]$ if*

$$\mathcal{E}' \subseteq \mathcal{E} \quad \textit{and} \quad \leq' = \leq \cap (\mathcal{E}' \times \mathcal{E}') \quad \textit{and} \quad \lambda' = \lambda|_{\mathcal{E}'}$$

A sub-pomset r' of r is a prefix *of r if $\leq \cap((\mathcal{E} \setminus \mathcal{E}') \times \mathcal{E}') = \emptyset$.*

Definition 10. *The* suffix *of a pomset r with respect to one of its prefixes r', denoted as $r - r'$, is the pomset $[\mathcal{E}',\ \leq_r \cap(\mathcal{E}' \times \mathcal{E}'),\ \lambda_r|_{\mathcal{E}'}]$, where $\mathcal{E}' = \mathcal{E}_r \setminus \mathcal{E}_{r'}$.*

The semantics of each branch in a choice $\mathsf{G}_1 + \mathsf{G}_2$ is, in general, a set of pomsets. Our definition relies on finding a "common part" of the branches for which participants behave uniformly in both branches G_1 and G_2 of the choice. To identify such common part we consider the behaviour of each participant A on the branches of the choice "in isolation"; namely, we analyse $[\![\mathsf{G}_1]\!]\lfloor_{\mathsf{A}}$ and $[\![\mathsf{G}_2]\!]\lfloor_{\mathsf{A}}$ that is the events on the branches that involve A. A complication in finding common prefixes is due to the fact that a pomset in $[\![\mathsf{G}_1]\!]\lfloor_{\mathsf{A}}$ could be "matched" by several pomsets in $[\![\mathsf{G}_2]\!]\lfloor_{\mathsf{A}}$. To address this problem, we need to identify how the pomsets in $[\![\mathsf{G}_1]\!]\lfloor_{\mathsf{A}}$ correspond to those in $[\![\mathsf{G}_2]\!]\lfloor_{\mathsf{A}}$. For instance, for the participant C in the example $\mathsf{G}_{(5)}$ (page 12), we need to relate the right branch of i' to both branches of i''' because of the common prefix consisting of the reception of y_1 followed by the reception of y_2.

Definition 11. *The pair of functions (ϕ, ψ) is an* A-prefix map of G_1 and G_2 *if*

- $\operatorname{dom}\phi$ *and* $\operatorname{cod}\phi$ *are partitions*[9] *of $[\![\mathsf{G}_1]\!]\lfloor_{\mathsf{A}}$ and $[\![\mathsf{G}_2]\!]\lfloor_{\mathsf{A}}$ respectively, ϕ is bijective, and*

[9] Namely, families of disjoint subsets covering $[\![\mathsf{G}_1]\!]\lfloor_{\mathsf{A}}$ and $[\![\mathsf{G}_2]\!]\lfloor_{\mathsf{A}}$, respectively.

- $\operatorname{dom}\psi = \operatorname{dom}\phi$ *and* $\operatorname{cod}\psi$ *is a set of pomsets*

such that, for all $\mathcal{R} \in \operatorname{dom}\phi$ *and* $(r, r') \in \mathcal{R} \times \phi(\mathcal{R})$*, the pomset* $\psi(\mathcal{R})$ *is a prefix pomset of both* r *and* r'*.*

Definition 12. *Let* $\tilde{l}_1$ *and* $\tilde{l}_2$ *be two subsets of* $\mathcal{L}$*, we say that* $(\tilde{l}_1, \tilde{l}_2)$ is the divergence point of G_1 and G_2 (*or that* G_1 and G_2 diverge at $(\tilde{l}_1, \tilde{l}_2)$) with respect to the A-prefix map (ϕ, ψ)*, denoted* $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\tilde{l}_1, \tilde{l}_2)$ *if*

$$\tilde{l}_1 = \bigcup_{\substack{\mathcal{R} \in \operatorname{dom}\phi, \\ r \in \mathcal{R}}} \lambda_r\big(\texttt{min}\ \big(r - \psi(\mathcal{R})\big)\big) \quad \textit{and} \quad \tilde{l}_2 = \bigcup_{\substack{\mathcal{R} \in \operatorname{cod}\phi, \\ r \in \mathcal{R}}} \lambda_r\big(\texttt{min}\ \big(r - \psi(\phi^{-1}(\mathcal{R}))\big)\big)$$

where, for a pomset r, $\texttt{min}\ r = \{e \in \mathcal{E}_r \mid \nexists e' \in \mathcal{E}_r : e' \neq e \wedge e' \leq_r e\}$.

The bijection ϕ "matches" sets of executions (pomsets) in $[\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{A}}$ with those in $[\![\mathsf{G}_2]\!]\!\downarrow_{\mathsf{A}}$ that have an initial common behaviour determined by the common prefix identified by the mapping ψ. Thus, $\tilde{l}_1$ (resp. $\tilde{l}_2$) is the union of the labels at the point of divergence for $[\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{A}}$ (resp. $[\![\mathsf{G}_2]\!]\!\downarrow_{\mathsf{A}}$), i.e. the union of the labels of the minimal events of the pomsets in $[\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{A}}$ (and $[\![\mathsf{G}_2]\!]\!\downarrow_{\mathsf{A}}$) after removing the common prefixes. We use the examples of Sections 3 and 4 to demonstrate the notion of divergence points.

Example $\mathsf{G}_{(3a)}$. For participant A we have $\mathcal{R}_1 = [\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{A}} = \{[\mathsf{AB!x}]\}$ and $\mathcal{R}_2 = [\![\mathsf{G}_2]\!]\!\downarrow_{\mathsf{A}} = \{[\mathsf{AB!y}]\}$. Since $\mathcal{R}_1$ and $\mathcal{R}_2$ are singletons, each of them has exactly one partition. Let $\phi = \mathcal{R}_1 \mapsto \mathcal{R}_2$ and $\psi = \mathcal{R}_1 \mapsto \boldsymbol{\varepsilon}$ (i.e. the empty pomset, which is the only common prefix) then $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\{\mathsf{AB!x}\}, \{\mathsf{AB!y}\})$. For participant B, $\mathcal{R}_1 = [\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{B}} = \{[\mathsf{AB?x}]\}$ and $\mathcal{R}_2 = [\![\mathsf{G}_2]\!]\!\downarrow_{\mathsf{B}} = \{[\mathsf{AB?y}]\}$. Let $\phi = \mathcal{R}_1 \mapsto \mathcal{R}_2$ and $\psi = \mathcal{R}_1 \mapsto \boldsymbol{\varepsilon}$ then $\mathsf{div}_{\mathsf{B}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\{\mathsf{AB?x}\}, \{\mathsf{AB?y}\})$.

Example $\mathsf{G}_{(3b)}$. Similarly to the previous example, participant A diverges at $(\{\mathsf{AB!x}\}, \{\mathsf{AC!y}\})$. For participant B, $\mathcal{R}_1 = [\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{B}} = \{[\mathsf{AB?x} \to \mathsf{BC!z}]\}$ and $\mathcal{R}_2 = [\![\mathsf{G}_2]\!]\!\downarrow_{\mathsf{B}} = \{[\mathsf{CB?w}]\}$, with $\phi = \mathcal{R}_1 \mapsto \mathcal{R}_2$ and $\psi = \mathcal{R}_1 \mapsto \boldsymbol{\varepsilon}$ we obtain $\mathsf{div}_{\mathsf{B}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\{\mathsf{AB?x}\}, \{\mathsf{CB?w}\})$. For participant C, $\mathcal{R}_1 = [\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{C}} = \{[\mathsf{BC?z}]\}$ and $\mathcal{R}_2 = [\![\mathsf{G}_2]\!]\!\downarrow_{\mathsf{C}} = \{[\mathsf{AC?y} \to \mathsf{CB!w}]\}$. Let $\phi = \mathcal{R}_1 \mapsto \mathcal{R}_2$ and $\psi = \mathcal{R}_1 \mapsto \boldsymbol{\varepsilon}$ then $\mathsf{div}_{\mathsf{B}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\{\mathsf{BC?z}\}, \{\mathsf{AC?y}\})$.

Example $\mathsf{G}_{(3c)}$. For participant A, we have two cases depending on which prefix map we consider. Note that $\mathcal{R}_1 = \mathcal{R}_2 = [\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{A}} = [\![\mathsf{G}_2]\!]\!\downarrow_{\mathsf{A}} = \{r\}$, where $r = [\mathsf{AB!m}]$, and let $\phi = \mathcal{R}_1 \mapsto \mathcal{R}_2$. If we choose the empty prefix (that is $\psi = \mathcal{R}_1 \mapsto \boldsymbol{\varepsilon}$) the point of divergence can be computed as $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\{\mathsf{AB!m}\}, \{\mathsf{AB!m}\})$. A different point of divergence can be computed using the prefix r: let $\psi = \mathcal{R}_1 \mapsto r$ then $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\emptyset, \emptyset)$. Similarly, for the participant B, $\mathcal{R}_1 = \mathcal{R}_2 = [\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{B}} = [\![\mathsf{G}_2]\!]\!\downarrow_{\mathsf{B}} = \{r\}$, where $r = [\mathsf{AB?m}]$. According with the prefix chosen, two different points of divergence can be computed. For the non-empty prefix: let $\phi = \mathcal{R}_1 \mapsto \mathcal{R}_2$ and $\psi = \mathcal{R}_1 \mapsto r$ then $\mathsf{div}_{\mathsf{B}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\emptyset, \emptyset)$.

Example $\mathsf{G}_{(3d)}$. Here the two branches have the same first interaction. For participant A, we have $\mathcal{R}_1 = \mathcal{R}_2 = [\![\mathsf{G}_1]\!]\restriction_{\mathsf{A}} = [\![\mathsf{G}_2]\!]\restriction_{\mathsf{A}} = \{[\mathsf{AB!m}]\}$; hence $\phi = \mathcal{R}_1 \mapsto \mathcal{R}_2$ is the unique candidate function for ϕ while either $\psi = \mathcal{R}_1 \mapsto \varepsilon$ or $\psi = \mathcal{R}_1 \mapsto [\mathsf{AB!m}]$; choosing the latter, we obtain that $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\emptyset, \emptyset)$. For participant C, $\mathcal{R}_1 = [\![\mathsf{G}_1]\!]\restriction_{\mathsf{C}} = \{[\mathsf{BC?x}]\}$ and $\mathcal{R}_2 = [\![\mathsf{G}_2]\!]\restriction_{\mathsf{C}} = \{[\mathsf{BC?y}]\}$. Let $\phi = \mathcal{R}_1 \mapsto \mathcal{R}_2$ and $\psi = \mathcal{R}_1 \mapsto \varepsilon$ (i.e. the only possible choice, since there is no common prefix) then $\mathsf{div}_{\mathsf{C}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\{\mathsf{BC?x}\}, \{\mathsf{BC?y}\})$. Finally, for participant B, $\mathcal{R}_1 = [\![\mathsf{G}_1]\!]\restriction_{\mathsf{B}} = \{[\mathsf{AB?m} \mapsto \mathsf{BC!x}]\}$ and $\mathcal{R}_2 = [\![\mathsf{G}_1]\!]\restriction_{\mathsf{B}} = \{[\mathsf{AB?m} \mapsto \mathsf{BC!y}]\}$. Let $\phi = \mathcal{R}_1 \mapsto \mathcal{R}_2$ and $\psi = \mathcal{R}_1 \mapsto [\mathsf{AB?m}]$ then $\mathsf{div}_{\mathsf{B}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\{\mathsf{BC!x}\}, \{\mathsf{BC!y}\})$.

Example $\mathsf{G}_{(4a)}$. Participant A sends either $\mathsf{AB!x}$ or $\mathsf{AC!y}$ and the point of divergence can be computed similarly to the case $\mathsf{G}_{(3b)}$. The divergence points for the other participants are more interesting. We consider participant B only as the case for C is similar. We have $\mathcal{R}_1 = [\![\mathsf{G}_1]\!]\restriction_{\mathsf{B}} = \{[\mathsf{AB?x}]\}$ and $\mathcal{R}_2 = [\![\mathsf{G}_2]\!]\restriction_{\mathsf{B}} = \{\varepsilon\}$. The only B-prefix map candidate is (ϕ, ψ) with $\phi = \mathcal{R}_1 \mapsto \mathcal{R}_2$ and $\psi = \mathcal{R}_1 \mapsto \varepsilon$, hence $\mathsf{div}_{\mathsf{B}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\{\mathsf{AB?x}\}, \emptyset)$. This reflects the fact that in one branch B receives notification of the choice made by A, while in the other branch no information is received.

Example $\mathsf{G}_{(4b)}$. Here, it is easy to verify that participant A diverges at $(\{\mathsf{AB!x}\}, \emptyset)$, B at $(\{\mathsf{AB?x}\}, \emptyset)$, C at $(\emptyset, \{\mathsf{CD!x}\})$, and D at $(\emptyset, \{\mathsf{CD?x}\})$.

Example $\mathsf{G}_{(5)}$. For participant C, the numbers of pomsets in the semantics of the two branches at control point i differ:

$$\mathcal{R}_1 = \{[\mathsf{AC?h}], [\mathsf{AC?y_1} \to \mathsf{AC?y_2} \to \mathsf{AC?z_1}]\}$$
$$\mathcal{R}_2 = \{[\mathsf{AC?h}], [\mathsf{AC?y_1} \to \mathsf{AC?y_2} \to \mathsf{AC?z_2}], [\mathsf{AC?y_1} \to \mathsf{AC?y_2} \to \mathsf{AC?z_3}]\}$$

The function ϕ can be chosen to

- map the subset $\mathcal{R}_{1a} = \{[\mathsf{AC?h}]\}$ of $\mathcal{R}_1$ to the subset $\{[\mathsf{AC?h}]\}$ of $\mathcal{R}_2$ and
- map the subset $\mathcal{R}_{1b} = \{[\mathsf{AC?y_1} \to \mathsf{AC?y_2} \to \mathsf{AC?z_1}]\}$ of $\mathcal{R}_1$ to the subset $\{[\mathsf{AC?y_1} \to \mathsf{AC?y_2} \to \mathsf{AC?z_2}], [\mathsf{AC?y_1} \to \mathsf{AC?y_2} \to \mathsf{AC?z_3}]\}$ of $\mathcal{R}_2$.

With $\psi = \begin{cases} \mathcal{R}_{1a} \mapsto [\mathsf{AC?h}] \\ \mathcal{R}_{1b} \mapsto [\mathsf{AC?y_1} \to \mathsf{AC?y_2}] \end{cases}$ we have that, for participant C, the divergence point is $\mathsf{div}_{\mathsf{C}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\{\mathsf{AC?z_1}\}, \{\mathsf{AC?z_2}, \mathsf{AC?z_3}\})$. With similar calculations, we can show that B diverges at $(\{\mathsf{AB?l}\}, \{\mathsf{AB?r}\})$ and that participant A diverges at $(\{\mathsf{AC!h}, \mathsf{AC!y_1}\}, \{\mathsf{AB!r}\})$. In passing, note that there are other possible choices of prefix maps for $\mathsf{G}_{(5)}$ not considered here.

We now formalise the concepts of active and passive participants.

Definition 13. *A participant* $\mathsf{A} \in \mathcal{P}$ *is* active *in* $\mathsf{G}_1 + \mathsf{G}_2$ *if there exists an* A*-prefix map* (ϕ, ψ) *of* G_1 *and* G_2 *such that, if* $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\tilde{l}_1, \tilde{l}_2)$*, then*

$$\tilde{l}_1 \cup \tilde{l}_2 \subseteq \mathcal{L}^! \qquad \tilde{l}_1 \cap \tilde{l}_2 = \emptyset \qquad \tilde{l}_1 \neq \emptyset \qquad \tilde{l}_2 \neq \emptyset$$

Thus, the behaviour of an active participant A in G_1 and G_2 must be the same up to the point where it informs the other participants, by sending different messages, which branch it chooses.

Definition 14. *A participant* $\mathsf{A} \in \mathcal{P}$ *is* passive *in* $\mathsf{G}_1 + \mathsf{G}_2$ *if there exists an* A*-prefix map* (ϕ, ψ) *of* G_1 *and* G_2 *such that, if* $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\tilde{l}_1, \tilde{l}_2)$*, then*

- $\tilde{l}_1 \cup \tilde{l}_2 \subseteq \mathcal{L}^?$ *and* $\tilde{l}_1 = \emptyset \iff \tilde{l}_2 = \emptyset$
- $\forall \mathcal{R} \in \operatorname{dom} \phi, r \in \mathcal{R} : \tilde{l}_2 \cap \lambda_r(\mathcal{E}_{r-\psi(\mathcal{R})}) = \emptyset$
- $\forall \mathcal{R} \in \operatorname{cod} \phi, r \in \mathcal{R} : \tilde{l}_1 \cap \lambda_r(\mathcal{E}_{r-\psi(\phi^{-1}(\mathcal{R}))}) = \emptyset$

Thus, the behaviour of a passive participant A in G_1 and G_2 must be the same up to a point where it receives either of two different messages, each one identifying which branch had been selected. Clearly, A cannot perform outputs at the points of branching.

Interestingly, if one always takes a mapping yielding the empty prefix in the determination of active and passive roles, the definitions above yield exactly the same notions used e.g., in [21, 3, 11].

Lemma 1. *Participant* A *cannot be both passive and active in* $\mathsf{G}_1 + \mathsf{G}_2$*.*

Proof sketch. The proof shows that if a participant is active (passive) for ϕ and ψ then for every ϕ' and ψ' the prefixes in the image of ψ' are prefixes of the pomsets in the codomain of ψ. Appendix A reports the details of the proof. □

We demonstrate the notion of well-branchedness on the examples of Section 3. When it exists, the *selector* of the choice is the active participant that determines the branch to execute. For instance, in the choreography $\mathsf{G}_{(3a)}$ participant A is the selector of the choice since it diverges at $(\{\mathsf{AB!x}\}, \{\mathsf{AB!y}\})$. Hence, choreography $\mathsf{G}_{(3a)}$ is well-branched because the other participant B diverges at $(\{\mathsf{AB?x}\}, \{\mathsf{AB?y}\})$ and is therefore passive. Likewise, choreography $\mathsf{G}_{(3b)}$ is well-branched since A diverges at $(\{\mathsf{AB!x}\}, \{\mathsf{AC!y}\})$, B diverges at $(\{\mathsf{AB?x}\}, \{\mathsf{CB?w}\})$, and C diverges at $(\{\mathsf{BC?z}\}, \{\mathsf{AC?y}\})$; thus A is active and the other participants are passive.

Unlike its correspondents in the rest of the literature, our notion of well-branchedness does not require the selector to exist. For instance, the choreography $\mathsf{G}_{(3c)}$ is well-branched even if it has no active participant. Both participants are passive, since for both of them is possible to find a prefix map that leads to divergence at $(\emptyset, \emptyset)$. Another example (usually discharged in the literature by imposing syntactic constraints) is the choreography $\mathsf{G}_{(3d)}$. Here, the two branches have the same first interaction. Choosing the prefix $[\mathsf{AB!m}]$ for the participant A makes it passive. The participant C diverges at $(\{\mathsf{BC?x}\}, \{\mathsf{BC?y}\})$, thus it is passive. Finally, choosing the prefix $\mathsf{AB?m}$ for participant B, makes its point of divergence to be $(\{\mathsf{BC!x}\}, \{\mathsf{BC!y}\})$. Thus B is active and the choice well-branched.

An example of non well-branched choreography is $\mathsf{G}_{(4a)}$: the participant A is active (sending either $\mathsf{AB!x}$ or $\mathsf{AC!y}$). However, B and C are neither passive nor

active. In fact, participant B diverges at $(\{\mathsf{AB?x}\}, \emptyset)$ (there is no other possible point of divergence for B). Another example of non well-branched choreography is $\mathsf{G}_{(4b)}$, since no participant is active or passive.

Choreography $\mathsf{G}_{(5)}$ illustrates that our notion of well-branchedness allows to specify complex distributed choices. This example clarifies the need of partitioning the semantics of branches to compute active and passive participants. As shown (page 19), even though the semantics yield different numbers of pomsets in the two branches for participant C, we can choose a C-prefix map (ϕ, ψ) such that $\mathsf{div}_{\mathsf{C}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\{\mathsf{AC?z}_1\}, \{\mathsf{AC?z}_2, \mathsf{AC?z}_3\})$; thus C is passive. Hence, $\mathsf{G}_{(5)}$ is well-branched since B is passive and that A is the unique active participant. The choreography $\mathsf{G}_{(5)}$ has two other interesting well-branched variants. One where the interactions in G_2 are replaced with

$$\mathsf{G}_2' = \ \mathsf{A} \xrightarrow{\mathsf{r}} \mathsf{B}; \mathsf{i}'' : (\mathsf{A} \xrightarrow{\mathsf{h}} \mathsf{C} + \mathsf{A} \xrightarrow{\mathsf{y_1}} \mathsf{C}; \mathsf{A} \xrightarrow{\mathsf{y_2}} \mathsf{C}; \mathsf{i}''' : (\mathsf{A} \xrightarrow{\mathsf{z_2}} \mathsf{C} + \mathsf{A} \xrightarrow{\mathsf{z_3}} \mathsf{C}))$$

In G_2' the interaction $\mathsf{i} \colon \mathsf{A} \xrightarrow{\mathsf{r}} \mathsf{B}$ is factorised before the choice in i''. The other variant is one where such interaction is executed in parallel to the others in G_{2a} or G_{2b}.

4.3. Languages of choreographies

The abstract semantics of a g-choreography is the set of partial orders among the events of the g-choreography. A more concrete semantics, akin to the usual trace-based semantics of global views, can be given by considering the *language* of a g-choreography. Informally, the language of a g-choreography $\mathsf{G} \in \mathcal{G}$ consists of the sequences of words over the alphabet consisting of the communication actions of the events in G that preserve the causal relations of $[\![\mathsf{G}]\!]$, provided that $[\![\mathsf{G}]\!]$ is defined. The *language* of $\mathsf{G} \in \mathcal{G}$ is the set $\mathbb{L}_{\mathsf{G}}$ defined by

$$\mathbb{L}_{\mathsf{G}} = \begin{cases} \bigcup_{r \in [\![\mathsf{G}]\!]} \mathbb{L}_r, & \text{if } [\![\mathsf{G}]\!] \text{ is defined} \\ \bot, & \text{otherwise} \end{cases}$$

where the language $\mathbb{L}_r$ of a pomset $r = [\mathcal{E}, \leq, \lambda]$ is the set of $\lambda(w)$ such that $w \in \mathcal{E}^\star$ and, for any indexes i and j between 0 and the length of w and $i \neq j$,

1. $w[i] \neq w[j]$
2. if $w[i] \leq w[j]$ then $i \leq j$
3. for every $e \in \mathcal{E}$, if $e \leq w[i]$ and $e \neq w[i]$ then there exists $h < i$ such that $w[h] = e$

Clause 1 states that events in the word are not repeated. Clause 2 states that words preserve the causal relations of events. Clause 3 requires that all the predecessors of an event which appears in the word must precede that event in that word. Notice that $\mathbb{L}_r$ is prefix-closed (hence $\mathbb{L}_{\mathsf{G}}$ is prefix-closed too).

Hereafter, we comment on the languages of the examples in Section 4. The language $\mathbb{L}_{\mathsf{G}_{(3a)}} = \{\epsilon, \mathsf{AB!}x, \mathsf{AB!}y, \mathsf{AB!}x.\mathsf{AB?}x, \mathsf{AB!}y.\mathsf{AB?}y\}$ is the prefix-closure

of $\{\mathsf{AB}!x.\mathsf{AB}?x, \mathsf{AB}!y.\mathsf{AB}?y\}$ (we write $\_.\_$ for concatenation of words). Similarly, the languages of $\mathbb{L}_{\mathsf{G}_{(3b)}}$, $\mathbb{L}_{\mathsf{G}_{(3c)}}$, and $\mathbb{L}_{\mathsf{G}_{(3d)}}$ respectively are the prefix-closure of

$$\{\mathsf{AB}!x.\mathsf{AB}?x.\mathsf{BC}!z.\mathsf{BC}?z, \mathsf{AC}!y.\mathsf{AC}?y.\mathsf{CB}!w.\mathsf{CB}?w\}$$
$$\{\mathsf{AB}!\mathsf{m}.\mathsf{AB}?\mathsf{m}\}$$
$$\{\mathsf{AB}!\mathsf{m}.\mathsf{AB}?\mathsf{m}.\mathsf{BC}!x.\mathsf{BC}?x, \mathsf{AB}!\mathsf{m}.\mathsf{AB}?\mathsf{m}.\mathsf{BC}!y.\mathsf{BC}?y\}$$

The languages $\mathbb{L}_{\mathsf{G}_{(4a)}}$ and $\mathbb{L}_{\mathsf{G}_{(4b)}}$ are undefined, since the choreographies are not well-branched. The language $\mathbb{L}_{\mathsf{G}_{(5)}}$ is bigger and defined in a similar way.

5. Projecting on communicating machines

As in [26, 15], we adopt *communicating finite state machines* (CFSM) as local artefacts. We borrow the definition of CFSMs in [5] adapting it to our context. A CFSM $M = (Q, q_0, \rightarrow)$ is a finite transition system where

- Q is a finite set of *states* with $q_0 \in Q$ the *initial* state, and
- $\rightarrow \subseteq\ Q \times \mathcal{L} \times Q$; we write $q \xrightarrow{l} q'$ for $(q, l, q') \in \rightarrow$.

A CFSM $M = (Q, q_0, \rightarrow)$ is A-*local* if $\mathsf{sbj}(l) = \mathsf{A}$ for each $q \xrightarrow{l} q'$. Given an A-local CFSM $M_\mathsf{A} = (Q_\mathsf{A}, q_{0\mathsf{A}}, \rightarrow_\mathsf{A})$ for each $\mathsf{A} \in \mathcal{P}$, the tuple $S = (M_\mathsf{A})_{\mathsf{A}\in\mathcal{P}}$ is a *(communicating) system*. For all $\mathsf{A} \neq \mathsf{B} \in \mathcal{P}$, it is assumed that there is an infinite FIFO queue b_{AB} where M_A puts the message to M_B and from which M_B consumes the messages from M_A.

The semantics of communicating systems is defined in terms of *transition systems*, which keep track of the state of each machine and the content of each queue. Let $S = (M_\mathsf{A})_{\mathsf{A}\in\mathcal{P}}$ be a communicating system. A *configuration* of S is a pair $s = \langle \tilde{q}\ ;\ \tilde{b}\rangle$ where $\tilde{q} = (q_\mathsf{A})_{\mathsf{A}\in\mathcal{P}}$ with $q_\mathsf{A} \in Q_\mathsf{A}$ and $\tilde{b} = (b_{\mathsf{AB}})_{\mathsf{AB}\in\mathcal{C}}$ with $b_{\mathsf{AB}} \in \mathcal{M}^*$; state q_A keeps track of the state of the machine M_A and buffer b_{AB} keeps track of the messages sent from A to B. The *initial* configuration s_0 is the one where, for all $\mathsf{A} \in \mathcal{P}$, q_A is the initial state of the corresponding CFSM and all buffers are empty.

A configuration $s' = \langle \tilde{q}'\ ;\ \tilde{b}'\rangle$ is *reachable* from another configuration $s = \langle \tilde{q}\ ;\ \tilde{b}\rangle$ by *firing a transition* l, written $s \overset{l}{\Rightarrow} s'$ if there is a message $\mathsf{m} \in \mathcal{M}$ such that either (1) or (2) below holds:

1. $l = \mathsf{AB}!\mathsf{m}$ and $q_\mathsf{A} \xrightarrow{l}_\mathsf{A} q'_\mathsf{A}$ and
   a. $q'_\mathsf{C} = q_\mathsf{C}$ for all $\mathsf{C} \neq \mathsf{A}$ and
   b. $b'_{\mathsf{AB}} = b_{\mathsf{AB}}.\mathsf{m}$ and
   c. $b'_{\mathsf{CD}} = b_{\mathsf{CD}}$ for all $(\mathsf{C},\mathsf{D}) \neq (\mathsf{A},\mathsf{B}) \in \mathcal{C}$
2. $l = \mathsf{AB}?\mathsf{m}$ and $q_\mathsf{B} \xrightarrow{l}_\mathsf{B} q'_\mathsf{B}$ and
   a. $q'_\mathsf{C} = q_\mathsf{C}$ for all $\mathsf{C} \neq \mathsf{B}$ and
   b. $b_{\mathsf{AB}} = \mathsf{m}.b'_{\mathsf{AB}}$ and
   c. $b'_{\mathsf{CD}} = b_{\mathsf{CD}}$ for all $(\mathsf{C},\mathsf{D}) \neq (\mathsf{A},\mathsf{B}) \in \mathcal{C}$

Condition (1) puts m on channel AB, while (2) gets m from channel AB. In both cases, any machine or buffer not involved in the transition is left unchanged in the new configuration s'.

A configuration $s = \langle \tilde{q} \ ; \ \tilde{b} \rangle$ is *stable* if all buffers are empty: $\tilde{b} = \tilde{\varepsilon}$. A configuration $s = \langle \tilde{q} \ ; \ \tilde{b} \rangle$ is a *deadlock* if $s \not\Rightarrow$ and

- there exists a participant $\mathsf{A} \in \mathcal{P}$ such that $q_{\mathsf{A}} \xrightarrow{\mathsf{AB?m}}_{\mathsf{A}} q'_{\mathsf{A}}$
- or $\tilde{b} \neq \tilde{\varepsilon}$

This definition is adapted from [10] and is meant to capture communications misbehaviour. Observe that, according to this definition, a configuration s where all machines are in a state with no outgoing transitions and all buffers are empty is not a deadlock configuration even though $s \not\Rightarrow$.

The language of a communicating system S is the set $\mathbb{L}_S \in \mathcal{L}^\star$ of sequences $l_0 \dots l_{n-1}$ such that $s_0 \stackrel{l_0}{\Longrightarrow} \dots \stackrel{l_{n-1}}{\Longrightarrow} s_n$. Note that $\mathbb{L}_S$ is prefix-closed.

Given two CFSMs $M = (Q, q_0, \rightarrow)$ and $M' = (Q', q'_0, \rightarrow')$, we use the following notations:

- $M \times M' = (Q \times Q', (q_0, q'_0), \rightarrow'')$ is the product of M and M' where $\big((q_1, q'_1), l, (q_2, q'_2)\big) \in \rightarrow''$ if, and only if,

$$\big((q_1, l, q_2) \in \rightarrow \text{ and } q'_1 = q'_2\big) \qquad \text{or} \qquad \big((q'_1, l, q'_2) \in \rightarrow' \text{ and } q_1 = q_2\big)$$

- $\left\{ {}^{q'}/_{q} \right\} M$ represents the machine obtained by substituting the state q with the state q', provided that q' is not in the states of M;
- $M \otimes \mathbf{n}$ represents the machine $(Q \times \mathbf{n}, (q_0, n), \rightarrow \otimes \mathbf{n})$;
- $M \circ M'$ represents the machine $(Q \cup Q', q_0, \rightarrow \cup \rightarrow')$;

To define the projection of a g-choreography G to CFSMs we provide a function defined by induction on the syntax of G that returns a triple (M, q_0, q_e) where M is a CFSM, q_0 is its initial state, and q_e is special state of M used to connect it to other machines. We will use $(M, q_0, q_e) \otimes \mathbf{n}$ to represent $(M \otimes \mathbf{n}, (q_0, n), (q_e, n))$. We will use a graphical notation to represent those triples where an arrow without target exiting a state singles out the second element of the pair (so, for (M, q_0, q_e) we will have a dangling arrow leaving q_e); also, an arrow without source entering a state of M singles out the initial state of the machine. Let G be a g-choreography, the function $\mathsf{G} \downarrow_{\mathsf{A}}$ yields the projection of G over the

participant A as follows:

$$
\mathsf{G}\downarrow_{\mathsf{A}} = \begin{cases}
\rightarrow (q_0) \rightarrow & \text{if } \mathsf{G} = \mathbf{0} \\
\rightarrow (q_0) \rightarrow & \text{if } \mathsf{G} = \mathsf{B} \xrightarrow{\mathsf{m}} \mathsf{C} \\
\rightarrow (q_0) \xrightarrow{\mathsf{AB}!m} (q_e) \rightarrow & \text{if } \mathsf{G} = \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}, \text{ with } q_0 \neq q_e \\
\rightarrow (q_0) \xrightarrow{\mathsf{BA}?m} (q_e) \rightarrow & \text{if } \mathsf{G} = \mathsf{B} \xrightarrow{\mathsf{m}} \mathsf{A}, \text{ with } q_0 \neq q_e \\
\left(M_1 \circ \left\{{}^{q_e^1}/_{q_0^2}\right\} M_2, q_0^1, q_e^2\right) & \text{if } \mathsf{G} = \mathsf{G}_1; \mathsf{G}_2 \\
& \text{and } (M_1, q_0^1, q_e^1) = \mathsf{G}_1 \downarrow_{\mathsf{A}} \otimes \mathbf{1} \\
& \text{and } (M_2, q_0^2, q_e^2) = \mathsf{G}_2 \downarrow_{\mathsf{A}} \otimes \mathbf{2} \\
\left(\left\{{}^{q_e^2}/_{q_e^1}\right\} M_1 \circ \left\{{}^{q_0^1}/_{q_0^2}\right\} M_2, q_0^1, q_e^2\right) & \text{if } \mathsf{G} = \mathsf{G}_1 + \mathsf{G}_2 \\
& \text{and } (M_1, q_0^1, q_e^1) = \mathsf{G}_1 \downarrow_{\mathsf{A}} \otimes \mathbf{1} \\
& \text{and } (M_2, q_0^2, q_e^2) = \mathsf{G}_2 \downarrow_{\mathsf{A}} \otimes \mathbf{2} \\
(M_1 \times M_2, (q_0^1, q_0^2), (q_e^1, q_e^2)) & \text{if } \mathsf{G} = \mathsf{G}_1 | \mathsf{G}_2, \\
& \text{and } (M_1, q_0^1, q_e^1) = \mathsf{G}_1 \downarrow_{\mathsf{A}} \otimes \mathbf{1} \\
& \text{and } (M_2, q_0^2, q_e^2) = \mathsf{G}_2 \downarrow_{\mathsf{A}} \otimes \mathbf{2}
\end{cases}
$$

Figure 9 shows examples of projections for participants A and B.

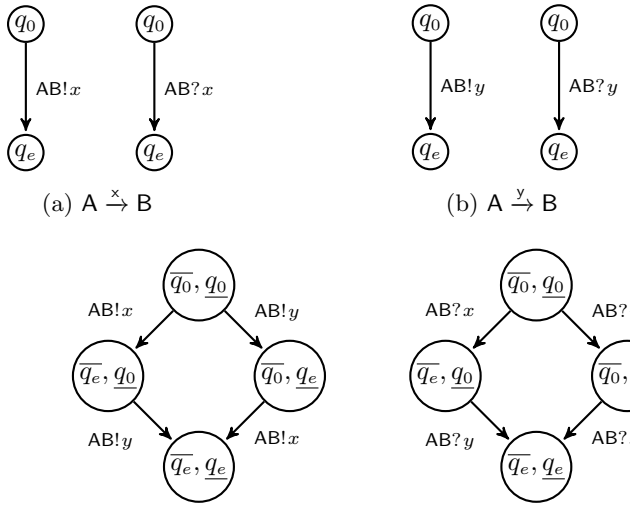


Figure 9: Examples of projections (we use $\overline{q}$ for $(q, 1)$ and $\underline{q}$ for $(q, 2)$) for participants A (left) and B (right)

We let $\mathsf{G}\downarrow_{\mathsf{A}}$ denote the CFSM in the first component of the triplet returned by the projection when the other components (i.e., the states) are not needed. Let $\Delta(M)$ denote the CFSM obtained by determinising M when interpreting them as finite automata on the alphabet $\mathcal{L}$. The following theorem shows that the system made of the projections of a g-choreography G is deadlock-free if $[\![\mathsf{G}]\!]$ is defined.

Theorem 1 (Progress)**.** *Given $\mathsf{G}\in\mathcal{G}$ such that $[\![\mathsf{G}]\!]\neq\bot$, if s is reachable from the initial configuration s_0 of the communicating system $(\Delta(\mathsf{G}\downarrow_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$ then s is not a deadlock.*

Proof sketch. By structural induction on the syntax of g-choreography. The base cases are straightforward, since the projection of an empty choreography or of a single interaction are deterministic and do not lead to a deadlock by construction. For the inductive steps, we rely on the fact that determinisation of CFSMs preserves the language of the communicating system and does not introduce deadlocks. For sequential and parallel composition, the proof is done by showing that if there is a deadlock in the composed communicating system, then there must be a deadlock in at least one of the constituent systems. This holds straightforwardly for the sequential composition. For the parallel composition, we note that

- in each thread, every output of message, say m, has a corresponding input action in a receiving machine, say B;
- the machine M_{B} of B is the product of the threads on B.

Therefore, the configurations where the message m is sent have to reach a configuration where B has the reception of m enabled (otherwise in one of the threads there would be a deadlock violating the inductive hypothesis). Hence, eventually m will be consumed.

For the non-deterministic composition, we show that if there is a trace in system S made of machines $(\Delta((\mathsf{G}_1+\mathsf{G}_2)\downarrow_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$, then there must be the same trace in one of the systems made of machines $(\Delta(\mathsf{G}_1\downarrow_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$ or $(\Delta(\mathsf{G}_2\downarrow_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$. This is due to the well-branched condition. If the active participant, say B, selects $\mathsf{G}_i\downarrow_{\mathsf{B}}$ in the communicating system S then all other participants are forced to follow the same choice. This allows us to build a simulation relation between the communicating system of the non-deterministic choice and the one consisting of the CFSMs $(\mathsf{G}_i\downarrow_{\mathsf{A}})_{\mathsf{A}\in\mathcal{P}}$. □

The following theorem shows that the traces of the system made of the projections of a g-choreography G are included in the language of the g-choreography if $[\![\mathsf{G}]\!]$ is defined.

Theorem 2 (Adequacy)**.** *If $\mathsf{G}\in\mathcal{G}$ with $[\![\mathsf{G}]\!]\neq\bot$ and $S=(\Delta(\mathsf{G}\downarrow_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$ then $\mathbb{L}_S\subseteq\mathbb{L}_{\mathsf{G}}$.*

Proof sketch. The proof is by structural induction over the syntax of the g-choreographies. The two main proof obligations are to show that (i) the dependencies are preserved in the case of sequential composition and (ii) no additional

communication occurs in the case of parallel composition. For the sequential composition we proceed as follows. By definition, every word w_0 in $\mathbb{L}_{\mathsf{G};\mathsf{G}'}$ is the shuffling of two words, $w \in \mathbb{L}_{\mathsf{G}}$ and $w' \in \mathbb{L}_{\mathsf{G}'}$. Additionally, the side condition of the semantics of sequential composition ensures that all the events of w having subject A precede in w_0 every event of w' with subject A. For the second task we rely on the fact that $[\![\mathsf{G}]\!]$ is defined and we follow the same reasoning done for Theorem 1. Appendix A reports the details of the proof. □

In general, the converse of the inclusion in Theorem 2, that is $\mathbb{L}_{\mathsf{G}} \subseteq \mathbb{L}_S$, does not hold. The reason is that the semantics of parallel composition of g-choreographies does not assume a FIFO policy on channels. In fact, the communicating system can have less behaviours than the interleaving of the two constituent threads because of the additional dependencies imposed by FIFO channels. For instance, take the g-choreography $\mathsf{G} = \mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B} | \mathsf{A} \xrightarrow{\mathsf{y}} \mathsf{B}$; the word $\mathsf{AB!x.AB!y.AB?y.AB?x}$ is in $\mathbb{L}_{\mathsf{G}}$ but it is not in $\mathbb{L}_{(\Delta(\mathsf{G}\downarrow_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}}$.

6. An alternative semantics of global graphs

The presentation of the semantics of choreographies in terms of pomsets has the benefits of being elegant and reusing existing theories [29, 16]. However, this presentation is not ideal for implementing tools[10] based on the semantics. Firstly, there is a computational explosion (common to trace-based methods): each choice duplicates events making the resulting number of pomsets exponential in the number of choices in the choreography. Secondly, the pomset semantics is based on equivalence classes, which are notoriously not straightforward to implement. To overcome these limitations, we give an alternative yet equivalent semantics in terms of *hypergraphs*, which yields compact representations of sets of partial orders.

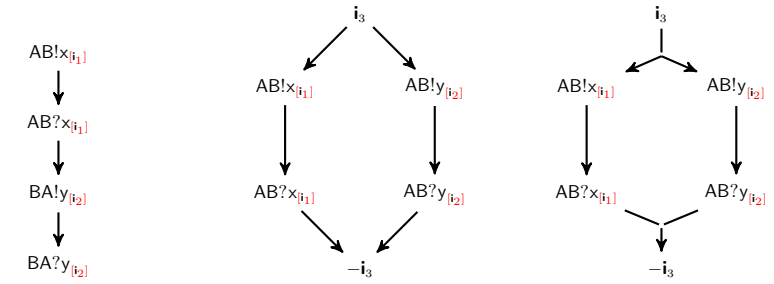
6.1. Preliminaries

Fixed a set $\mathcal{V}$ (of vertices), a (directed) hypergraph on $\mathcal{V}$ is a relation $H \subseteq 2^{\mathcal{V}} \times 2^{\mathcal{V}}$, namely a hyperarc $(\!|\, \tilde{v}, \tilde{v}' \,|\!)$ in H relates two sets of vertices, the source $\tilde{v}$ and the target $\tilde{v}'$. (To avoid cumbersome parenthesis, singleton sets in hyperarcs are shortened by their element, e.g., we write $(\!|\, v, \tilde{v} \,|\!)$ instead of $(\!|\, \{v\}, \tilde{v} \,|\!)$.) The vertices of our hypergraphs are drawn from the set

$$\mathcal{V} = (\mathcal{L} \times \mathcal{K}) \cup \mathcal{K} \qquad \text{(ranged over by } v\text{)}$$

Vertices in $\mathcal{L} \times \mathcal{K}$ represent communication actions together with the originating control points while those in $\mathcal{K}$ represent "non-observable" actions, like (the execution of) a choice or a merge. In the following, we shorten (l, i) as $l_{[\mathsf{i}]}$, extend cp to vertices so that $\mathsf{cp}(v)$ denotes the control point of a vertex v, we use

[10] We are currently developing the **ChorGram** [25, 27] tool chain. In particular, we are extending the toolkit with algorithms based on the semantics presented in this section. However, the development is in too a preliminary phase to be in the scope of this paper.



(a) $H_{(10a)}$ (b) $H_{(10b)}$ (c) $H_{(10c)}$

Figure 10: Example hypergraphs

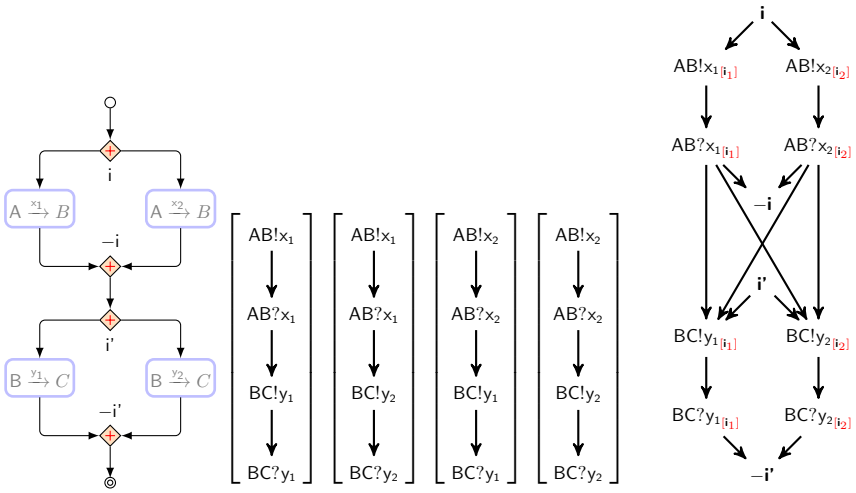
$\mathsf{act}(l_{[\mathsf{i}]}) = l$ to denote the action at $l_{[\mathsf{i}]}$, and we extend function $\mathsf{sbj}(\_)$ to vertices in the obvious way. Take $\mathsf{sbj}(\_)$ and $\mathsf{act}(\_)$ to be undefined on $\mathcal{K}$ and the *subject* of a vertex $l_{[\mathsf{i}]}$ to be simply $\mathsf{sbj}(l)$, the subject of its corresponding action l.

To systematically identify vertices of hypergraphs corresponding to non-observable events, we fix a function $\mu : \mathcal{G} \to (\mathcal{K} \to \mathcal{K})$ such that, for each g-choreography $\mathsf{G} \in \mathcal{G}$, $\mu(\mathsf{G})$ (written μ_{G})

- is bijective when restricted to $\mathsf{cp}(\mathsf{G})$ and
- for all $\mathsf{i} \in \mathsf{cp}(\mathsf{G})$, $\mu_{\mathsf{G}}(\mathsf{i}) \notin \mathsf{cp}(\mathsf{G})$.

Basically, μ yields a bijective correspondence between (i) branch and merge control points corresponding to choices, (ii) fork and join control points corresponding to parallel compositions, and (iii) complementary send/receive actions. Since we decided to concretely represent control points of g-choreographies as positive integers, in the following we assume $\mu_{\mathsf{G}}(\mathsf{i})$ to be the opposite $-\mathsf{i}$, for each $\mathsf{i} \in \mathsf{cp}(\mathsf{G})$.

Examples of hypergraphs. In Fig. 10, the graphs $H_{(10a)}$ and $H_{(10b)}$ contain only simple arcs, while the graph $H_{(10c)}$ contains two hyperarcs: $(\!|\, \mathsf{i}_3, \{\mathsf{AB!x}_{[\mathsf{i}_1]}, \mathsf{AB!y}_{[\mathsf{i}_2]}\} \,|\!)$ and $(\!|\, \{\mathsf{AB?x}_{[\mathsf{i}_1]}, \mathsf{AB?y}_{[\mathsf{i}_2]}\}, -\mathsf{i}_3 \,|\!)$. Intuitively, $H_{(10a)}$ corresponds to the pomset $r_{(6a)}$ of Fig. 6 (establishing a total causal order from the top-most to the bottom-most vertex); graph $H_{(10b)}$ represents a choice at control point i_3 between the left and the right branch and corresponds to the set of pomsets $\mathcal{R}_{(6c)}$ of Fig. 6. Finally, graph $H_{(10c)}$ represents the parallel execution of two threads at the control point i_3 and corresponds to the pomset $r_{(6b)}$; note that the edge $(\!|\, \mathsf{i}_3, \{\mathsf{AB!x}_{[\mathsf{i}_1]}, \mathsf{AB!y}_{[\mathsf{i}_2]}\} \,|\!)$ of $H_{(10c)}$ relates the vertex i_3 to both $\mathsf{AB!x}_{[\mathsf{i}_1]}$ and $\mathsf{AB!y}_{[\mathsf{i}_2]}$. Fig. 11 depicts a further example. Here, both the hypergraph $H_{(11c)}$ and the set of pomsets $\mathcal{R}_{(11b)}$ represent the dependencies of the choreography $\mathsf{G}_{(11a)}$. Notice that the hypergraph based representation does not grow exponentially when choices are composed sequentially.



(a) $\mathsf{G}_{(11a)}$ (b) $\mathcal{R}_{(11b)}$ (c) $H_{(11c)}$

Figure 11: A choreography and its semantics represented as set of pomsets and hypergraph

We define the maximal and minimal elements of H respectively as

$$\mathsf{max}\, H = \{v \in \mathcal{V} \mid \nexists (\!| \tilde{v}, \tilde{v}' |\!) \in H : v \in \tilde{v}\}$$
$$\mathsf{min}\, H = \{v \in \mathcal{V} \mid \nexists (\!| \tilde{v}, \tilde{v}' |\!) \in H : v \in \tilde{v}'\}$$

For instance, $H_{(10b)}$ and $H_{(10c)}$ in Fig. 10 respectively have $\mathsf{min}\, H_{(10b)} = \mathsf{min}\, H_{(10c)} = \{\mathsf{i}_3\}$ and $\mathsf{max}\, H_{(10b)} = \mathsf{max}\, H_{(10c)} = \{-\mathsf{i}_3\}$, while the minimal and maximal elements of $H_{(10a)}$ are $\mathsf{AB!x}_{[\mathsf{i}_1]}$ and $\mathsf{BA?y}_{[\mathsf{i}_2]}$ respectively.

We can now define $\mathtt{seq}(H, H')$, the *sequential* composition of hypergraphs H and H':

$$\begin{aligned}\mathtt{seq}(H, H') \;=\; H \cup H' \cup \big\{ (\!| v, v' |\!) \;\big|\; & \exists (\!| \tilde{v}_1, \tilde{v}_2 |\!) \in H, (\!| \tilde{v}_1', \tilde{v}_2' |\!) \in H' : \\ & v \in (\tilde{v}_1 \cup \tilde{v}_2) \setminus \mathcal{K} \;\wedge\; v' \in (\tilde{v}_1' \cup \tilde{v}_2') \setminus \mathcal{K} \;\wedge \\ & \mathsf{sbj}(v) = \mathsf{sbj}(v') \big\}\end{aligned}$$

As done in Section 4.1 for pomsets, the sequential composition of two hypergraphs H and H' adds dependencies between a vertex in H and one in H' when they have the same subject.

A hyperedge $\mathtt{h} = (\!| \tilde{v}, \tilde{v}' |\!) \in 2^{\mathcal{V}} \times 2^{\mathcal{V}}$ represents that the *causes* in $\tilde{v}$ must precede the *effects* in $\tilde{v}'$. Let $\mathtt{cs}, \mathtt{ef} : 2^{\mathcal{V}} \times 2^{\mathcal{V}} \to 2^{\mathcal{V}}$ be the maps respectively returning the causes ($\mathtt{cs}$) and effects ($\mathtt{ef}$) of a hyperedge, that is: if $\mathtt{h} = (\!| \tilde{v}, \tilde{v}' |\!)$ then $\mathtt{cs}(\mathtt{h}) = \tilde{v}$ and $\mathtt{ef}(\mathtt{h}) = \tilde{v}'$. Given $H, H' \subseteq 2^{\mathcal{V}} \times 2^{\mathcal{V}}$, define the hypergraph

$$H \circ H' = \{ (\!| \mathtt{cs}(\mathtt{h}), \mathtt{ef}(\mathtt{h}') |\!) \mid \mathtt{h} \in H, \mathtt{h}' \in H', \mathtt{ef}(\mathtt{h}) \cap \mathtt{cs}(\mathtt{h}') \neq \emptyset \}$$

That is, $H \circ H'$ is the relational composition of H and H'. The reflexive-transitive closure $H^\star$ of H with respect to the composition relation $\circ$ is defined as expected

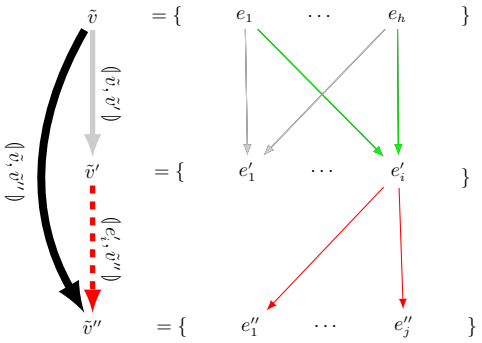


Figure 12: Composition of hyperedges

as $H^\star = \bigcup_{n\geq 0} \underbrace{H \circ \cdots \circ H}_{n\text{-times}}$. Fig. 12 yields an intuitive representation of how causal relations follow composition: the vertices in $\tilde{v}$ cause all the vertices in $\tilde{v}''$ due to the dependency of the vertex e'_i from the vertices in $\tilde{v}$ and the fact that e'_i causes each of the vertices in $\tilde{v}''$. Formally, the *happens-before relation* $\widehat{H} \subseteq \mathcal{V} \times \mathcal{V}$ *induced by a hypergraph* H is

$$\widehat{H} \quad = \quad \bigcup_{\langle\!| \tilde{v}, \tilde{v}' |\!\rangle \in H} \tilde{v} \times \tilde{v}'$$

Basically, $\langle v, v' \rangle \in \widehat{H}$ when v precedes v' in H, namely $\widehat{H}$ are the causal dependencies among the vertices in H.

Let H be an acyclic hypergraph, the happens-before relation of the reflexive-transitive closure of H yields a partial order on the vertices defined as

$$\sqsubseteq_H \quad = \quad \widehat{(H^\star)}$$

Sometimes it will be convenient to take the intersection of sets of vertices $\tilde{v} \sqcap \tilde{v}'$ disregarding their control points: $\tilde{v} \sqcap \tilde{v}' = \mathsf{act}(\tilde{v}) \cap \mathsf{act}(\tilde{v}')$.

6.2. Semantics of choreographies

The semantics of a g-choreography is a partial map $\{\!|\_|\!\} : \mathcal{G} \to 2^{(2^{\mathcal{V}} \times 2^{\mathcal{V}})}$ defined by induction on the structure of choreographies as per Equations 3-7 below. The base cases and the inductive case for the parallel composition are very simple:

$$\{\!|\mathbf{0}|\!\} = \emptyset \tag{3}$$

$$\{\!|\mathsf{i}\colon \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}|\!\} = \{\langle\!| \mathsf{AB}!\mathsf{m}_{[\mathsf{i}]}, \mathsf{AB}?\mathsf{m}_{[\mathsf{i}]} |\!\rangle\} \tag{4}$$

$$\{\!|\mathsf{i}\!:\!(\mathsf{G}|\mathsf{G}')|\!\} = \begin{cases} \{\!|\mathsf{G}|\!\} \cup \{\!|\mathsf{G}'|\!\} \cup H & \text{if } (\{\!|\mathsf{G}|\!\} \sqcap \{\!|\mathsf{G}'|\!\}) \cap \mathcal{L}^? = \emptyset \\ \bot & \text{otherwise} \end{cases} \tag{5}$$

where, in (5), $H = \{(\!|\, \mathsf{i}, \mathsf{min}\,\{\!|\mathsf{G}|\!\} \cup \mathsf{min}\,\{\!|\mathsf{G}'|\!\} \,|\!), (\!|\, \mathsf{max}\,\{\!|\mathsf{G}|\!\} \cup \mathsf{max}\,\{\!|\mathsf{G}'|\!\}, -\mathsf{i} \,|\!)\}$ and the side condition is analogous to the well-forkedness condition in Section 4.2. Note that the hyperedges in H are not necessary and are added only to make the graphs connected so to simplify the definition of some of the following constructions.

The case of sequential composition requires some auxiliary definitions. We first define the (hyperedges involving) "first" and the "last" communication actions in a hypergraph H:

$$\mathsf{fst}\, H = \{(\!|\, \tilde{v}, \tilde{v}' \,|\!) \in H \;\big|\; (\tilde{v} \cup \tilde{v}') \cap \mathcal{K} = \emptyset \;\wedge\; \forall (\!|\, \tilde{v}'', \tilde{v} \,|\!) \in H^\star : \tilde{v}'' \setminus \tilde{v} \subseteq \mathcal{K}\}$$
$$\mathsf{lst}\, H = \{(\!|\, \tilde{v}, \tilde{v}' \,|\!) \in H \;\big|\; (\tilde{v} \cup \tilde{v}') \cap \mathcal{K} = \emptyset \;\wedge\; \forall (\!|\, \tilde{v}', \tilde{v}'' \,|\!) \in H^\star : \tilde{v}'' \setminus \tilde{v}' \subseteq \mathcal{K}\}$$

(note that $\mathsf{fst}\, H = \left(\mathsf{lst}\,(H^{-1})\right)^{-1}$). Also, let $H_{seq} = \mathtt{seq}(H, H')$ and $H_{com} = \mathtt{cs}(\mathsf{lst}\, H) \times \mathtt{ef}(\mathsf{fst}\, H')$; the predicate $\mathrm{ws}(H, H')$ is defined as $\widehat{H_{seq}} \supseteq \widehat{H_{com}}$, which requires that the sequential composition covers the dependencies between the causes of the last communication actions of H with the effects of the first actions of H'. This condition corresponds to the predicate $ws(r, r')$ of Section 4.2.

The semantics of sequential composition can now be defined as follows:

$$\{\!|\mathsf{G}; \mathsf{G}'|\!\} = \begin{cases} \mathtt{seq}(\{\!|\mathsf{G}|\!\}, \{\!|\mathsf{G}'|\!\}) & \text{if } \mathrm{ws}(\{\!|\mathsf{G}|\!\}, \{\!|\mathsf{G}'|\!\}) \\ \bot & \text{otherwise} \end{cases} \tag{6}$$

Equation 6 establishes happens-before relations as computed by $\mathtt{seq}(\{\!|\mathsf{G}|\!\}, \{\!|\mathsf{G}'|\!\})$ provided that the well-sequencedness condition holds. Note that well-sequencedness is violated in the example in Fig. 7e (on page 14).

Similarly to the pomset semantics of Section 4.2, the semantics of a choice is defined provided that well-branchedness holds; this amounts to saying that (i) there is at most one active participant and (ii) all the other participants are passive. However, the notions of active and passive participants for the hypergraph semantics are defined in a different way; we give the new definitions of active and passive participants in Definitions 15 and 16 below and prove them equivalent to the notions in Definitions 13 and 14 in Appendix B.

$$\{\!|\mathsf{i}{:}(\mathsf{G} + \mathsf{G}')|\!\} = \begin{cases} \{\!|\mathsf{G}|\!\} \cup \{\!|\mathsf{G}'|\!\} \cup H & \text{if } \mathrm{wb}(\mathsf{G}, \mathsf{G}') \text{ and} \\ & H = \{(\!|\, \mathsf{i}, \mathsf{min}\,\{\!|\mathsf{G}|\!\} \,|\!), (\!|\, \mathsf{i}, \mathsf{min}\,\{\!|\mathsf{G}'|\!\} \,|\!), \\ & \qquad (\!|\, \mathsf{max}\,\{\!|\mathsf{G}|\!\}, -\mathsf{i} \,|\!), (\!|\, \mathsf{max}\,\{\!|\mathsf{G}'|\!\}, -\mathsf{i} \,|\!)\} \\ \bot & \text{otherwise} \end{cases} \tag{7}$$

The semantics of a choice $\mathsf{i}{:}(\mathsf{G} + \mathsf{G}')$ returns the hypergraphs of the branches connected to the control points of the start and merge of the choice. Besides the dependencies induced by G and G', $\{\!|\mathsf{i}{:}(\mathsf{G} + \mathsf{G}')|\!\}$ contain those making i (the control point of the branch) precede all minimal vertices of G and G'; similarly, the maximal vertices of G and G' have to precede the conclusion of the choice (marked by the control point $-\mathsf{i}$). This captures the fact that either the actions of the first branch or the actions of the second one should be performed.

Lemma 2. $\forall \mathsf{G} \in \mathcal{G} : (\!|\, i, \tilde{v} \,|\!), (\!|\, i, \tilde{v}' \,|\!) \in \{\!|\mathsf{G}|\!\} \implies \tilde{v} \cap \tilde{v}' = \emptyset$.

Proof. The lemma is immediate given that control points cannot have more than one occurrence in a g-choreography G. □

Given a hypergraph H, we say that $\tilde{v}$ *causes* $\tilde{v}'$ *(in* H*)* if $\tilde{v} \neq \tilde{v}'$ and $(\!|\, \tilde{v}, \tilde{v}' \,|\!) \in H^\star$.

Lemma 3 (Acyclicity)**.** *For all* $\mathsf{G} \in \mathcal{G}$ *such that* $\{\!|\mathsf{G}|\!\}$ *is defined, if* $\tilde{v}$ *causes* $\tilde{v}'$ *in* $\{\!|\mathsf{G}|\!\}$ *then* $\tilde{v}'$ *does not cause* $\tilde{v}$ *in* $\{\!|\mathsf{G}|\!\}$*.*

Proof. By induction on the syntax of G. If $\mathsf{G} = \mathbf{0}$ then $\{\!|\mathsf{G}|\!\} = \emptyset$ and if $\mathsf{G} = \mathsf{i} \colon \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$ then $\{\!|\mathsf{G}|\!\} = \{(\!|\, \mathsf{AB}!\mathsf{m}_{[\mathsf{i}]}, \mathsf{AB}?\mathsf{m}_{[\mathsf{i}]} \,|\!)\}$ hence the thesis holds vacuously. For the parallel composition $\mathsf{G} = \mathsf{i}{:}(\mathsf{G}_1|\mathsf{G}_2)$, we have $\{\!|\mathsf{G}|\!\} = \{\!|\mathsf{G}_1|\!\} \cup \{\!|\mathsf{G}_2|\!\} \cup \{(\!|\, \mathsf{i}, \mathsf{min}\{\!|\mathsf{G}_1|\!\} \cup \mathsf{min}\{\!|\mathsf{G}_2|\!\} \,|\!), (\!|\, \mathsf{max}\{\!|\mathsf{G}_1|\!\} \cup \mathsf{max}\{\!|\mathsf{G}_2|\!\}, -\mathsf{i} \,|\!)\}$; by induction we have the thesis since each vertex of G is either a control point j or a pair $(\tilde{l}, \mathsf{j})$ and each i has a unique occurrence in G, then the vertices of $\{\!|\mathsf{G}_1|\!\}$ and $\{\!|\mathsf{G}_2|\!\}$ are disjoint. A similar argument applies in the remaining cases noting that the additional dependencies added in the sequential or non-deterministic composition are all between vertices taken from disjoint sets of vertices. □

The notions of well-sequencedness and of well-branchedness (used in the last two clauses in the definition of $\{\!|\_|\!\}$) are very similar to the corresponding ones for pomsets given in Section 4.2. However, they are defined on different structures (indeed, the attentive reader would have noticed the different fonts) and require alternative definitions of active and passive participants, which have now to be adapted to the hypergraphs-based semantics.

The A-*only* part of a set of vertices $\tilde{v} \in 2^{\mathcal{V}}$ for a participant $\mathsf{A} \in \mathcal{P}$ is the set $\tilde{v}_{@\mathsf{A}}$ where the actions of $\tilde{v}$ not having subject A are replaced with the control point of the action; formally

$$\begin{array}{rcl} \tilde{v}_{@\mathsf{A}} & = & \{v \in \tilde{v} \;\big|\; \mathsf{sbj}(v) = \mathsf{A} \vee v \in \mathcal{K}\} \\ & \cup & \{\mathsf{cp}(v) \;\big|\; v \in \tilde{v} \cap (\mathcal{L}^! \times \mathcal{K}) \;\wedge\; \mathsf{sbj}(v) \neq \mathsf{A}\} \\ & \cup & \{-\mathsf{cp}(v) \;\big|\; v \in \tilde{v} \cap (\mathcal{L}^? \times \mathcal{K}) \;\wedge\; \mathsf{sbj}(v) \neq \mathsf{A}\} \end{array}$$

Accordingly, the A-*only* part of a hypergraph H is defined as

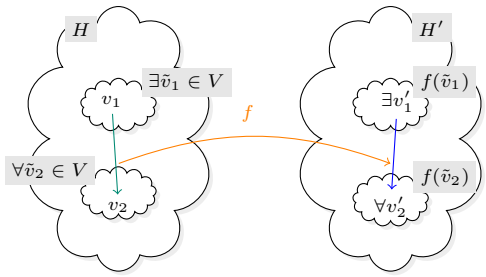
$$H_{@\mathsf{A}} = \bigcup_{(\!|\, \tilde{v}, \tilde{v}' \,|\!) \in H} \{(\!|\, \tilde{v}_{@\mathsf{A}}, \tilde{v}'_{@\mathsf{A}} \,|\!)\}$$

For instance, the B-only part of $H_{(10c)}$ (on page 27) is

$$\{(\!|\, \mathsf{i}_3, \{\mathsf{i}_1, \mathsf{i}_2\} \,|\!),\ (\!|\, \mathsf{i}_1, \mathsf{AB}?\mathsf{x}_{[\mathsf{i}_1]} \,|\!),\ (\!|\, \mathsf{i}_2, \mathsf{AB}?\mathsf{y}_{[\mathsf{i}_2]} \,|\!),\ (\!|\, \{\mathsf{AB}?\mathsf{x}_{[\mathsf{i}_1]}, \mathsf{AB}?\mathsf{y}_{[\mathsf{i}_2]}\}, -\mathsf{i}_3 \,|\!)\}$$

Notice that we use $\mathsf{cp}(v)$ and $-\mathsf{cp}(v)$ for outputs and inputs respectively, so that different vertices not belonging to A remain distinguished and the causality relations are not spoiled.

Given a hypergraph H, $v_1, v_2 \in H$ are *independent in* H if there are $\mathtt{h} \in H$ and $v_1', v_2' \in \mathtt{ef}(\mathtt{h})$ such that, for each $i, j \in \{1, 2\}$, $\langle v_i', v_j \rangle \in \widehat{(H^\star)} \iff i = j$.



The relations H and H' have to be thought of as specifying the causal relations of two branches of a distributed choice. All the vertices of $\tilde{v} \in V$ and of $f(\tilde{v})$ have the same subject.
The bijection f preserves both actions and causality relation in all the sets $\tilde{v} \in V$. So, any predecessor v_1 in H of a vertex $v_2 \in \tilde{v}_2$ with the same subject as v_2 must be in a set $\tilde{v}_1$ of V. Moreover, the f-images of $\tilde{v}_1$ and $\tilde{v}_2$ must reflect such order in H'. Such condition must also hold for the inverse of f.

Figure 13: Reflectivity

Intuitively, two vertices are independent if they have a common fork ancestor and belong to two different threads. If the first common ancestor is a branch, then the two events are not independent, since they can not both occur in the same execution. For instance, vertices $\mathsf{AB?x}_{[i_1]}$ and $\mathsf{AB?y}_{[i_2]}$ of Fig. 10b are not independent, because their common ancestor is a branch (i.e. the two events do not belong to the same edge departing from the ancestor and, therefore, are mutually exclusive). Vertices $\mathsf{AB!x}_{[i_1]}$ and $\mathsf{AB?x}_{[i_1]}$ of Fig. 10c are not independent, because they are successors of the same child of the common ancestor (i.e. they belong to the same thread). Finally, vertices $\mathsf{AB?x}_{[i_1]}$ and $\mathsf{AB?y}_{[i_2]}$ of Fig. 10c are independent. For a participant $\mathsf{A} \in \mathcal{P}$, a set of vertices $\tilde{v} \subseteq H$ is A-*uniform in* H if $\tilde{v} \cap \mathcal{K} = \emptyset$, $\mathsf{sbj}(\tilde{v}) = \{\mathsf{A}\}$, $\mathsf{act}(\tilde{v})$ is a singleton, and each $v \neq v' \in \tilde{v}$ are not independent and are such that $\{\langle v, v'\rangle, \langle v', v\rangle\} \cap \widehat{(H^\star)} = \emptyset$.

The notions of active and passive participants of choices for the hypergraph-based semantics are defined in terms of *reflectivity*. A partition V of a subset of vertices of H A-*reflects* a partition V' of a subset of vertices of H' if there is a bijection $f : V \to V'$ such that the following conditions hold:

- for each $\tilde{v} \in V$ both $\tilde{v}$ and $f(\tilde{v})$ are A-uniform and $\mathsf{act}(\tilde{v}) = \mathsf{act}(f(\tilde{v}))$
- $\forall \tilde{v}_2 \in V, v_2 \in \tilde{v}_2,\ \langle v_1, v_2\rangle \in \widehat{H} : \mathsf{sbj}(v_1) = \mathsf{A} \implies \big(\exists \tilde{v}_1 \in V : v_1 \in \tilde{v}_1 \ \wedge\ \forall v \in \tilde{v}_2, v' \in \tilde{v}_1 : \langle v, v'\rangle \notin \widehat{H} \ \wedge\ \forall v'_2 \in f(\tilde{v}_2) \exists v'_1 \in f(\tilde{v}_1) : \langle v'_1, v'_2\rangle \in \widehat{H'}\big)$, and symmetrically
- $\forall \tilde{v}'_2 \in V', v'_2 \in \tilde{v}'_2,\ \langle v'_1, v'_2\rangle \in \widehat{H'} : \mathsf{sbj}(v'_1) = \mathsf{A} \implies \big(\exists \tilde{v}'_1 \in V' : v'_1 \in \tilde{v}'_1 \ \wedge\ \forall v \in \tilde{v}'_2, v' \in \tilde{v}'_1 : \langle v, v'\rangle \notin \widehat{H'} \ \wedge\ \forall v_2 \in f^{-1}(\tilde{v}'_2) \exists v_1 \in f^{-1}(\tilde{v}'_1) : \langle v_1, v_2\rangle \in \widehat{H}\big)$.

The notion of *reflection* has an intuitive explanation given in Fig. 13.

For a participant $\mathsf{A} \in \mathcal{P}$, two g-choreographies $\mathsf{G}, \mathsf{G}' \in \mathcal{G}$, a partition V of a subset of vertices of $\{\!|\mathsf{G}|\!\}$, and a partition V' of a subset of vertices of $\{\!|\mathsf{G}'|\!\}$ we say that $(\tilde{v}_1, \tilde{v}_2)$ is the A-*branching pair of* $\mathsf{G} + \mathsf{G}'$ *with respect to* V *and* V',

denoted $(\tilde{v}_1, \tilde{v}_2) = \mathtt{div}_{\mathsf{A}}^{V,V'}(\mathsf{G}, \mathsf{G}')$, if

$$V' \text{ A-reflects } V \quad \text{and} \quad \begin{cases} \tilde{v}_1 = \bigcup \mathtt{cs}(\mathsf{fst}\,(\{\!|\mathsf{G}|\!\}_{@\mathsf{A}})) \setminus \bigcup V \\ \text{and} \\ \tilde{v}_2 = \bigcup \mathtt{cs}(\mathsf{fst}\,(\{\!|\mathsf{G}'|\!\}_{@\mathsf{A}})) \setminus \bigcup V' \end{cases}$$

The requirement of A-*reflectivity* is used to identify such common behaviour (i.e. all vertices in V and V') and to ignore it when checking the behaviour of A in the branches. This has a similar role of prefix-maps of Section 4.2. In fact, by taking the A-only parts of these hypergraphs and selecting their first interactions (that is the A-branching pair $\tilde{v}_1$, $\tilde{v}_2$) we identify when the behaviour of A in G starts to be different with respect to the behaviour in G'.

Definition 15. *A participant* $\mathsf{A} \in \mathcal{P}$ *is* active *in* $\mathsf{G} + \mathsf{G}'$ *if there are* V *and* V' *such that* $\mathtt{div}_{\mathsf{A}}^{V,V'}(\mathsf{G}, \mathsf{G}')$ *is defined and, if* $(\tilde{v}_1, \tilde{v}_2) = \mathtt{div}_{\mathsf{A}}^{V,V'}(\mathsf{G}, \mathsf{G}')$ *then the following holds:*

$$\tilde{v}_1 \cup \tilde{v}_2 \subseteq (\mathcal{L}^! \times \mathcal{K}) \qquad \tilde{v}_1 \sqcap \tilde{v}_2 = \emptyset \qquad \tilde{v}_1 \neq \emptyset \qquad \tilde{v}_2 \neq \emptyset$$

Given a g-choreography $\mathsf{G} \in \mathcal{G}$, the happens-before relation of the reflexive-transitive closure of $\{\!|\mathsf{G}|\!\}$ yields a partial order (by Lemma 3) on the vertices of G defined as

$$\sqsubseteq_{\mathsf{G}} \quad = \quad \begin{cases} (\widehat{\{\!|\mathsf{G}|\!\}^\star}), & \text{if } \{\!|\mathsf{G}|\!\} \text{ is defined} \\ \emptyset, & \text{otherwise} \end{cases}$$

Observe that $\sqsubseteq_{\mathsf{G}} = \sqsubseteq_{\{\mathsf{G}\}}$ when $\{\!|\mathsf{G}|\!\}$ is defined.

Definition 16. *A participant* $\mathsf{A} \in \mathcal{P}$ *is* passive *in* $\mathsf{G} + \mathsf{G}'$ *if there are* V *and* V' *such that* $\mathtt{div}_{\mathsf{A}}^{V,V'}(\mathsf{G}, \mathsf{G}')$ *is defined and, if* $(\tilde{v}_1, \tilde{v}_2) = \mathtt{div}_{\mathsf{A}}^{V,V'}(\mathsf{G}, \mathsf{G}')$ *then the following holds:*

$$\begin{array}{r|l} \tilde{v}_1 \sqcap \{v \in \mathsf{G}'\} & \nexists v' \in \tilde{v}_2 : v \sqsubseteq_{\mathsf{G}'} v'\} = \emptyset \\ \tilde{v}_2 \sqcap \{v \in \mathsf{G}\} & \nexists v' \in \tilde{v}_1 : v \sqsubseteq_{\mathsf{G}} v'\} = \emptyset \end{array} \qquad \begin{array}{l} \tilde{v}_1 \cup \tilde{v}_2 \subseteq (\mathcal{L}^? \times \mathcal{K}) \\ \tilde{v}_1 = \emptyset \iff \tilde{v}_2 = \emptyset \end{array}$$

6.3. Languages of choreographies from hypergraphs

Informally, the language of a g-choreography $\mathsf{G} \in \mathcal{G}$ consists of the sequences of words made of the communication actions of the vertices in G that preserve the causal relations of $\{\!|\mathsf{G}|\!\}$, provided that $\{\!|\mathsf{G}|\!\}$ is defined.

Given a g-choreography $\mathsf{G} \in \mathcal{G}$, let $\mathsf{G}^{\oplus} = \{\!|\mathsf{G}|\!\} \cap (2^{\mathcal{K}} \times 2^{\mathcal{V}})$ (the hyperedges in G whose source is a control point) and let $D = \bigcup \mathtt{cs}(\mathsf{G}^{\oplus})$ be the set of vertices in $\mathsf{G}^{\oplus}$ which do not represent communication events. A map $\rho : D \to 2^{\mathcal{V}}$ is a *resolution of* G if $(\!|\mathsf{i}, \rho(\mathsf{i})|\!) \in \mathsf{G}^{\oplus}(\mathsf{i})$ for each $\mathsf{i} \in D$, where $\mathsf{G}^{\oplus}(\mathsf{i})$ is the set of the outgoing hyperedges of i defined as $\mathsf{G}^{\oplus}(\mathsf{i}) = \mathsf{G}^{\oplus} \cap \{(\!|\mathsf{i}, \tilde{v}|\!) \mid \tilde{v} \subseteq \mathcal{V}\}$. We will confound ρ with its graph $\{(\!|\mathsf{i}, \rho(\mathsf{i})|\!) \mid \mathsf{i} \in D\}$. The partial order corresponding to a resolution is the reflexive and transitive closure of the relation obtained by (i) removing the hyperedges not chosen by the resolution and (ii) removing every

dead vertex (i.e. vertices that are not reachable from the initial vertices after removing the non-selected hyperedges).

We use $\Re_{\mathsf{G}}$ to denote the set of resolutions of G. Intuitively, a resolution fixes a branch for every choice in a g-choreography G and therefore it induces a partial order of the vertices compatible with G and the resolution. Formally, given a resolution $\rho \in \Re_{\mathsf{G}}$ of G, we define the hypergraph $\mathsf{G}\circledR\rho$ as follows:

$$\mathsf{G}\circledR\rho \quad = \quad \mathit{reachable}\Big(\mathsf{min}\,\{\!|\mathsf{G}|\!\},\ \{\!|\mathsf{G}|\!\} \setminus \big(\bigcup_{\mathsf{i}\in\mathrm{dom}\,\rho} \mathsf{G}^{\oplus}(\mathsf{i}) \setminus \rho\big)\Big)$$

where $\mathit{reachable}(\tilde{v}, H)$ is the function that removes every node in the hypergraph H that is not reachable from $\tilde{v}$. The important property of $\mathsf{G}\circledR\rho$ is that resolutions yield consistent choices.

Lemma 4 (Consistency)**.** *Given a resolution $\rho \in \Re_{\mathsf{G}}$ of a g-choreography $\mathsf{G} \in \mathcal{G}$, if for all $i \neq j \in \mathrm{dom}\,\rho$*

$$(\!|\, i, \rho(i) \,|\!) \in \mathsf{G}\circledR\rho \quad \textit{and} \quad i \sqsubseteq_{\mathsf{G}} j \quad \textit{and} \quad \forall v \in \rho(i) : v \not\sqsubseteq_{\mathsf{G}} j$$

then $\forall \mathtt{h} \in \mathsf{G}\circledR\rho : j \notin \mathtt{cs}(\mathtt{h})$.

Proof. Let $\bar{\rho} = \bigcup_{\mathsf{k}\in\mathrm{dom}\,\rho} \mathsf{G}^{\oplus}(\mathsf{k}) \setminus \rho$. Since $\mathsf{i} \sqsubseteq_{\mathsf{G}} \mathsf{j}$, there must be $(\!|\, \mathsf{i}, \tilde{v} \,|\!) \in \mathsf{G}^{\oplus}(\mathsf{i})$ such that $v \sqsubseteq_{\mathsf{G}} \mathsf{j}$ for a $v \in \tilde{v}$. Also, it must be $\rho(\mathsf{i}) \neq \tilde{v}$ (otherwise there is a vertex in $\rho(\mathsf{i})$ preceeding j), therefore $(\!|\, \mathsf{i}, \tilde{v} \,|\!) \in \bar{\rho}$ (by Lemma 2). Hence, each vertex in $\tilde{v}$ will not be in $\{\!|\mathsf{G}|\!\} \setminus \bar{\rho} \supseteq \mathsf{G}\circledR\rho$, which makes j not reachable in $\mathsf{G}\circledR\rho$ (otherwise we would contradict the hypothesis that $(\!|\, \mathsf{i}, \rho(\mathsf{i}) \,|\!) \in \mathsf{G}\circledR\rho$ or that each vertex in $\rho(\mathsf{i})$ does not preceed j). Therefore for any $\mathtt{h} \in \mathsf{G}\circledR\rho$ we have that $\mathsf{j} \notin \mathtt{cs}(\mathtt{h})$. □

Let $\Sigma_{\mathsf{G}\circledR\rho} = \{v \in \mathcal{L} \times \mathcal{K} \mid v \text{ is a vertex in } \mathsf{G}\circledR\rho\}$. The *language* of $\mathsf{G} \in \mathcal{G}$ is undefined when $\{\!|\mathsf{G}|\!\}$ is undefined, otherwise it is

$$\mathbb{L}_{\mathsf{G}} = \bigcup_{\rho\in\Re_{\mathsf{G}}} \big\{\mathsf{act}(w) \;\big|\; w \in (\Sigma_{\mathsf{G}\circledR\rho})^{\star} \ \wedge\ \gamma(w, \rho)\big\}$$

where $\gamma(w, \rho)$ holds iff for all $i \neq j$ between 1 and the length of w we have that

1. $w[i] \neq w[j]$, where $w[i]$ stands for the i-th vertex in w
2. if $w[i] \sqsubseteq_{\mathsf{G}\circledR\rho} w[j]$ and $w[i] \neq w[j]$ then $i < j$
3. if $v \sqsubseteq_{\mathsf{G}\circledR\rho} w[i]$ and $v \neq w[i]$ then there is $h < i$ such that $w[h] = v$

Clause 1 states that vertices in the word are not repeated and, since $w \in (\Sigma_{\mathsf{G}\circledR\rho})^{\star}$, by the consistency lemma (Lemma 4), w is made only of vertices present in the order of the resolution, i.e. w does not mix vertices belonging to different branches. Clause 2 states that words preserve the causal relations of vertices. Clause 3 requires that all the predecessors of an event in the word must precede the event in the word. Notice that $\mathbb{L}_{\mathsf{G}}$ is prefix-closed.

Let $\mathsf{G} \in \mathcal{G}$ be a g-choreography such that $[\![\mathsf{G}]\!]$ and $\{\!|\mathsf{G}|\!\}$ are defined. We say that the two semantics are equivalent if the languages of $[\![\mathsf{G}]\!]$ and $\{\!|\mathsf{G}|\!\}$ are

equal. This is demonstrated by showing language equivalence between pomsets and resolutions. Let $r \in [\![\mathsf{G}]\!]$ be a pomset and ρ be a resolution of G, we say that r is equivalent to ρ when ρ *corresponds to* r *up to* $\{\!|\mathsf{G}|\!\}$ and each event in r *corresponds* to some vertex in $\{\!|\mathsf{G}|\!\}$; these correspondences are formalised in Appendix B (on page 56); here it suffices to give an intuition:

- a vertex v in $\{\!|\mathsf{G}|\!\}$ *corresponds to* $e \in \mathcal{E}_r$ for an $r \in [\![\mathsf{G}]\!]$ (and vice versa) when $\lambda_r(e)$ is the label of v and for each $v' \sqsubseteq_{\mathsf{G}} v$ on the same branch of v there is an event $e' \in \mathcal{E}_r$ such that $e' \leq_r e$ with e' corresponding to v'.
- a resolution $\rho \in \Re_{\mathsf{G}}$ *corresponds to* $r \in [\![\mathsf{G}]\!]$ when for all vertices v on the branch corresponding to ρ wich are not in $\mathcal{K}$ there is $e \in r$ that corresponds to v (basically, there is a partial order homomorphism between ρ and r).

To prove the equivalence of our two semantics we must show that there is a bijective correspondence between pomsets in $[\![\mathsf{G}]\!]$ and the set $\Re_{\mathsf{G}}$ of resolutions of $\{\!|\mathsf{G}|\!\}$.

Theorem 3 (Correctness and completeness)**.** *For each* $\mathsf{G} \in \mathcal{G}$, $\{\!|\mathsf{G}|\!\} \neq \bot$ *iff* $[\![\mathsf{G}]\!] \neq \bot$. *Moreover, for each* $\rho \in \Re_{\mathsf{G}}$ *there is a pomset* $r \in [\![\mathsf{G}]\!]$ *equivalent to* ρ *and vice versa.*

Appendix B reports the details of the proof.

7. Conclusions

We introduced an abstract semantics framework of choreographies expressed as global graphs. Our approach is oblivious of the underlying communication semantics and, as discussed below, can be easily adapted to alternative semantics. The semantics of the global artefacts permits to analyse the distributed coordination at the global-level, without the need of examining the local behaviours produced by the projections. Moreover, we can establish precise relations between specifications (choreographies) and their implementations (local artefacts) and we can verify correctness of projections and refinements.

Our framework is more expressive than existing ones; it allows the same participant to operate in both threads of the parallel composition and it does not force passive participants to receive a message signalling the selected choice as first operation in a non-deterministic composition. This is possible due to the well-branched condition. Interestingly, this condition is parametric and depends on the strategy used to find the partitions and the common prefixes required to identify the A-prefix-map. This can range from using always the trivial partitions and the empty prefix (thus enforcing the same syntactical constrains of the existing proposals, some of which have been discussed in Section 2) to finding the partitioning relative to the longest common prefixes. We plan to give a more formal comparison between our semantics and those available in the literature. For this, an interesting approach would be to follow the ideas applied to CCS in [4]. There pomsets where used in combination with proved

transition systems to give an non-interleaving semantics of CCS; basically, given a sequence of transitions $p \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_n} q$ between two CCS processes p and q, a pomset r can be derived from a *proved transition system* so that r represents the equivalence class of traces between p and q "compatible" with traces labelled $\alpha_1, \ldots, \alpha_n$.

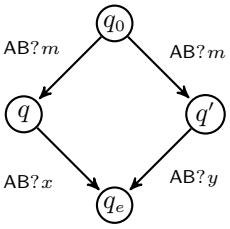
The adequacy of our framework is demonstrated by considering the projections of g-choreographies on communicating machines. With this aim we provide a projection algorithm and we prove its soundness, showing (Theorem 1) that the projection of a sound choreography (i.e. whose semantics is defined) is a deadlock-free system and (Theorem 2) that every execution of the projections is accepted by the choreography.

Theorem 2 manifests the independence of the global semantics from the local one. We regard as a good property of our semantics the fact that global artefacts have "more executions" than the local ones obtained from their projections. Intuitively, this amounts to saying that projections are refinements of the (more abstract) global view. Another advantage is that changing local artefacts does not necessarily require to modify the semantics of the global view. For example, if we consider CFSMs where buffers are used as multisets (instead of as FIFO queues), then all our constructions apply and Theorem 2 can be proved with language equality rather than just inclusion.

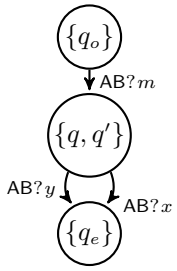
Our semantic framework is amenable to variations to consider different semantics at the global level. For instance, an alternative semantics of global views could consider out-of-order outputs; this can be easily formalised by removing the causal dependency between the outputs of two sequential interactions (i.e. the topmost dotted arrows of both Fig. 7a and Fig. 7d are removed). However, soundness of this change depends on the semantics of the local artefacts. In fact, the projections of Fig. 7d can lead to a deadlock if the outputs are interleaved and FIFO CFSM are used as local artefacts, while the interleaved outputs do not cause deadlocks if multiset buffers are used by local artefacts. As another variant one could consider a semantics where a sender has to wait for the receiver to consume the sent messages before proceeding; this is simply attained by adding a causal dependency from the input of B in Fig. 7a to the output from A to C (while removing the dotted relation). We conjecture that this semantics would correspond to the half-duplex semantics of CFSMs. Finally, one can allow sequentially composing choreographies that involve disjoint participants (i.e. allowing the choreography in Fig. 7e since both C and D do not occur in $\mathsf{A} \xrightarrow{x} \mathsf{B}$).

For simplicity we did not considered recursion/iteration. This can be simply added with a sort of an iterative construct $!\mathsf{G}$ where after the execution of the body G, participants "agree" about repeating the loop (unfolding it and generating new communication events) or exiting it. This essentially reduces the problem to a distributed choice, which we solved with the well-branchedness condition. Intuitively, one has to be careful when giving the semantics of $(!\mathsf{G}); \mathsf{G}'$. In fact, in order to decide when to exit the body of the iteration G and continue with G' we have to require the existence of an active participant, the well-branchedness of $\mathsf{G} + \mathsf{G}'$, and the well-sequencedness of $\mathsf{G}; \mathsf{G}'$.

In order to simplify the development of tools based on the semantics we also provide an alternative semantics of choreographies in terms of hypergraphs. Even if its presentation is more cumbersome, it has the benefit that its size does not grow exponentially with the number of choices in the choreography. Moreover, we believe that our semantics could lead to alternative projection algorithms. For instance, we plan to define projections that exploit reflections (which in the hypergraph based semantics resembles the notion of A-prefix-map). This could be better explained by observing what happens when projecting the simple choreography $\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}; \mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B} + \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}; \mathsf{A} \xrightarrow{\mathsf{y}} \mathsf{B}$, say on participant B (we ignore control points because immaterial). Our algorithm yields the following machine:



which after determinisation becomes



However, exploiting the bijection of the reflection, one could directly obtain the machine on the right (avoiding the cost of determinising machines). Note that other projection algorithms capable of handling the example above (as e.g., the one in [26]) also require determinisation, while projections based on types (as e.g., the ones in [21]) are undefined on the previous example because they require prefixes of branches to be pairwise different. Finally, the hypergraph based semantics can be easily extended to graphs with structured loops that are represented as repetitions of g-choreography. This is possible since the semantics side-conditions do not depend on the (possibly infinite) language of the choreography, but rather on the hypergraphs, which are finite.

References

[1] Charlton Barreto & et al. (2007): *Web Services Business Process Execution Language Version 2.0*. `https://www.oasis-open.org/committees/download.php/23964/wsbpel-v2.0-primer.htm`.

[2] Davide Basile, Pierpaolo Degano, Gian-Luigi Ferrari & Emilio Tuosto (2016): *Relating two automata-based models of orchestration and choreography*. *JLAMP* 85(3), pp. 425 – 446.

[3] Laura Bocchi, Hernán C. Melgratti & Emilio Tuosto (2014): *Resolving Non-determinism in Choreographies*. In: *ESOP*, pp. 493–512.

[4] Gérard Boudol & Ilaria Castellani (1988): *Permutation of transitions: an event structure semantics for CCS and SCCS*. In J.W. de Bakker, W.-P.

de Roever & G. Rozenberg, editors: *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, *Lecture Notes in Computer Science* 354, Springer-Verlag, pp. 411–427.

[5] Daniel Brand & Pitro Zafiropulo (1983): *On Communicating Finite-State Machines. Journal of the ACM* 30(2), pp. 323–342.

[6] Mario Bravetti & Gianluigi Zavattaro (2007): *Towards a Unifying Theory for Choreography Conformance and Contract Compliance.* In: *Proceedings of the 6th International Conference on Software Composition*, SC'07, Springer-Verlag, pp. 34–50.

[7] Marco Carbone, Kohei Honda & Nobuko Yoshida (2007): *A Calculus of Global Interaction based on Session Types. Electronic Notes in Theoretical Computer Science* 171(3), pp. 127 – 151.

[8] Marco Carbone & Fabrizio Montesi (2013): *Deadlock-freedom-by-design: multiparty asynchronous global programming.* In: *POPL13*, pp. 263–274.

[9] Giuseppe Castagna, Mariangiola Dezani-Ciancaglini & Luca Padovani (2012): *On Global Types and Multi-Party Session. LMCS* 8(1).

[10] Gérard Cécé & Alain Finkel (2005): *Verification of programs with half-duplex communication. I&C* 202(2), pp. 166–190.

[11] Mario Coppo, Mariangiola Dezani-Ciancaglini, Nobuko Yoshida & Luca Padovani (2016): *Global progress for dynamically interleaved multiparty sessions. Mathematical Structures in Computer Science* 26(2), pp. 238–302.

[12] Mila Dalla Preda, Maurizio Gabbrielli, Saverio Giallorenzo, Ivan Lanese & Mauro Jacopo (2015): *Dynamic Choreographies - Safe Runtime Updates of Distributed Applications.* In: *COORDINATION 2015*, pp. 67–82.

[13] Mila Dalla Preda, Maurizio Gabbrielli, Saverio Giallorenzo, Ivan Lanese & Mauro Jacopo (2017): *Dynamic Choreographies - Safe Runtime Updates of Distributed Applications. Logical methods in Computer Science* 13(2).

[14] Pierpaolo Degano & Ugo Montanari (1987): *A Model for Distributed Systems Based on Graph Rewriting. Journal of the ACM* 34(2), pp. 411–449.

[15] Pierre-Malo Deniélou & Nobuko Yoshida (2012): *Multiparty Session Types Meet Communicating Automata.* In: *ESOP*, pp. 194–213.

[16] Haim Gaifman & Vaughan R Pratt (1987): *Partial order models of concurrency and the computation of functions.* In: *LICS*, pp. 72–85.

[17] Object Management Group: *Business Process Model and Notation.* `http://www.bpmn.org`.

[18] Roberto Guanciale & Emilio Tuosto (2016): *An Abstract Semantics of the Global View of Choreographies*. In: *Proceedings 9th Interaction and Concurrency Experience, ICE 2016, Heraklion, Greece, 8-9 June 2016.*, pp. 67–82.

[19] Kohei Honda, Nobuko Yoshida & Marco Carbone (2008): *Multiparty asynchronous session types*. In George C. Necula & Philip Wadler, editors: *Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2008, San Francisco, California, USA, January 7-12, 2008*, ACM, pp. 273–284, doi:10.1145/1328438.1328472. Available at `http://doi.acm.org/10.1145/1328438.1328472`.

[20] Kohei Honda, Nobuko Yoshida & Marco Carbone (2008): *Multiparty asynchronous session types*. In: *POPL*, pp. 273–284.

[21] Kohei Honda, Nobuko Yoshida & Marco Carbone (2016): *Multiparty Asynchronous Session Types*. *Journal of the ACM* 63(1), pp. 9:1–9:67. Extended version of a paper presented at POPL08.

[22] Joost-Pieter Katoen & Lennard Lambert (1998): *Pomsets for message sequence charts*. *Formale Beschreibungstechniken für Verteilte Systeme*, pp. 197–208.

[23] Nickolas Kavantzas, Davide Burdett, Gregory Ritzinger, Tony Fletcher & Yves Lafon (2004): *Web Services Choreography Description Language Version 1.0*. `http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217`.

[24] Ivan Lanese, Claudio Guidi, Fabrizio Montesi & Gianluigi Zavattaro (2008): *Bridging the Gap Between Interaction- and Process-Oriented Choreographies*. In: *Proceedings of the 2008 Sixth IEEE International Conference on Software Engineering and Formal Methods*, SEFM '08, IEEE Computer Society, pp. 323–332.

[25] Julien Lange & Emilio Tuosto: *ChorGram*. `https://bitbucket.org/emlio_tuosto/chorgram/wiki/Home`.

[26] Julien Lange, Emilio Tuosto & Nobuko Yoshida (2015): *From Communicating Machines to Graphical Choreographies*. In: *POPL15*, pp. 221–232.

[27] Julien Lange, Emilio Tuosto & Nobuko Yoshida (2017): *A tool for choreography-based analysis of message-passing software*. ACM. To appear. Available at `http://www.cs.le.ac.uk/~et52/chorgram_betty_ch.pdf`.

[28] James Lewis & Martin Fowler (2014): *Microservices: a definition of this new architectural term*. `http://martinfowler.com/articles/microservices.html`.

[29] Vaughan Pratt (1986): *Modeling concurrency with partial orders*. *International Journal of Parallel Programming* 15(1), pp. 33–71.

[30] Zongyan Qiu, Xiangpeng Zhao, Chao Cai & Hongli Yang (2007): *Towards the Theoretical Foundation of Choreography.* In: *WWW07*, ACM, pp. 973–982.

Appendix A. Proofs of consistency of resolutions

Lemma 1. *Participant* A *cannot be both passive and active in* $\mathsf{G}_1 + \mathsf{G}_2$.

Proof. We show the proof for A active (the passive case is similar). Let (ϕ, ψ) a A-prefix map such that p is active at the point of divergence $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\tilde{l}_1, \tilde{l}_2)$, that is (cf. Definition 13)

- both $\tilde{l}_1$ and $\tilde{l}_2$ contain only events in $\mathcal{L}^!$
- also $\tilde{l}_1 \cap \tilde{l}_2 = \emptyset$, $\tilde{l}_1 \neq \emptyset$, and $\tilde{l}_2 \neq \emptyset$
- ϕ is a bijection from partitions of $[\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{A}}$ to partitions of $[\![\mathsf{G}_2]\!]\!\downarrow_{\mathsf{A}}$

Clearly A cannot be passive for (ϕ, ψ) since $\tilde{l}_1 \cup \tilde{l}_2 \not\subseteq \mathcal{L}^?$. Consider an A-prefix map (ϕ', ψ') such that $\mathsf{div}_{\mathsf{A}}^{\phi',\psi'}(\mathsf{G}_1, \mathsf{G}_2) = (\tilde{l}'_1, \tilde{l}'_2) \neq (\tilde{l}_1, \tilde{l}_2)$. Then for every $\mathcal{R}' \in \operatorname{dom}\phi'$ there exists $\mathcal{R} \in \operatorname{dom}\phi$ such that $\psi'(\mathcal{R}')$ is a prefix of $\psi(\mathcal{R})$. Indeed, let $\mathcal{R} \in \operatorname{dom}\phi$ and $\mathcal{R}' \in \operatorname{dom}\phi'$ such that $\mathcal{R} \cap \mathcal{R}' \neq \emptyset$ (such $\mathcal{R}$ and $\mathcal{R}'$ exist because $\operatorname{dom}\phi$ and $\operatorname{dom}\phi'$ are partitions of $[\![\mathsf{G}_1]\!]\!\downarrow_{\mathsf{A}}$). Note that $\psi(\mathcal{R})$ and $\psi'(\mathcal{R}')$ are both prefixes of each pomset in $\mathcal{R} \cap \mathcal{R}'$ (by Definition 11). Suppose now that $\psi'(\mathcal{R}')$ is not a prefix of $\psi(\mathcal{R})$, then $l \in \tilde{l}_1$ for the label l of each minimal event e in $\psi'(\mathcal{R}')$ and not in $\psi(\mathcal{R})$ (by construction). We distinguish two cases:

- If $\phi(\mathcal{R}) \cap \phi'(\mathcal{R}') \neq \emptyset$ then both $\psi(\mathcal{R})$ and $\psi'(\mathcal{R}')$ are prefixes of each $r \in \phi(\mathcal{R}) \cap \phi'(\mathcal{R}')$; then for every minimal e in $\psi'(\mathcal{R}')$ and not in $\psi(\mathcal{R})$ there is an event in each pomset r with the same label l, therefore $l \in \tilde{l}_2$.
- If $\phi(\mathcal{R}) \cap \phi'(\mathcal{R}') = \emptyset$, let $r \in \phi'(\mathcal{R}')$ and let $\bar{\mathcal{R}} \in img(\phi)$ such that $r \in \bar{\mathcal{R}}$. If the pomset $\psi'(\mathcal{R}')$ is not a prefix of $\psi(\phi^{-1}(\bar{\mathcal{R}}))$ then there is no event in $\psi(\phi^{-1}(\bar{\mathcal{R}}))$ labelled by l then $l \in \tilde{l}_2$.

In either cases $l \in \tilde{l}_2$ contradicts the hypothesis that $\tilde{l}_1 \cap \tilde{l}_2 = \emptyset$ of Definition 13. Hence $\psi'(\mathcal{R}')$ is a prefix of $\psi(\mathcal{R})$ and A cannot be passive at the branching point of (ϕ', ψ'). □

Figure A.14 summarises the crucial steps to prove Theorems 1 and 2. We first define *delayed-choice machines* in Definition 18, which are finite-state automata built from set of pomsets. Delayed-choice machines built from the semantics of choreographies are deterministic (cf. Lemma 8) and language equivalent to the set of pomsets of the semantics (cf. Lemma 7). This implies that the delayed-choice machine $M([\![\mathsf{G}]\!])$ built from the pomset semantics of a g-choreography G yields an alternative characterization of the language of G. Although $M([\![\mathsf{G}]\!])$ is "global" (it contains transitions of all the participants in G), delayed-choice machines can also be obtained for single participants and hence be seen as CFSMs. In fact, when taking the delayed-choice machine of pomsets projected on a participant p one gets an A-local CFSM. Also, Lemma 9 shows that for a participant A the CMFS obtained by projection $(\mathsf{G}\downarrow_{\mathsf{A}})$ and the delayed-choice machine $(M([\![\mathsf{G}]\!]\!\downarrow_{\mathsf{A}}))$ are language equivalent. Thus, due determinisation, Lemma 10 shows that $\Delta(\mathsf{G}\downarrow_{\mathsf{A}})$

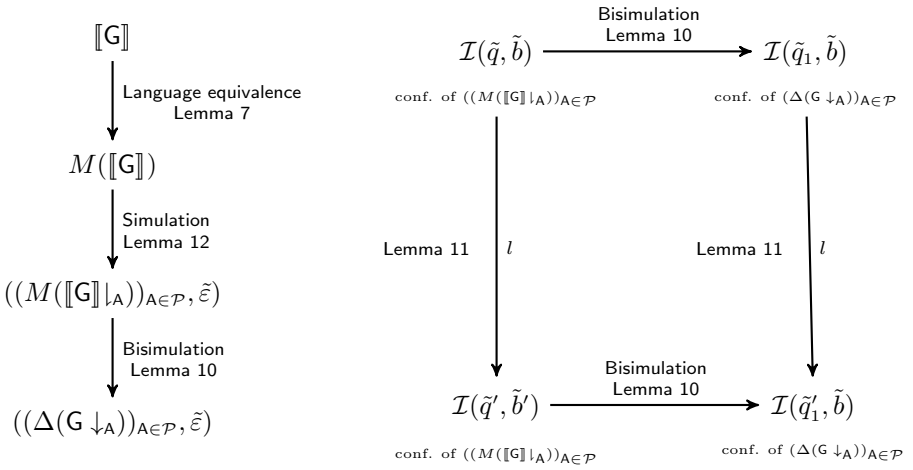


Figure A.14: Proof strategy

and $M(\llbracket \mathsf{G} \rrbracket \downarrow_{\mathsf{A}})$ are bisimilar. In practice, we can use the A-local delayed-choice machine in place of the A's CFMS machine obtained using the projection.

To prove Theorem 1 we introduce a property, called causality invariant, which guarantees that the communications among the machines implement the inter-participant causal dependencies of the choreography and that all participants agree on the selected branches of choices. A configuration that satisfies the invariant is not a deadlock. Lemma 11 shows that all reachable configurations of the system obtained by composing the local delayed-choice machines is causally invariant. This property can be transferred to the system consisting of the CMFS projections using the bisimulation.

Finally, Lemma 12 shows that the system obtained by composing the local delayed-choice machines is simulated by the global delayed-choice machine. Which together with Lemma 10 and Lemma 7 guarantees that the language of the system consisting of the CMFS projections is a subset of the language of the choreography (Theorem 2).

We start associating associating auxiliary automata to pomsets.

Definition 17. *Given a pomset r, the r-automaton is the finite-state automaton $M(r) = (2^{\mathcal{E}_r}, \emptyset, 2^{\mathcal{E}_r}, \rightarrow)$, where $q \xrightarrow{l} q'$ iff exists $e \in \mathcal{E}_r \setminus q$ such that $q' = q \cup \{e\}$, $\lambda_r(e) = l$, and for all $e' \leq_r e \wedge e' \neq e$ holds $e' \in q$.*

Note that, in Definition 17, all the states of $M(r)$ are accepting. By disregarding the accepting states, such automata can be use used as CFSMs. Also, it is a simple observation that, for any participant A, $M(r\downarrow_{\mathsf{A}})$ is A-local.

The construction r-automaton Definition 17 yields a different characterization of the language of r (hereafter, $\mathbb{L}_{M(r)}$ is the language accepted an r-automaton $M(r)$). Given a CFSM M, $\mathbb{L}_M$ is the language accepted by the finite-state automaton corresponding to M when all states are of M are accepting states.

Lemma 5. *If r is a pomset then $\mathbb{L}_{M(r)} = \mathbb{L}_r$*

Proof. The proof is straightforward by induction on the length of the words. It uses the fact that $\emptyset \xrightarrow{\omega} q'$ iff ω is a permutation of the labels of events in q'. □

Lemma 6. *Let $\mathsf{G} \in \mathcal{G}$ with $[\![\mathsf{G}]\!] \neq \bot$ and $r \in [\![\mathsf{G}]\!]$ then $M(r)$ is deterministic.*

Proof. By induction on the syntax of G. The base cases are trivial, since they produce machines with zero or one transition. For sequential composition, we notice that $\mathsf{seq}(r, r')$ ensures that all transitions involving events of r' occurs after transitions involving events of r. The proof is trivial for non-deterministic choice, since the semantics consists of the union of the pomsets of each branch. For parallel composition $\mathsf{G}|\mathsf{G}'$, non-determinism can only be introduced by the interleaving of two threads that send or receive the same message. However, $wf(\mathsf{G}, \mathsf{G}')$ ensures that this can not happen. □

Definition 18. *Let $\mathcal{R} = \{r_1, \ldots, r_n\}$ be a set of pomsets and, for $1 \leq i \leq n$, $M(r_i) = (Q_i, \emptyset, Q_i, \rightarrow_i)$ be the r_i-automaton of $r_i \in \mathcal{R}$ and let $Q = \prod_{1 \leq i \leq n}(Q_i \cup \{\bot\})$. The* delayed-choice machine *$M(\mathcal{R})$ of $\mathcal{R}$ is the automaton $M(\mathcal{R}) = (Q, (\emptyset)_{1 \leq i \leq n}, Q, \rightarrow)$ where $(q_1, \ldots, q_n) \xrightarrow{l} (q'_1, \ldots, q'_n)$ iff*

- *there exists $1 \leq i \leq n$ such that $q_i \xrightarrow{l}_i q'_i$ and*
- *for all $1 \leq i \leq n$, $q'_i = \begin{cases} q & \text{if } q_i \xrightarrow{l}_i q \\ \bot & \text{otherwise} \end{cases}$.*

Hereafter, for each $r, r' \in \mathcal{R}$, we implicitly assume that $\mathcal{E}_r \cap \mathcal{E}_{r'} = \emptyset$ when considering $M(\mathcal{R})$.

Recalling that sets of pomsets are used to give semantics the non-deterministic composition (choice) of g-choreographies, we note a key property guaranteed by Definition 18: a transition of a delayed-choice machine "forces" all branches to progress together when they can offer the same interactions.

Lemma 7. *If $\mathcal{R}$ is a set of pomset then $\mathbb{L}_{M(\mathcal{R})} = \mathbb{L}_{\mathcal{R}}$*

Proof. The proof directly follows from Lemma 5. □

Lemma 8. *If $\mathsf{G} \in \mathcal{G}$ with $[\![\mathsf{G}]\!] \neq \bot$ then $M([\![\mathsf{G}]\!])$ is deterministic. Also, if A is a participant then $M([\![\mathsf{G}]\!]\!\downarrow_{\mathsf{A}})$ is A-local and deterministic.*

Proof. The proof directly follows from the definition of delayed-choice machine and Lemma 6 □

Lemma 9. *If A is a participant of $\mathsf{G} \in \mathcal{G}$ with $[\![\mathsf{G}]\!] \neq \bot$ then $\mathbb{L}_{\mathsf{G}\downarrow_{\mathsf{A}}} = \mathbb{L}_{M([\![\mathsf{G}]\!]\downarrow_{\mathsf{A}})}$.*

Proof. By induction on the syntax of G.
Case $\mathsf{G} = \mathbf{0}$. $\mathbb{L}_{\mathsf{G}\downarrow_{\mathsf{A}}} = \mathbb{L}_{M(\mathcal{R})} = \epsilon$ by construction.
Case $\mathsf{G} = \mathsf{C} \xrightarrow{\mathsf{m}} \mathsf{B}$. If $\mathsf{A} \neq \mathsf{C}$ and $\mathsf{A} \neq \mathsf{B}$ then, as in the previous case, both languages contain only the empty word. If $\mathsf{A} = \mathsf{C}$ then $\mathbb{L}_{\mathsf{G}\downarrow_{\mathsf{A}}} = \mathbb{L}_{M(\mathcal{R})} =$

$\{\epsilon, \mathsf{AB}!m\}$ by construction. If $\mathsf{A} = \mathsf{B}$ then, by construction, $\mathbb{L}_{\mathsf{G}\downarrow_{\mathsf{A}}} = \mathbb{L}_{M(\mathcal{R})} = \{\epsilon, \mathsf{BA}?m\}$.

Case $\mathsf{G} = \mathsf{G}_1|\mathsf{G}_2$**.** Let $\_ \sqcup \_$ be the shuffling operator of formal languages, then $\mathbb{L}_{\mathsf{G}\downarrow_{\mathsf{A}}} = \{\omega_1 \sqcup \omega_2 \mid \omega_1 \in \mathbb{L}_{\mathsf{G}_1\downarrow_{\mathsf{A}}} \wedge \omega_2 \in \mathbb{L}_{\mathsf{G}_2\downarrow_{\mathsf{A}}}\}$, and $\mathbb{L}_{[\![\mathsf{G}]\!]_{\mathsf{A}}} = \{\omega_1 \sqcup \omega_2 \mid \omega_1 \in \mathbb{L}_{M([\![\mathsf{G}_1]\!]_{\mathsf{A}})} \wedge \omega_2 \in \mathbb{L}_{M([\![\mathsf{G}_2]\!]_{\mathsf{A}})}$. Then the thesis directly follows by inductive hypothesis, since $\mathbb{L}_{\mathsf{G}_h\downarrow_{\mathsf{A}}} = \mathbb{L}_{M([\![\mathsf{G}_h]\!]_{\mathsf{A}})}$ for $h \in \{1, 2\}$.

Case $\mathsf{G} = \mathsf{G}_1; \mathsf{G}_2$**.** We use the fact that $\{\mathsf{seq}(r_1, r_2) \mid (r_1, r_2) \in [\![\mathsf{G}_1]\!] \times [\![\mathsf{G}_2]\!]\}\lfloor_{\mathsf{A}} = \{\mathsf{seq}(r_1, r_2) \mid (r_1, r_2) \in ([\![\mathsf{G}_1]\!]\lfloor_{\mathsf{A}}) \times ([\![\mathsf{G}_2]\!]\lfloor_{\mathsf{A}})\}$. and that, by inductive hypothesis, $\mathbb{L}_{\mathsf{G}_h\downarrow_{\mathsf{A}}} = \mathbb{L}_{M([\![\mathsf{G}_h]\!]_{\mathsf{A}})}$ for $h \in \{1, 2\}$. Then the thesis follows by observing that $\mathbb{L}_{\mathsf{G}\downarrow_{\mathsf{A}}} = \mathbb{L}_{\mathsf{G}_1\downarrow_{\mathsf{A}}} \cdot \mathbb{L}_{\mathsf{G}_2\downarrow_{\mathsf{A}}} = \mathbb{L}_{M([\![\mathsf{G}_1]\!]_{\mathsf{A}})} \cdot \mathbb{L}_{M([\![\mathsf{G}_2]\!]_{\mathsf{A}})} = \mathbb{L}_{M([\![\mathsf{G}]\!]_{\mathsf{A}})}$ where $\_ \cdot \_$ is the usual operation of language concatenation.

Case $\mathsf{G} = \mathsf{G}_1 + \mathsf{G}_2$**.** The proof is similar to the previous case noting that for both machines the language is equal to the union of languages of the machines obtained by G_1 and G_2. □

Lemma 10. *For each participant* A *of* $\mathsf{G} \in \mathcal{G}$ *with* $[\![\mathsf{G}]\!] \neq \bot$*,* $\Delta(\mathsf{G}\downarrow_{\mathsf{A}})$ *is bisimilar to* $M([\![\mathsf{G}]\!]\lfloor_{\mathsf{A}})$*.*

Proof. The proof follows from Lemmas 8 and 9 and the fact that language equivalence implies bisimilarity for deterministic finite automata. □

In the following we use tuples as functions; for example, $\tilde{q}(r)$ denotes the component q_r of a tuple $\tilde{q} = (q_r)_{r\in\mathcal{R}}$ on a set of pomsets $\mathcal{R}$.

We now introduce a property, dubbed *causality invariant*, that intuitively guarantees that (1) the machine of each participant executes the interactions of a choice uniformly across the branches of the choice; (2) inputs can only be executed after their corresponding outputs; (3) buffers contains messages that have been sent but not consumed; (4) all participants agree on the branch selected. Since the local choice taken by the active participant is communicated asynchronously (and using different messages for each passive participant), some participants can be unaware of the branch taken. Hereafter, for a set of pomsets $\mathcal{R}$, a pomset $r \in \mathcal{R}$, and a configuration $s = \langle \tilde{q} \ ; \ \tilde{b} \rangle$ of $(M(\mathcal{R}\lfloor_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$, we say that the pomset r has been discarded, and we write $\tilde{q} \not\triangleright r$, iff there exists $\mathsf{A} \in \mathcal{P}$ such that $\tilde{q}(\mathsf{A})(r\lfloor_{\mathsf{A}}) = \bot$. We also write $\tilde{q} \triangleright r$ if the pomset has not been discarded.

Definition 19. *Given a set of pomsets* $\mathcal{R}$*, a configuration* $s = \langle \tilde{q} \ ; \ \tilde{b} \rangle$ *of* $(M(\mathcal{R}\lfloor_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$ *is* causally invariant *iff*

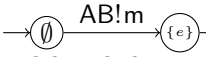
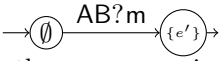
1. *for all* $\mathsf{A} \in \mathcal{P}$ *and* $r, r' \in \mathcal{R}$ *if* $\tilde{q} \triangleright r$ *and* $\tilde{q} \triangleright r'$ *then there is a label- and order-preserving bijection between* $\tilde{q}(\mathsf{A})(r\lfloor_{\mathsf{A}})$ *and* $\tilde{q}(\mathsf{A})(r'\lfloor_{\mathsf{A}})$*.*
2. *for all* $\mathsf{A} \neq \mathsf{B}$*, for every pomsets* $r \in \mathcal{R}$ *if* $\tilde{q} \triangleright r$ *then, for every output event* $e \in r\lfloor_{\mathsf{A}}$ *and input event* $e' \in r\lfloor_{\mathsf{B}}$*, if* e *is an immediate predecessor of* e' *with respect to* $\leq_r$*, then* $e' \in \tilde{q}(\mathsf{B})(r\lfloor_{\mathsf{B}}) \Rightarrow e \in \tilde{q}(\mathsf{A})(r\lfloor_{\mathsf{A}})$
3. *for all* $\mathsf{A} \neq \mathsf{B}$*, for every pomsets* $r \in \mathcal{R}$ *if* $\tilde{q} \triangleright r$ *then* $\tilde{b}(\mathsf{AB})$ *is a permutation of* $(\mathsf{m})_{\lambda r(e)=\mathsf{AB}!\mathsf{m}, e\in\tilde{e}}$ *where* $\tilde{e}$ *is the set of output events of* $\tilde{q}(\mathsf{A})(r\lfloor_{\mathsf{A}})$ *whose immediate input successor in* $r\lfloor_{\mathsf{B}}$ *is not in* $\tilde{q}(\mathsf{B})(r\lfloor_{\mathsf{B}})$*.*
4. *there exists* $r \in \mathcal{R}$ *such that* $\tilde{q} \triangleright r$

We write $\mathcal{I}(s)$ when s is causally invariant.

Lemma 11. *If $\mathsf{G} \in \mathcal{G}$ with $[\![\mathsf{G}]\!] \neq \bot$ then $\mathcal{I}(s)$ for all reachable configurations of $S = (M([\![\mathsf{G}]\!]\lfloor_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$.*

Proof. The proof is by induction on the syntax of G.
Case $\mathsf{G} = \mathbf{0}$. We have that the only reachable configuration of S is the initial one which consists of a tuple of empty sets and empty buffers and therefore it is trivially causally consistent.
Case $\mathsf{G} = \mathsf{i}\colon \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$. We have $[\![\mathsf{G}]\!] = \{r\}$ where

$$\leq_r = \begin{array}{c} \circlearrowright \\ e \longrightarrow e' \\ \circlearrowleft \end{array} \quad \text{with} \quad \lambda_r = \begin{cases} e \mapsto \mathsf{AB!m} \\ e' \mapsto \mathsf{AB?m} \end{cases}$$

Hence, $M([\![\mathsf{G}]\!]\lfloor_{\mathsf{A}}) =$  and $M([\![\mathsf{G}]\!]\lfloor_{\mathsf{B}}) =$ 
which implies that the delayed-choice machine of G produces the message m in $M([\![\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}]\!]\lfloor_{\mathsf{A}})$ and consumes the same message in $M([\![\mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}]\!]\lfloor_{\mathsf{B}})$ going through causally invariant configurations only.

In all the remaining cases, for $h \in \{1, 2\}$, we let $S_h = (M([\![\mathsf{G}_h]\!]\lfloor_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$, and let $s = (\tilde{q}, \tilde{b})$ and $s' = (\tilde{q}', \tilde{b}')$. Before continuing with the proof it is convenient to introduce the following notation; given a map f from pomsets to sets of events, we define

$$\partial_l(f) = r \mapsto \begin{cases} f(r) \cup \{e\} & \text{if } e \in \mathcal{E}_r \setminus f(r) \wedge \lambda_r(e) = l \wedge \forall e' \leq_r e \colon e' \in f(r) \\ \bot & \text{otherwise} \end{cases}$$

Case $\mathsf{G} = \mathsf{G}_1|\mathsf{G}_2$. The proof consists of the following steps:

1. decompose s into two configurations s_1 of S_1 and s_2 of S_2 such that $\mathcal{I}(s_1)$ and $\mathcal{I}(s_2)$ hold
2. from $s \stackrel{l}{\Rightarrow} s'$, exhibit $s_h \stackrel{l}{\Rightarrow} s'_h$ with $\mathcal{I}(s'_1)$ for a $h \in \{1, 2\}$, say for $h = 1$
3. show that s'_1 and s_2 is a decomposition of s' and that $\mathcal{I}(s')$ holds.

By definition, $[\![\mathsf{G}]\!] = \{\mathsf{par}(r_1, r_2) \mid (r_1, r_2) \in [\![\mathsf{G}_1]\!] \times [\![\mathsf{G}_2]\!]\}$, hence $[\![\mathsf{G}]\!]\lfloor_{\mathsf{A}} = \{\mathsf{par}(r_1, r_2) \mid (r_1, r_2) \in [\![\mathsf{G}_1]\!]\lfloor_{\mathsf{A}} \times [\![\mathsf{G}_2]\!]\lfloor_{\mathsf{A}}\}$, for each participant A.

Regarding step (1), for $h \in \{1, 2\}$ we let $s_h = \langle \tilde{q}_h \ ; \ \tilde{b}_h \rangle$ where, for $r \in [\![\mathsf{G}_1]\!]$ and $\mathsf{A} \in \mathcal{P}$

$$\tilde{q}_1(\mathsf{A})(r\lfloor_{\mathsf{A}}) = \begin{cases} \bigcup_{r' \in [\![\mathsf{G}_2]\!]} \{e \mid (e, 1) \in \tilde{q}(\mathsf{A})(\mathsf{par}(r, r')\lfloor_{\mathsf{A}})\} & \exists r' \in [\![\mathsf{G}_2]\!] \colon \tilde{q}(\mathsf{A})(\mathsf{par}(r, r')\lfloor_{\mathsf{A}}) \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

and, for $r \in [\![\mathsf{G}_2]\!]$ and $\mathsf{A} \in \mathcal{P}$

$$\tilde{q}_2(\mathsf{A})(r\lfloor_{\mathsf{A}}) = \begin{cases} \bigcup_{r' \in [\![\mathsf{G}_1]\!]} \{e \mid (e, 2) \in \tilde{q}(\mathsf{A})(\mathsf{par}(r', r)\lfloor_{\mathsf{A}})\} & \exists r' \in [\![\mathsf{G}_1]\!] \colon \tilde{q}(\mathsf{A})(\mathsf{par}(r', r)\lfloor_{\mathsf{A}}) \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

Moreover, for all $\mathsf{AB} \in \mathcal{C}$ and $h \in \{1,2\}$, let $\tilde{b}_h(\mathsf{AB})$ be the buffer obtained by removing from $\tilde{b}(\mathsf{AB})$ all messages that do not occur in G_h. Note that

- for all $r_1 \in [\![\mathsf{G}_1]\!]$, $r_2 \in [\![\mathsf{G}_2]\!]$, and $\mathsf{A} \in \mathcal{P}$, we have $\tilde{q}(\mathsf{A})(\mathsf{par}(r_1, r_2)\!\upharpoonright_{\mathsf{A}}) = \tilde{q}_1(\mathsf{A})(r_1\!\upharpoonright_{\mathsf{A}}) \uplus \tilde{q}_2(\mathsf{A})(r_2\!\upharpoonright_{\mathsf{A}})$, for $\mathsf{A} \in \mathcal{P}$, $r_1 \in [\![\mathsf{G}_1]\!]$, and $r_2 \in [\![\mathsf{G}_2]\!]$ (by the definition of $\mathsf{par}(\_,\_)$)
- due to $wf(\mathsf{G}_1, \mathsf{G}_2)$, G_1 and G_2 contain distinct messages, thus for all $\mathsf{AB} \in \mathcal{C}$ a message of $\tilde{b}(\mathsf{AB})$ cannot be in both $\tilde{b}_1(\mathsf{AB})$ and $\tilde{b}_h(\mathsf{AB})$, and $\tilde{b}(\mathsf{AB}) \in \tilde{b_1}(\mathsf{AB}) \sqcup \tilde{b_2}(\mathsf{AB})$.

Also, for $h \in \{1,2\}$, $s_h = \langle \tilde{q}_h \; ; \; \tilde{b}_h \rangle$ is a reachable configuration of $(M([\![\mathsf{G}_h]\!]\!\upharpoonright_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$ otherwise s would not be a reachable configuration of S. Hence, $\mathcal{I}(s_1)$ and $\mathcal{I}(s_2)$ hold by the inductive hypothesis.

For (2), we rely on the fact that the pending messages those that have been sent but not been consumed. Since $wf(\mathsf{G}_1, \mathsf{G}_2)$ guarantees that l either come from G_1 or from G_2, without loss of generality we can consider the case only where l is generated in G_1 (the other case is analogous). Let $\mathsf{A} = \mathsf{sbj}(l)$, if $s \stackrel{l}{\Rightarrow} s'$ then $\tilde{q}(\mathsf{A}) \stackrel{l}{\rightarrow} \tilde{q}'(\mathsf{A})$ and there is an event e of a pomset $r \in [\![\mathsf{G}_1]\!]$ such that $e \notin \tilde{q}_1(\mathsf{A})(r)$ and $\lambda_r(e) = l$. Therefore, $\tilde{q}_1(\mathsf{A}) \stackrel{l}{\rightarrow} \tilde{\partial}_l(\tilde{q}_1(\mathsf{A}))$.

We consider the case $l = \mathsf{AB!m}$ first. By Definitions 17 and 18.

$$\tilde{q}' = \tilde{q}[\mathsf{A} \mapsto \partial_l(\tilde{q}(\mathsf{A}))] \qquad \text{and} \qquad \tilde{b}' = \tilde{b}[\mathsf{AB} \mapsto \tilde{b}(\mathsf{AB}) \cdot \mathsf{m}]$$

Also, $s_1 \stackrel{l}{\Rightarrow} s_1'$ and $s_1' = \langle \tilde{q}_1' \; ; \; \tilde{b}_1' \rangle$ where

$$\tilde{q}_1' = \tilde{q}_1[\mathsf{A} \mapsto \partial_l(\tilde{q}_1(\mathsf{A}))] \qquad \text{and} \qquad \tilde{b}_1' = \tilde{b}_1[\mathsf{AB} \mapsto \tilde{b}_1(\mathsf{AB}) \cdot \mathsf{m}]$$

Due to the inductive hypotheses configuration s_1' is causally invariant. Also, for all $\mathsf{C} \neq \mathsf{A} \in \mathcal{P}$ and all $r = \mathsf{par}(r_1, r_2) \in [\![\mathsf{G}]\!]\!\upharpoonright_{\mathsf{C}}$, $\tilde{q}'(\mathsf{C})(r) = \tilde{q}_1'(\mathsf{C})(r_1) \uplus \tilde{q}_2(\mathsf{A})(r_2)$ and, for all $\mathsf{CD} \in \mathcal{C}$, $\tilde{b}'(\mathsf{CD}) = \tilde{b}_1'(\mathsf{CD}) \sqcup \tilde{b}_2(\mathsf{CD})$.

Finally for (3), we verify that s' satisfies the conditions of Definition 19:

- the first condition holds because $\tilde{q}'$ differs from $\tilde{q}$ only on A and for each $r = \mathsf{par}(r_1, r_2) \in [\![\mathsf{G}_1]\!]\!\upharpoonright_{\mathsf{A}}$ and $r' = \mathsf{par}(r_1', r_2') \in [\![\mathsf{G}_2]\!]\!\upharpoonright_{\mathsf{A}}$ with $\tilde{q}'(\mathsf{A})(r) \neq \bot$ and $\tilde{q}'(\mathsf{A})(r') \neq \bot$ we have that $\tilde{q}'(\mathsf{A})(r) = q(\mathsf{A})(r) \cup \{(e,1)\}$ and $\tilde{q}'(\mathsf{A})(r') = q(\mathsf{A})(r') \cup \{(e',1)\}$ for some $e \in r_1\!\upharpoonright_{\mathsf{A}}$ and $e' \in r_1'\!\upharpoonright_{\mathsf{A}}$ such that $\lambda_{r_1}(e) = \lambda_{r_1'}(e') = l$. Therefore, we can extend the bijection between $\tilde{q}(\mathsf{A})(r)$ and $\tilde{q}(\mathsf{A})(r')$ by mapping the $(e,1)$ to $(e',1)$.
- for the second condition we have to check that for all $\mathsf{A} \neq \mathsf{C}$ and all $r = \mathsf{par}(r_1, r_2) \in [\![\mathsf{G}]\!]$, if $\tilde{q}'(\mathsf{A})(r\!\upharpoonright_{\mathsf{A}}) \neq \bot$ and $\tilde{q}'(\mathsf{C})(r\!\upharpoonright_{\mathsf{C}}) \neq \bot$ then, for every output event $(e,1) \in \mathcal{E}_{r_1\downarrow_{\mathsf{A}}} \times \mathbf{1}$ and input event $(e',1) \in \mathcal{E}_{r_1\downarrow_{\mathsf{C}}} \times \mathbf{1}$, if $(e,1)$ is an immediate predecessor of $(e',1)$ with respect to $\leq_r$, then $e' \in \tilde{q}'(\mathsf{C})(r\!\upharpoonright_{\mathsf{C}}) \implies e \in \tilde{q}'(\mathsf{A})(r\!\upharpoonright_{\mathsf{A}})$ which holds otherwise $\mathcal{I}(s_1')$ would not hold

- for the third condition, take $r = \mathsf{par}(r_1, r_2) \in [\![\mathsf{G}]\!]$ such that $\tilde{q}'(\mathsf{A})(r\!\upharpoonright_{\mathsf{A}}) \neq \bot$ and $\tilde{q}'(\mathsf{C})(r\!\upharpoonright_{\mathsf{C}}) \neq \bot$; then $\tilde{b}'(\mathsf{AC})$ is a permutation of $(\mathsf{m})_{\lambda_r(e)=\mathsf{AC!m}, e\in\tilde{e}}$ where $\tilde{e}$ is the set of output events of $\tilde{q}(\mathsf{A})(r\!\upharpoonright_{\mathsf{A}})$ whose immediate input successor in $r\!\upharpoonright_{\mathsf{C}}$ is not in $\tilde{q}(\mathsf{C})(r\!\upharpoonright_{\mathsf{C}})$, again because $\mathcal{I}(s'_1)$ and $\mathcal{I}(s_2)$ hold.
- finally, there is $r = \mathsf{par}(r_1, r_2) \in [\![\mathsf{G}]\!]$ such that $\tilde{q}'(\mathsf{C})(r\!\upharpoonright_{\mathsf{C}}) \neq \bot$ for all $\mathsf{C} \in \mathcal{P}$ because, if a participant $\mathsf{C} \in \mathcal{P}$ discarded a branch $r_1 \in [\![\mathsf{G}_1]\!]$ in $\tilde{q}'_1$ (i.e. $\tilde{q}'_1(\mathsf{C})(r_1\!\upharpoonright_{\mathsf{C}}) \neq \tilde{q}_1(\mathsf{C})(r_1\!\upharpoonright_{\mathsf{C}})$ and $\tilde{q}'_1(\mathsf{C})(r_1\!\upharpoonright_{\mathsf{C}}) = \bot$) then it has discarded all corresponding branches in $\tilde{q}'$ (i.e. for all $r_2 \in [\![\mathsf{G}_2]\!]$, $\tilde{q}'(\mathsf{C})(\mathsf{par}(r_1, r_2)\!\upharpoonright_{\mathsf{C}}) = \bot$).

Hence $\mathcal{I}(s')$.

We consider the case $l = \mathsf{BA?m}$.

$$\tilde{q}' = \tilde{q}[\mathsf{A} \mapsto \partial_l(\tilde{q}(\mathsf{A}))] \qquad \text{and} \qquad \tilde{b} = \tilde{b}'[\mathsf{BA} \mapsto \mathsf{m} \cdot \tilde{b}'(\mathsf{BA})]$$

by Definitions 17 and 18. Due to $wf(\mathsf{G}, \mathsf{G}')$ we know that m cannot be in the buffer $\tilde{b}_2(\mathsf{BA})$. In fact, messages exchanged in concurrently composed g-choreographies must be disjoint. This, and the fact that $\tilde{b}(\mathsf{BA})$ contains an interleaving of the messages of the buffers of $\tilde{b}_1(\mathsf{BA})$ and $\tilde{b}_2(\mathsf{BA})$, ensures that the message is in the head of the buffer $\tilde{b}_1(\mathsf{BA})$. Therefore $s_1 \stackrel{l}{\Rightarrow} s'_1$ and $s'_1 = \langle \tilde{q}'_1 \ ; \ \tilde{b}'_1 \rangle$ where

$$\tilde{q}'_1 = \tilde{q}_1[\mathsf{A} \mapsto \partial_l(\tilde{q}_1(\mathsf{A}))] \qquad \text{and} \qquad \tilde{b}_1 = \tilde{b}'_1[\mathsf{BA} \mapsto \mathsf{m} \cdot \tilde{b}'_1(\mathsf{BA})]$$

The proof continues similarly to the output case.

Case $\mathsf{G} = \mathsf{G}_1; \mathsf{G}_2$. By definition, $[\![\mathsf{G}]\!] = \{\mathsf{seq}(r_1, r_2) \mid (r_1, r_2) \in [\![\mathsf{G}_1]\!] \times [\![\mathsf{G}_2]\!]\}$ and, for a participant A, $[\![\mathsf{G}]\!]\!\upharpoonright_{\mathsf{A}} = \{\mathsf{seq}(r_1, r_2) \mid (r_1, r_2) \in [\![\mathsf{G}_1]\!]\!\upharpoonright_{\mathsf{A}} \times [\![\mathsf{G}_2]\!]\!\upharpoonright_{\mathsf{A}}\}$. The proof works as follows:

1. we find two causally invariant configurations s_1 of S_1 and s_2 of S_2 that correspond to s
2. we show that since $s \stackrel{l}{\Rightarrow} s'$ for one of the two configurations holds $s_h \stackrel{l}{\Rightarrow} s'_h$
3. due to the inductive hypotheses the invariant is preserved by $s_h \stackrel{l}{\Rightarrow} s'_h$
4. we show that s' corresponds to s'_1 and s'_2 and prove that s' is causally invariant.

For (1), let $h \in \{1, 2\}$ and $s_h = \langle \tilde{q}_h \ ; \ \tilde{b}_h \rangle$ where, for $r \in [\![\mathsf{G}_1]\!]$ and $\mathsf{A} \in \mathcal{P}$

$$\tilde{q}_1(\mathsf{A})(r\!\upharpoonright_{\mathsf{A}}) = \begin{cases} \bigcup_{r' \in [\![\mathsf{G}_2]\!]} \{e \mid (e, 1) \in \tilde{q}(\mathsf{A})(\mathsf{seq}(r, r')\!\upharpoonright_{\mathsf{A}})\} & \exists r' \in [\![\mathsf{G}_2]\!] : \tilde{q}(\mathsf{A})(\mathsf{seq}(r, r')\!\upharpoonright_{\mathsf{A}}) \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

and, for $r \in [\![\mathsf{G}_2]\!]$ and $\mathsf{A} \in \mathcal{P}$

$$\tilde{q}_2(\mathsf{A})(r\!\upharpoonright_{\mathsf{A}}) = \begin{cases} \bigcup_{r' \in [\![\mathsf{G}_1]\!]} \{e \mid (e, 2) \in \tilde{q}(\mathsf{A})(\mathsf{seq}(r', r)\!\upharpoonright_{\mathsf{A}})\} & \exists r' \in [\![\mathsf{G}_1]\!] : \tilde{q}(\mathsf{A})(\mathsf{seq}(r', r)\!\upharpoonright_{\mathsf{A}}) \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

Moreover, for all $\mathsf{AB} \in \mathcal{C}$, let $\mathtt{idx}(\mathsf{AB})$ be the difference between the number of output events in the image of $\tilde{q}(\mathsf{A})$ from A to B obtained from the pomsets in $[\![\mathsf{G}_1]\!]$ and the input events of B from A in the image of $\tilde{q}(\mathsf{A})$, and define

$$\tilde{b}_1(\mathsf{AB}) = \mathsf{m}_1 \cdots \mathsf{m}_{\mathtt{idx}(\mathsf{AB})} \quad \text{and} \quad \tilde{b}_2(\mathsf{AB}) = \mathsf{m}_{\mathtt{idx}(\mathsf{AB})+1} \cdots \mathsf{m}_i$$

Note that $\mathtt{idx}(\mathsf{AB})$ is between 0 and the number of messages in $\tilde{b}$ because $ws(\mathsf{G}_1, \mathsf{G}_2)$ holds by hypothesis and by the definition of $\mathsf{seq}(\mathsf{G}_1, \mathsf{G}_2)$.

- for all $\mathsf{A} \in \mathcal{P}$, $r_1 \in [\![\mathsf{G}_1]\!]$, and $r_2 \in [\![\mathsf{G}_2]\!]$, we have $\tilde{q}(\mathsf{A})(\mathsf{seq}(r_1, r_2)\!\upharpoonright_{\mathsf{A}}) = \tilde{q}_1(\mathsf{A})(r_1\!\upharpoonright_{\mathsf{A}}) \uplus \tilde{q}_2(\mathsf{A})(r_2\!\upharpoonright_{\mathsf{A}})$, (by the definition of $\mathsf{seq}(\_,\_)$)
- and $\tilde{b}(\mathsf{AB}) = \tilde{b_1}(\mathsf{AB}) \cdot \tilde{b_2}(\mathsf{AB})$ for all $\mathsf{AB} \in \mathcal{C}$.

Also, for $h \in \{1, 2\}$, $s_h = \langle \tilde{q}_h \,;\, \tilde{b}_h \rangle$ is a reachable configuration of $(M([\![\mathsf{G}_h]\!]\!\upharpoonright_{\mathsf{A}}))_{\mathsf{A} \in \mathcal{P}}$ otherwise s would not be a reachable configuration of S. Hence, $\mathcal{I}(s_1)$ and $\mathcal{I}(s_2)$ by the inductive hypothesis.

For (2), we rely on the fact that the pending messages of the buffers in a reachable configuration are those obtained by output events not matched by corresponding input events. If $s \overset{l}{\Rightarrow} s'$ then, by construction, we have $s_1 \overset{l}{\Rightarrow} s'_1$ or $s_2 \overset{l}{\Rightarrow} s'_2$. We consider the case $l = \mathsf{AB!m}$ first. If $s_1 \overset{l}{\Rightarrow} s'_1$ then there is an event e of a pomset $r \in [\![\mathsf{G}_1]\!]$ such that $e \notin \tilde{q}_1(\mathsf{A})(r)$, $\lambda_r(e) = \mathsf{AB!m}$, and $s'_1 = \langle \tilde{q}'_1 \,;\, \tilde{b}'_1 \rangle$ where

$$\tilde{q}'_1 = \tilde{q}_1[\mathsf{A} \mapsto \partial_{\mathsf{AB!m}}(\tilde{q}_1(\mathsf{A}))] \qquad \text{and} \qquad \tilde{b}'_1 = \tilde{b}_1[\mathsf{AB} \mapsto \tilde{b}_1(\mathsf{AB}) \cdot \mathsf{m}]$$

by Definitions 17 and 18. Therefore,

$$\tilde{q}' = \tilde{q}[\mathsf{A} \mapsto \partial_{\mathsf{AB!m}}(\tilde{q})] \qquad \text{and} \qquad \tilde{b}' = \tilde{b}[\mathsf{AB} \mapsto \tilde{b}(\mathsf{AB}) \cdot \mathsf{m}]$$

(again by Definitions 17 and 18) where the latter equality holds because $ws(\mathsf{G}_1, \mathsf{G}_2)$ and by definition of $\mathsf{seq}(\mathsf{G}_1, \mathsf{G}_2)$ imply that $\tilde{b}_2(\mathsf{AB})$ is empty. Finally, we verify that s' satisfy the conditions of Definition 19:

- the first condition holds because $\tilde{q}'$ differs from $\tilde{q}$ only on A and for every $r_1, r'_1 \in [\![\mathsf{G}_1]\!]\!\upharpoonright_{\mathsf{A}}$ and $r_2 \in [\![\mathsf{G}_2]\!]\!\upharpoonright_{\mathsf{A}}$ with $\tilde{q}'(\mathsf{A})(\mathsf{seq}(r_1, r_2)\!\upharpoonright_{\mathsf{A}}) \neq \bot$ and $\tilde{q}'(\mathsf{A})(\mathsf{seq}(r'_1, r_2)\!\upharpoonright_{\mathsf{A}}) \neq \bot$ we have that $\tilde{q}'(\mathsf{A})(\mathsf{seq}(r_1, r_2)\!\upharpoonright_{\mathsf{A}}) = q(\mathsf{A})(\mathsf{seq}(r_1, r_2)\!\upharpoonright_{\mathsf{A}}) \cup \{(e, 1)\}$ and $\tilde{q}'(\mathsf{A})(\mathsf{seq}(r_1, r_2)\!\upharpoonright_{\mathsf{A}}) = q(\mathsf{A})(\mathsf{seq}(r_1, r_2)\!\upharpoonright_{\mathsf{A}}) \cup \{(e', 1)\}$ for some $e \in r_1\!\upharpoonright_{\mathsf{A}}$ and $e' \in r'_1\!\upharpoonright_{\mathsf{A}}$, and $\lambda_{r_1}(e) = \lambda_{r'_1}(e') = l$. Therefore we can extend the bijection between $\tilde{q}(\mathsf{A})(\mathsf{seq}(r_1, r_2)\!\upharpoonright_{\mathsf{A}})$ and $\tilde{q}(\mathsf{A})(\mathsf{seq}(r'_1, r_2)\!\upharpoonright_{\mathsf{A}})$ by mapping the $(e, 1)$ to $(e', 1)$.
- for the second condition we have to check that for all $\mathsf{A} \neq \mathsf{C}$ and all $r = \mathsf{seq}(r_1, r_2) \in [\![\mathsf{G}]\!]$, if $\tilde{q}'(\mathsf{A})(r\!\upharpoonright_{\mathsf{A}}) \neq \bot$ and $\tilde{q}'(\mathsf{C})(r\!\upharpoonright_{\mathsf{C}}) \neq \bot$ then, for every output event $(e, 1) \in \mathcal{E}_{r_1\!\upharpoonright_{\mathsf{A}}} \times \mathbf{1}$ and input event $(e', 1) \in \mathcal{E}_{r_1\!\upharpoonright_{\mathsf{C}}} \times \mathbf{1}$, if $(e, 1)$ is an immediate predecessor of $(e', 1)$ with respect to $\leq_r$, then $e' \in \tilde{q}'(\mathsf{C})(r\!\upharpoonright_{\mathsf{C}}) \implies e \in \tilde{q}'(\mathsf{A})(r\!\upharpoonright_{\mathsf{A}})$ which holds otherwise $\mathcal{I}(s_1)$ would not hold

- for the third condition we have that, for all $r = \mathsf{seq}(r_1, r_2) \in [\![\mathsf{G}]\!]$, when $\tilde{q}'(\mathsf{A})(r\downarrow_{\mathsf{A}}) \neq \bot$ and $\tilde{q}'(\mathsf{B})(r\downarrow_{\mathsf{B}}) \neq \bot$, then $\tilde{b}'(\mathsf{AB})$ is a permutation of $(\mathsf{m})_{\lambda_r(e)=\mathsf{AB!m}, e\in\tilde{e}}$ where $\tilde{e}$ is the set of output events of $\tilde{q}(\mathsf{A})(r\downarrow_{\mathsf{A}})$ whose immediate input successor in $r\downarrow_{\mathsf{B}}$ is not in $\tilde{q}(\mathsf{B})(r\downarrow_{\mathsf{B}})$, again because $\mathcal{I}(s_1)$ holds

- finally, there is $r = \mathsf{seq}(r_1, r_2) \in [\![\mathsf{G}]\!]$ such that $\tilde{q}'(\mathsf{C})(r\downarrow_{\mathsf{C}}) \neq \bot$ for all $\mathsf{C} \in \mathcal{P}$ because, if a participant $\mathsf{C} \in \mathcal{P}$ "discarded" a branch $r_1 \in [\![\mathsf{G}_1]\!]$ in $\tilde{q}'_1$ (i.e. $\tilde{q}'_1(\mathsf{C})(r_1\downarrow_{\mathsf{C}}) \neq \tilde{q}'_1(\mathsf{C})(r_1\downarrow_{\mathsf{C}})$ and $\tilde{q}'_1(\mathsf{C})(r_1\downarrow_{\mathsf{C}}) = \bot$) then it has discarded all corresponding branches in $\tilde{q}'$ (i.e. for all $r_2 \in [\![\mathsf{G}_2]\!]$, $\tilde{q}'(\mathsf{C})(\mathsf{seq}(r_1, r_2)\downarrow_{\mathsf{C}}) = \bot$).

Hence $\mathcal{I}(s')$. We now consider the case $s_2 \xRightarrow{\mathsf{AB!m}} s'_2$

As before, there is an event e of a pomset $r \in [\![\mathsf{G}_2]\!]$ such that $e \notin \tilde{q}_2(\mathsf{A})(r)$, $\lambda_r(e) = \mathsf{AB!m}$, and $s'_2 = \langle \tilde{q}'_2 \ ; \ \tilde{b}'_2 \rangle$ where

$$\tilde{q}'_2 = \tilde{q}_2[\mathsf{A} \mapsto \partial_{\mathsf{AB!m}}(\tilde{q}_2(\mathsf{A}))] \qquad \text{and} \qquad \tilde{b}'_2 = \tilde{b}_2[\mathsf{AB} \mapsto \tilde{b}_2(\mathsf{AB}) \cdot \mathsf{m}]$$

by Definitions 17 and 18. Therefore,

$$\tilde{q}' = \tilde{q}[\mathsf{A} \mapsto \partial_{\mathsf{AB!m}}(\tilde{q})] \qquad \text{and} \qquad \tilde{b}' = \tilde{b}[\mathsf{AB} \mapsto \tilde{b}(\mathsf{AB}) \cdot \mathsf{m}]$$

(again by Definitions 17 and 18) where the latter equality holds because $ws(\mathsf{G}_1, \mathsf{G}_2)$ and by definition of $\mathsf{seq}(\mathsf{G}_1, \mathsf{G}_2)$ imply that $\tilde{b}_1(\mathsf{AB})$ is empty. Finally, reasoning as in the previous case we can verify that s' satisfy the conditions of Definition 19.

In the case of input, assume $l = \mathsf{BA?m}$. Suppose first that the input transition is performed by $s_1(\mathsf{A})$. Then, $\tilde{b}_1(\mathsf{BA}) = \mathsf{m} \cdot \omega$ for some word $\omega \in \mathcal{M}^\star$ since $\mathcal{I}(s_1)$ (otherwise $s_1 \stackrel{l}{\Rightarrow} s'_1$ would not be possible). If the input transition is performed by $s_2(\mathsf{A})$ for every $r_1 \in [\![\mathsf{G}_1]\!]$ we have $s_1(\mathsf{A})(r_1\downarrow_{\mathsf{A}}) = \mathcal{E}_{r_1\Downarrow_{\mathsf{A}}}$ because of the order imposed by the sequential composition and Definition 18. Hence, buffer $\tilde{b}_1(\mathsf{BA})$ is empty and buffer $\tilde{b}_2(\mathsf{BA})$ has message m on its top (that is, $\tilde{b}_2(\mathsf{BA}) = \mathsf{m} \cdot \omega$ for some word $\omega \in \mathcal{M}^\star$ since $\mathcal{I}(s_2)$) otherwise $s_1 \stackrel{l}{\Rightarrow} s'_1$ would not be possible.

Finally, $\mathcal{I}(s')$ follows by the fact that no event of A of G_2 can be performed before the events A of G_1 and that, for $h \in \{1, 2\}$, the messages produced by outputs of G_h are consumed by inputs of G_h.

Case $\mathsf{G} = \mathsf{G}_1 + \mathsf{G}_2$. The proof works as follows:

1. we find a causally invariant configuration s_1 of S_1 or s_2 of S_2 that corresponds to s (without loss of generality, we consider the former case only, since the other is analogous)
2. from $s \stackrel{l}{\Rightarrow} s'$ we find a transition $s_1 \stackrel{l}{\Rightarrow} s'_1$ and show that s'_1 corresponds to s' and it is causally invariant

For (1), let $h \in \{1, 2\}$ and $\mathsf{A} \in \mathcal{P}$ and $s_h = \langle \tilde{q}_h \ ; \ \tilde{b} \rangle$ where

$$\tilde{q}_h(\mathsf{A})(r\downarrow_{\mathsf{A}}) = \begin{cases} \tilde{q}(\mathsf{A})(r\downarrow_{\mathsf{A}}) & \text{if } r \in [\![\mathsf{G}_h]\!] \wedge \exists r' \in [\![\mathsf{G}_h]\!] : \tilde{q}(\mathsf{A})(r'\downarrow_{\mathsf{A}}) \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

Since $\mathcal{I}(s)$ by hypothesis, by property (4) of Definition 19, there is $\hat{r} \in [\![\mathsf{G}]\!] = [\![\mathsf{G}_1]\!] \cup [\![\mathsf{G}_2]\!]$ such that, for all $\mathsf{A} \in \mathcal{P}$, $\tilde{q}(\mathsf{A})(\hat{r}\upharpoonright_{\mathsf{A}}) \neq \bot$. Therefore, there is a $h \in \{1,2\}$ such that $\hat{r} \in [\![\mathsf{G}_h]\!]$ and s_h is reachable in $S_h = (M([\![\mathsf{G}_h]\!]\upharpoonright_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$ otherwise s would not be a reachable configuration of S. Hence, $\mathcal{I}(s_h)$ holds by the inductive hypothesis. Hereafter we assume that $\hat{r} \in [\![\mathsf{G}_1]\!]$ (the other case is analogous).

Let $\mathsf{A} = \mathsf{sbj}(l)$ and $\mathcal{R} = \{r \in [\![\mathsf{G}]\!] \mid \partial_l(\tilde{q}(\mathsf{A}))(r\upharpoonright_{\mathsf{A}}) \neq \bot\}$. Since $s \xRightarrow{l} s'$, then $\tilde{q}' = \tilde{q}[\mathsf{A} \mapsto \partial_l(\tilde{q}(\mathsf{A}))]$, $\tilde{q}(\mathsf{A}) \xrightarrow{l} \tilde{q}'(\mathsf{A})$, and $\mathcal{R} \neq \emptyset$.

If $\mathcal{R} \cap [\![\mathsf{G}_1]\!] \neq \emptyset$, then $\tilde{q}_1(\mathsf{A}) \xrightarrow{l} \tilde{q}'_1(\mathsf{A})$ and $s_1 \xRightarrow{l} s'_1$, where $s'_1 = (\tilde{q}_1[\mathsf{A} \mapsto \partial_l(\tilde{q}_1(\mathsf{A}))], \tilde{b}')$. By the inductive hypothesis, $\mathcal{I}(s'_1)$ holds. Therefore, properties (1 – 3) of Definition 19 are satisfied by s' for every $r_1 \in [\![\mathsf{G}_1]\!]$. Also, every pomset $r_1 \in [\![\mathsf{G}_1]\!]$ that satisfies property (4) in q'_1 also satisfies (4) in q'. To prove that properties (1 – 3) are satisfied by s' for $r_2 \in [\![\mathsf{G}_2]\!]$ we distinguish three cases. If $\mathcal{R} \cap [\![\mathsf{G}_2]\!] = \emptyset$, then $q'(\mathsf{A})(r_2) = \bot$ and for every $\mathsf{B} \neq \mathsf{A}$ holds $q'(\mathsf{B})(r_2) = q(\mathsf{B})(r_2)$, thus (1 – 3) trivially hold. If $\mathcal{R} \cap [\![\mathsf{G}_2]\!] \neq \emptyset$ and s_2 is reachable in S_2, then $\tilde{q}_2(\mathsf{A}) \xrightarrow{l} \tilde{q}'_2(\mathsf{A})$ and $s_2 \xRightarrow{l} s'_2$, where $s'_2 = (\tilde{q}_2[\mathsf{A} \mapsto \partial_l(\tilde{q}_2(\mathsf{A}))], \tilde{b}')$. By the inductive hypothesis, $\mathcal{I}(s'_2)$ holds, thus properties (1 – 3) are satisfied by s' (this case corresponds to A executing events belonging to the common prefixes of the two branches). Finally, If $\mathcal{R} \cap [\![\mathsf{G}_2]\!] \neq \emptyset$ and s_2 is not reachable in S_2, then (1 – 3) vacuously hold, since there is $\mathsf{B} \neq \mathsf{A}$ such that $q(\mathsf{B})(r_2\upharpoonright_{\mathsf{B}}) = \bot$. Hence, $\mathcal{I}(s')$ holds.

If s_2 is not reachable in S_2 then for every pomset $r_2 \in [\![\mathsf{G}_2]\!]$ and every $\mathsf{B} \in \mathcal{P}$ holds $q'(\mathsf{B})(r_2) = \bot$

If $\mathcal{R} \cap [\![\mathsf{G}_1]\!] = \emptyset$, then $\mathcal{R} \cap [\![\mathsf{G}_2]\!] \neq \emptyset$ and $\tilde{q}_2(\mathsf{A}) \xrightarrow{l} \tilde{\partial}_l(\tilde{q}_2(\mathsf{A}))$. If s_2 is reachable in $S_2 = (M([\![\mathsf{G}_2]\!]\upharpoonright_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$, then there is $\hat{r}' \in [\![\mathsf{G}_2]\!]$ such that for all $\mathsf{B} \in \mathcal{P}$, $\tilde{q}(\mathsf{B})(\hat{r}'\upharpoonright_{\mathsf{B}}) \neq \bot$. The proof can continue as the previous case (i.e. s_1 is reachable and $\mathcal{R} \cap [\![\mathsf{G}_1]\!] \neq \emptyset$). Notice that this case corresponds to participant A being active and selecting the branch G_2. In fact, up to the event l, every participant $\mathsf{B} \in \mathcal{P}$ executed the same actions in G_1 and G_2 (i.e. for every $r_1 \in \mathcal{R} \cap [\![\mathsf{G}_1]\!]$ and $r_2 \in \mathcal{R} \cap [\![\mathsf{G}_2]\!]$ there are order- and label-preserving bijections among $r_1\upharpoonright_{\mathsf{B}}$, $r_2\upharpoonright_{\mathsf{B}}$, and the prefixes of the corresponding partitions). Also, A is executing an action that is not part of the common prefix, thus l is in the points of divergence on G_2 of A.

If s_2 is not reachable in $S_2 = (M([\![\mathsf{G}_2]\!]\upharpoonright_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$ then for every pomset $r_2 \in \mathcal{R} \cap [\![\mathsf{G}_2]\!]$, holds $q_2(\mathsf{A})(r_2) \neq \bot$ and there exists $\mathsf{B} \in \mathcal{P}$ such that $q_2(\mathsf{B})(r_2) = \bot$. Up to the event l, the participant A executed the same actions in G_1 and G_2 (i.e. for every $r_1 \in \mathcal{R} \cap [\![\mathsf{G}_1]\!]$ and $r_2 \in \mathcal{R} \cap [\![\mathsf{G}_2]\!]$ there are order- and label-preserving bijections among $r_1\upharpoonright_{\mathsf{A}}$, $r_2\upharpoonright_{\mathsf{A}}$, and the prefixes of the corresponding partitions). Also, A is executing an action that is not part of the common prefix, thus l is on the points of divergence on G_2 of A. This case corresponds participant A selecting the branch of G_2, and participant B selecting the branch of G_1. We show that this case is a contradiction and violates the well-branchedness condition. We distinguish two cases: input and output.

If $l = \mathsf{CA?m}$, then participant C of the corresponding output $\mathsf{CA!m}$ has

already sent the message by property (2) of Definition 19. Thus there exists a corresponding output event in $q(\mathsf{C})(r)$ for every $r \in [\![\mathsf{G}]\!]$. However, these events cannot be in the common prefixes, thus C has already discarded branch G_1 (i.e. for every $r_1 \in [\![\mathsf{G}_1]\!]$ holds $\tilde{q}(\mathsf{C})(r_1{\restriction_{\mathsf{C}}}) = \bot$), which contraddicts the hypotesis that s_1 is reachable.

If $l = \mathsf{AC!m}$, then A must be active, because its points of divergence contains a output. However, there cannot be any participant $\mathsf{B} \in \mathcal{P}$ that has already discarded all branches of G_2, which contraddicts the condition that for every pomset $r_2 \in \mathcal{R} \cap [\![\mathsf{G}_2]\!]$ exists $\mathsf{B} \in \mathcal{P}$ such that $q_2(\mathsf{B})(r_2) = \bot$, which is implied by the assumption that s_2 is not reachable. □

Theorem 1 (Progress)**.** *Given $\mathsf{G} \in \mathcal{G}$ such that $[\![\mathsf{G}]\!] \neq \bot$, if s is reachable from the initial configuration s_0 of the communicating system $(\Delta(\mathsf{G}\downarrow_{\mathsf{A}}))_{\mathsf{A}\in\mathcal{P}}$ then s is not a deadlock.*

Proof. The theorem directly follows from Lemma 11 and Lemma 10. □

Lemma 12. *If $\mathsf{G} \in \mathcal{G}$ with $[\![\mathsf{G}]\!] \neq \bot$ and $S = (M([\![\mathsf{G}]\!]{\restriction_{\mathsf{A}}}))_{\mathsf{A}\in\mathcal{P}}$ then $M([\![\mathsf{G}]\!])$ simulates S.*

Proof. The proof of the theorem is done by induction on the syntax of G. The proof is trivial for choreographies that are empty or are simple interactions.
Case $\mathsf{G} = \mathsf{G}_1; \mathsf{G}_2$. For $h \in \{1, 2\}$, let $S_h = (M([\![\mathsf{G}_h]\!]{\restriction_{\mathsf{A}}}))_{\mathsf{A}\in\mathcal{P}}$; by inductive hypothesis, there exists a simulation $\lesssim_h$ such that $M([\![\mathsf{G}_h]\!]) \lesssim_h S_h$. We exhibit a simulation relation $\lesssim$ such that $M([\![\mathsf{G}]\!]) \lesssim S$.

We know that $[\![\mathsf{G}]\!] = \{\mathsf{seq}(r_1, r_2) \mid (r_1, r_2) \in [\![\mathsf{G}_1]\!] \times [\![\mathsf{G}_1]\!]\}$. In the following, for $r \in [\![\mathsf{G}]\!]$, we use r^1 and r^2 to denote the pomsets $r_1 \in [\![\mathsf{G}_1]\!]$ and $r_2 \in [\![\mathsf{G}_1]\!]$ such that $r = \mathsf{seq}(r_1, r_2)$. Let g be a state of $M([\![\mathsf{G}]\!])$, by definition of delayed choice machine for every $r \in [\![\mathsf{G}]\!]$ holds $g(r) \subseteq \mathcal{E}_r \cup \{\bot\}$. Also, let s be a configuration of S, then for every A and $r \in [\![\mathsf{G}]\!]$ holds $s(\mathsf{A})(r{\restriction_{\mathsf{A}}}) \subseteq \mathcal{E}_{r{\restriction_{\mathsf{A}}}} \cup \{\bot\}$. For a participant A and $r \in [\![\mathsf{G}]\!]$, $g(r) = g_1(r) \uplus g_2(r)$ and $s(\mathsf{A})(r) = s_1(\mathsf{A})(r^1{\restriction_{\mathsf{A}}}) \uplus s_2(\mathsf{A})(r^2{\restriction_{\mathsf{A}}})$ where, for $h \in \{1, 2\}$ and $j \neq h$,

- for $r \in [\![\mathsf{G}_1]\!]$

$$g_1(r) = \begin{cases} \bigcup_{r' \in [\![\mathsf{G}_2]\!]} \{e \mid (e, 1) \in g(\mathsf{seq}(r, r'))\} & \exists r' \in [\![\mathsf{G}_2]\!] : g(\mathsf{seq}(r, r')) \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

- for $r \in [\![\mathsf{G}_2]\!]$

$$g_2(r) = \begin{cases} \bigcup_{r' \in [\![\mathsf{G}_1]\!]} \{e \mid (e, 2) \in g(\mathsf{seq}(r', r))\} & \exists r' \in [\![\mathsf{G}_1]\!] : g(\mathsf{seq}(r', r)) \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

- s_1 and s_2 are the configurations of S_1 and S_2 that correspond to s defined in case "Seq" of proof of Lemma 11

We say that $g \lesssim s$ iff $g_1 \lesssim_1 s_1$ and $g_2 \lesssim_2 s_2$. To prove that $\lesssim$ is a simulation, assume that $s \stackrel{l}{\Rightarrow} s'$ and $g \lesssim s$, then we must show that $g \stackrel{l}{\rightarrow} g'$ and $g' \lesssim s'$. Let $\mathsf{A} = \mathsf{sbj}(l)$, the proof follows the following steps:

1. Since $g_1 \lesssim_1 s_1$ and $g_2 \lesssim_2 s_2$ one of the two system (say $h \in \{1,2\}$) is able to execute the same transition: if exists $r_1 \in [\![\mathsf{G}_1]\!]$ such that $s_1(\mathsf{A})(r_1{\downarrow_{\mathsf{A}}}) \neq \{\mathcal{E} r_1{\downarrow_{\mathsf{A}}}, \bot\}$ then l is performed by G_1; otherwise l is performed by G_2.
2. Due to the relation among the buffers we show that the configuration of the corresponding system S_h performs the transition $s_h \stackrel{l}{\Rightarrow} s'_h$
3. The inductive hypotheses ensures that the state g_h of the CFSM machine $M([\![\mathsf{G}_h]\!])$ that is similar to s_h via $\lesssim_h$ performs the transition $g_h \stackrel{l}{\rightarrow} g'_h$.
4. finally, we show that $g \stackrel{l}{\rightarrow} g'$

The proof of (2) reuses the same arguments of the proof of Lemma 11 case "seq"'. Then (4) follows by the definition of delayed choice machine and from the fact that if the transition is enabled in the machine produced by $[\![\mathsf{G}_1]\!]$, then the same transition is executed by the machine produced by $[\![\mathsf{G}]\!]$. Similarly, if a transition is enabled in the machine produced by $[\![\mathsf{G}_2]\!]$, then Lemma 10 ensures that all events of $[\![\mathsf{G}_1]\!]$ of A have been processed, thus the same transition is enabled in $[\![\mathsf{G}]\!]$.

Case $\mathsf{G} = \mathsf{G}_1 + \mathsf{G}_2$. For $h \in \{1,2\}$, let $S_h = (M([\![\mathsf{G}_h]\!]{\downarrow_{\mathsf{A}}}))_{\mathsf{A} \in \mathcal{P}}$; by inductive hypothesis, there exists a simulation $\lesssim_h$ such that $M([\![\mathsf{G}_h]\!]) \lesssim_h S_h$. We exhibit a simulation relation $\lesssim$ such that $M([\![\mathsf{G}]\!]) \lesssim S$.

We know that $[\![\mathsf{G}]\!] = [\![\mathsf{G}_1]\!] \cup [\![\mathsf{G}_1]\!]$ Let g be a state of $M([\![\mathsf{G}]\!])$, by definition of delayed choice machine for every $r \in [\![\mathsf{G}]\!]$ holds $g(r) \subseteq \mathcal{E}_r \cup \{\bot\}$. Also, let s be a configuration of S, then for every A and $r \in [\![\mathsf{G}]\!]$ holds $s(\mathsf{A})(r{\downarrow_{\mathsf{A}}}) \subseteq \mathcal{E}_{r{\downarrow_{\mathsf{A}}}} \cup \{\bot\}$. For $h \in \{1,2\}$ let

- $g_h = (g(r))_{r \in [\![\mathsf{G}_h]\!]}$
- s_h be the configurations of S_h that correspond to s defined in case "Choice" of proof of Lemma 11

We say that $g \lesssim s$ iff

- $g_1 \lesssim_1 s_1$ or $g^2 \lesssim_2 s_2$
- for $h \in \{1,2\}$ if $g^h \not\lesssim_h s_h$ then $g_h = (\bot)_{r \in [\![\mathsf{G}_h]\!]}$

To prove that $\lesssim$ is a simulation, assume that $s \stackrel{l}{\Rightarrow} s'$ and $g \lesssim s$, then we must show that $g \stackrel{l}{\rightarrow} g'$ and $g' \lesssim s'$. Let $\mathsf{A} = \mathsf{sbj}(l)$, the proof follows the following steps:

1. if $g_1 \lesssim_1 s_1$ (the other case is analogous) and the CFSM of A obtained by $M([\![\mathsf{G}_1]\!]{\downarrow_{\mathsf{A}}})$ performs the transition $s_1(\mathsf{A}) \stackrel{l}{\rightarrow} s'_1(\mathsf{A})$ then
   (a) $s_1 \stackrel{l}{\Rightarrow} s'_1$

   (b) the inductive hypotheses ensures that the state g_1 of the automaton $M(\llbracket \mathsf{G}_1 \rrbracket)$ that is similar to s_1 via $\lesssim_1$ performs the transition $g_1 \xrightarrow{l} g_1'$ and preserves the simulation.
   (c) this guarantees that $g \xrightarrow{l} g'$ and that the simulation is preserved.
2. if $g_1 \lesssim_1 s_1$ (the other case is analogous) and the CFSM of A obtained by $M(\llbracket \mathsf{G}_1 \rrbracket\!\upharpoonright_{\mathsf{A}})$ does not perform the transition $s_1(\mathsf{A}) \xrightarrow{l} s_1'(\mathsf{A})$ then we must demonstrate that $g_1' = (\bot)_{r \in \llbracket \mathsf{G}_1 \rrbracket}$

Proof of (1.a) reuses the same arguments of the proof of Lemma 11 case "Choice. Proof of (1.c) follows by the definition of delayed choice machine and from the fact that if the transition is enabled in the machine produced by $\llbracket \mathsf{G}_1 \rrbracket$, then the same transition is executed by the machine produced by $\llbracket \mathsf{G} \rrbracket$ (which is obtained by the union of the two sets of pomsets $\llbracket \mathsf{G}_1 \rrbracket$ and $\llbracket \mathsf{G}_2 \rrbracket$), by scheduling the same event of G_1. Finally, (2) is demonstrated by showing that the transition uses an event that can not be in any pomset of $\llbracket \mathsf{G}_1 \rrbracket$. In fact, this case corresponds to the execution of an event on the points of divergence of A that has label in $\tilde{l}_{\mathsf{A},2}$.
Case $\mathsf{G} = \mathsf{G}_1 | \mathsf{G}_2$. For $h \in \{1,2\}$, let $S_h = (M(\llbracket \mathsf{G}_h \rrbracket\!\upharpoonright_{\mathsf{A}}))_{\mathsf{A} \in \mathcal{P}}$; by inductive hypothesis, there exists a simulation $\lesssim_h$ such that $M(\llbracket \mathsf{G}_h \rrbracket) \lesssim_h S_h$. We exhibit a simulation relation $\lesssim$ such that $M(\llbracket \mathsf{G} \rrbracket) \lesssim S$.

We know that $\llbracket \mathsf{G} \rrbracket = \{\mathsf{par}(r_1, r_2) \mid (r_1, r_2) \in \llbracket \mathsf{G}_1 \rrbracket \times \llbracket \mathsf{G}_2 \rrbracket\}$. In the following, for $r \in \llbracket \mathsf{G} \rrbracket$, we use r^1 and r^2 to denote the pomsets $r_1 \in \llbracket \mathsf{G}_1 \rrbracket$ and $r_2 \in \llbracket \mathsf{G}_2 \rrbracket$ such that $r = \mathsf{par}(r_1, r_2)$. Let g be a state of $M(\llbracket \mathsf{G} \rrbracket)$, by definition of delayed choice machine, for all $r \in \llbracket \mathsf{G} \rrbracket$, $g(r) \subseteq \mathcal{E}_r \cup \{\bot\}$. Also, let s be a configuration of S, then for every A and $r \in \llbracket \mathsf{G} \rrbracket$ holds $s(\mathsf{A})(r\!\upharpoonright_{\mathsf{A}}) \subseteq (\mathcal{E}_{r\!\upharpoonright_{\mathsf{A}}}) \cup \{\bot\}$. For a participant A and $r \in \llbracket \mathsf{G} \rrbracket$, $g(r) = g_1(r) \uplus g_2(r)$ and $s(\mathsf{A})(r) = s_1(\mathsf{A})(r^1\!\upharpoonright_{\mathsf{A}}) \uplus s_2(\mathsf{A})(r^2\!\upharpoonright_{\mathsf{A}})$ where, for $h \in \{1,2\}$ and $j \neq h$,

- for $r \in \llbracket \mathsf{G}_1 \rrbracket$

$$g_1(r) = \begin{cases} \bigcup_{r' \in \llbracket \mathsf{G}_2 \rrbracket} \{e \mid (e,1) \in g(\mathsf{par}(r,r'))\} & \exists r' \in \llbracket \mathsf{G}_2 \rrbracket : g(\mathsf{par}(r,r')) \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

- for $r \in \llbracket \mathsf{G}_2 \rrbracket$

$$g_2(r) = \begin{cases} \bigcup_{r' \in \llbracket \mathsf{G}_1 \rrbracket} \{e \mid (e,2) \in g(\mathsf{par}(r',r))\} & \exists r' \in \llbracket \mathsf{G}_1 \rrbracket : g(\mathsf{par}(r',r)) \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

- s_1 and s_2 are the configurations of S_1 and S_2 that correspond to s defined in case "Par" of proof of Lemma 11

We say that $g \lesssim s$ iff $g_1 \lesssim_1 s_1$ and $g_2 \lesssim_2 s_2$. To prove that $\lesssim$ is a simulation, assume that $s \overset{l}{\Rightarrow} s'$ and $g \lesssim s$, then we must show that $g \xrightarrow{l} g'$ and $g' \lesssim s'$. The proof follows the following steps:

1. Due to $wf(\mathsf{G}_1, \mathsf{G}_2)$ the message of l occurs exactly in one of G_1 and G_2. We assume that G_1 contains the message (the other case is analogous).
2. Due to the relation among the buffers and $wf(\mathsf{G}_1, \mathsf{G}_2)$ we show that the configuration of the system S_1 performs the transition $s_1 \stackrel{l}{\Rightarrow} s_1'$
3. The inductive hypotheses ensures that the state g_1 of machine $M(\llbracket \mathsf{G}_1 \rrbracket)$, which is similar to s_1 via $\lesssim_1$, performs the transition $g_1 \stackrel{l}{\rightarrow} g_1'$.
4. finally, we show that $g \stackrel{l}{\rightarrow} g'$

The proof of (2) reuses the same arguments of the proof of Lemma 11 case "Par. Finally (4) follows by the definition of delayed choice machine and from the fact that if the transition is enabled in the machine produced by $\llbracket \mathsf{G}_1 \rrbracket$, then the same transition is executed by the machine produced by $\llbracket \mathsf{G} \rrbracket$ (which is obtained by parallel composition), by scheduling the same event of G_1.

□

Theorem 2 (Adequacy)**.** *If* $\mathsf{G} \in \mathcal{G}$ *with* $\llbracket \mathsf{G} \rrbracket \neq \bot$ *and* $S = (\Delta(\mathsf{G} \downarrow_{\mathsf{A}}))_{\mathsf{A} \in \mathcal{P}}$ *then* $\mathbb{L}_S \subseteq \mathbb{L}_{\mathsf{G}}$.

Proof. The theorem directly follows from Lemma 12, Lemma 10, and Lemma 7. □

Appendix B. Proof of equivalence of the semantics

To prove the equivalence of our two semantics we will show that there is a bijective correspondence between pomsets in $\llbracket \mathsf{G} \rrbracket$ and the set $\Re_{\mathsf{G}}$ of resolutions of $\{\!|\mathsf{G}|\!\}$. To exhibit this bijective correspondence it is useful to use the following variant of the semantics given in Section 4.2.

$$\llbracket \mathbf{0} \rrbracket = \{\varepsilon\}$$

$$\llbracket \mathsf{i} \colon \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B} \rrbracket = \{(\{(\mathsf{i},\mathtt{s}),(\mathsf{i},\mathtt{r})\}, \{((\mathsf{i},\mathtt{s}),(\mathsf{i},\mathtt{s})),\ ((\mathsf{i},\mathtt{r}),(\mathsf{i},\mathtt{r})),\ ((\mathsf{i},\mathtt{s}),(\mathsf{i},\mathtt{r}))\}, \lambda)\}$$

$$\text{where } \lambda : \begin{cases} (\mathsf{i},\mathtt{s}) \mapsto \mathsf{AB!m}, \\ (\mathsf{i},\mathtt{r}) \mapsto \mathsf{AB?m} \end{cases}$$

$$\llbracket \mathsf{G} | \mathsf{G}' \rrbracket \begin{cases} \{\mathsf{par}(r,r') \mid (r,r') \in \llbracket \mathsf{G} \rrbracket \times \llbracket \mathsf{G}' \rrbracket\} & \text{if } \forall (r,r') \in \llbracket \mathsf{G} \rrbracket \times \llbracket \mathsf{G}' \rrbracket : wf(r,r') \\ \bot & \text{otherwise} \end{cases}$$

$$\llbracket \mathsf{G}; \mathsf{G}' \rrbracket = \begin{cases} \{\mathsf{seq}(r,r') \mid (r,r') \in \llbracket \mathsf{G} \rrbracket \times \llbracket \mathsf{G}' \rrbracket\} & \text{if } \forall (r,r') \in \llbracket \mathsf{G} \rrbracket \times \llbracket \mathsf{G}' \rrbracket : ws(r,r') \\ \bot & \text{otherwise} \end{cases}$$

$$\llbracket \mathsf{G} + \mathsf{G}' \rrbracket = \begin{cases} \llbracket \mathsf{G} \rrbracket \cup \llbracket \mathsf{G}' \rrbracket & \text{if } wb(\mathsf{G}, \mathsf{G}') \\ \bot & \text{otherwise} \end{cases}$$

The above semantics, which we call *representative*, assumes that the set of events $\mathcal{E}$ includes the set $\mathcal{K} \times \{\mathtt{s}, \mathtt{r}\}$; events $(\mathsf{i}, \mathtt{s})$ (resp. $(\mathsf{i}, \mathtt{r})$) are used for output (resp. input) events. The representative semantics is trivially equivalent to the one in Section 4.2 since the only difference is in the identities of the communication

events. Hereafter, we work with the representative pomsets semantics of g-choreographies, $v \in H$ will abbreviate $\exists (\!| \tilde{v}, \tilde{v}' |\!) \in H : v \in \tilde{v} \cup \tilde{v}'$ and, given a pomset r, $e \in r$ will abbreviate $e \in \mathcal{E}_r$.

An event e is an $(i, \mathtt{s})$-*event* (resp. $(i, \mathtt{r})$-*event*) if $e = (\mathsf{i}, \mathtt{s})$ (resp. $e = (\mathsf{i}, \mathtt{r})$) or e is of the form $(e', 1)$ or $(e', 2)$ and e' is an $(i, \mathtt{s})$-*event* (resp. $(i, \mathtt{r})$-*event*). For an $(\mathsf{i}, \mathtt{s})$-event (resp. $(\mathsf{i}, \mathtt{r})$-event) e, $\overline{e}$ denotes the corresponding $(\mathsf{i}, \mathtt{r})$-event (resp. $(\mathsf{i}, \mathtt{s})$-event), namely $\overline{(\mathsf{i}, \mathtt{s})} = (\mathsf{i}, \mathtt{r})$, $\overline{(\mathsf{i}, \mathtt{r})} = (\mathsf{i}, \mathtt{s})$, and $\overline{(e, h)} = (\overline{e}, h)$ for $h \in \{1, 2\}$. Note that, when $[\![\mathsf{G}]\!] \neq \bot$ and $e \in [\![\mathsf{G}]\!]$ then e is either an $(\mathsf{i}, \mathtt{s})$- or an $(\mathsf{i}, \mathtt{r})$-event.

The following lemmata will be useful in the rest of the proofs.

Lemma 13. *Let* $\mathsf{G} \in \mathcal{G}$ *(i) if* $[\![\mathsf{G}]\!] \neq \bot$ *then* $e \in [\![\mathsf{G}]\!] \iff \overline{e} \in [\![\mathsf{G}]\!]$*, and (ii) if* $\{\!|\mathsf{G}|\!\} \neq \bot$ *then* $\mathsf{AB}!\mathsf{m}_{[i]} \in \{\!|\mathsf{G}|\!\} \iff \mathsf{AB}?\mathsf{m}_{[i]} \in \{\!|\mathsf{G}|\!\}$.

Proof. By induction on the syntax of G.
Case $\mathsf{G} = \mathbf{0}$**.** We have $\{\!|\mathbf{0}|\!\} = \emptyset$ and $[\![\mathbf{0}]\!] = \{\varepsilon\}$, hence the thesis vacuously holds.
Case $\mathsf{G} = \mathsf{i} : \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$**.** We have $\{\!|\mathsf{i} : \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}|\!\} = \{(\!| \mathsf{AB}!\mathsf{m}_{[i]}, \mathsf{AB}?\mathsf{m}_{[i]} |\!)\}$ and $[\![\mathsf{i} : \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}]\!] = \{(\{(\mathsf{i}, \mathtt{s}), (\mathsf{i}, \mathtt{r})\}, \ldots\}$ by definition, and again the thesis hold.
Case $\mathsf{G} = \mathsf{G}_1 | \mathsf{G}_2$**.** We have $\{\!|\mathsf{G}_1 | \mathsf{G}_2|\!\} = \{\!|\mathsf{G}_1|\!\} \cup \{\!|\mathsf{G}_2|\!\} \cup H$ (where H is an hypergraph on the vertices of $\{\!|\mathsf{G}_1|\!\}$ and $\{\!|\mathsf{G}_2|\!\}$ as in Eq. (5) on page 29 and it is immaterial here) and since $v \in \{\!|\mathsf{G}_1 | \mathsf{G}_2|\!\} \setminus H$ iff $v \in \{\!|\mathsf{G}_1|\!\}$ or $v \in \{\!|\mathsf{G}_2|\!\}$, then part (i) of the thesis immediately follows by the inductive hypothesis.
For part (ii) of the thesis, $[\![\mathsf{G}]\!] = \{\mathsf{par}(r_1, r_2) \mid (r_1, r_2) \in [\![\mathsf{G}_1]\!] \times [\![\mathsf{G}_2]\!]\}$. Hence, for each $r \in [\![\mathsf{G}]\!]$ there are $r_1 \in [\![\mathsf{G}_1]\!]$ and $r_2 \in [\![\mathsf{G}_2]\!]$ such that $r = \mathsf{par}(r_1, r_2)$. Therefore, the events in r are of the form $(e, 1)$ with $e \in r_1$ or $(e, 2)$ with $e \in r_2$. In the former case $\overline{e} \in r_1$ by inductive hypothesis and hence $(\overline{e}, 1) \in r$; the other case is similar.

The remaining cases are similar to the last one noting that if $\mathsf{G} = \mathsf{G}_1 ; \mathsf{G}_2$ then $\mathtt{seq}(\_, \_)$ yields (disjoint) union of the events of G_1 and those of G_2 while if $\mathsf{G} = \mathsf{G}_1 + \mathsf{G}_2$ then $\{\!|\mathsf{G}|\!\}$ includes $\{\!|\mathsf{G}_1|\!\} \cup \{\!|\mathsf{G}_2|\!\}$ and $[\![\mathsf{G}]\!] = [\![\mathsf{G}_1]\!] \cup [\![\mathsf{G}_2]\!]$. □

Lemma 14. *Let* $\mathsf{G} \in \mathcal{G}$ *be such that* $[\![\mathsf{G}]\!] \neq \bot$ *and* $\{\!|\mathsf{G}|\!\} \neq \bot$*. An* $(i, \mathtt{s})$*- or* $(i, \mathtt{r})$*-event* e *is in a pomset* $r \in [\![\mathsf{G}]\!]$ *iff* $\big(\lambda_r(e)\big)_{[i]} \in \{\!|\mathsf{G}|\!\}$.

Proof sketch. Easy by structural induction on G, observing that $[\![\_]\!]$ assigns to $(\mathsf{i}, \mathtt{s})$- and $(\mathsf{i}, \mathtt{r})$-events labels that are the output or input communication actions decorated in the definition of $\{\!|\_|\!\}$ with the control point i. □

The resolution of a g-choreography corresponds to particular pomset obtained by taking the semantics after "cutting" all the branches not chosen in the resolution. Formally

Definition 20. *For* $\mathsf{G} \in \mathcal{G}$ *we define*

$$\mathsf{G}(\rho) = \begin{cases} \mathsf{G}_1(\rho_1);\mathsf{G}_2(\rho_2) & \text{if } \mathsf{G} = \mathsf{G}_1;\mathsf{G}_2 \text{ and } \forall h \in \{1,2\} : \rho_h = \rho|_{\mathit{cp}(\mathsf{G}_h)} \\ \mathsf{G}_1(\rho_1)|\mathsf{G}_2(\rho_2) & \text{if } \mathsf{G} = \mathsf{G}_1|\mathsf{G}_2 \text{ and } \forall h \in \{1,2\} : \rho_h = \rho|_{\mathit{cp}(\mathsf{G}_h)} \\ \mathsf{G}_h(\rho_h) & \text{if } \mathsf{G} = i{:}(\mathsf{G}_1 + \mathsf{G}_2) \text{ and } \exists h \in \{1,2\} : \mathit{cp}(\rho(i)) \subseteq \mathit{cp}(\mathsf{G}_h) \\ \mathsf{G} & \text{otherwise} \end{cases}$$

Note that $\mathsf{G}(\rho)$ is well defined because when $\mathsf{G} = \mathsf{i}{:}(\mathsf{G}_1 + \mathsf{G}_2)$ the events in $\rho(\mathsf{i})$ are all either in G_1 or in G_2 by the definition of resolution and that of $\{\!|\mathsf{G}|\!\}$.

Lemma 15. *If* $[\![\mathsf{G}]\!] \neq \bot$ *and* $\rho \in \Re_{\mathsf{G}}$ *then* $[\![\mathsf{G}(\rho)]\!]$ *is a singleton.*

Proof. By construction $\mathsf{G}(\rho)$ is a branching-free g-choreography and the thesis follows by definition of $[\![\_]\!]$. □

The next lemma simplifies the proof of correctness and completeness.

Lemma 16. *If* $\mathsf{G}_1, \mathsf{G}_2 \in \mathcal{G}$ *and* $\{\!|\mathsf{G}_1;\mathsf{G}_2|\!\} \neq \bot$ *then any resolution of* $\mathsf{G}_1;\mathsf{G}_2$ *can be obtained by extending a resolution of* G_1 *with a resolution of* G_2*.*

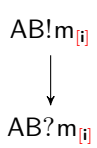
Proof. We show that for each resolution $\rho \in \Re_{\mathsf{G}_1;\mathsf{G}_2}$ there are $\rho_1 \in \Re_{\mathsf{G}_1}$ and $\rho_2 \in \Re_{\mathsf{G}_2}$ such that $\rho = \rho_1[\mathsf{i} \mapsto \rho_2(\mathsf{i})]_{\mathsf{i} \in \operatorname{dom} \rho_2}$. Note that $\operatorname{dom}\rho_1 \cap \operatorname{dom}\rho_2 = \emptyset$ since $\mathsf{cp}(\mathsf{G}_1;\mathsf{G}_2) = \mathsf{cp}(\mathsf{G}_1) \cup \mathsf{cp}(\mathsf{G}_2)$ and $\mathsf{cp}(\mathsf{G}_1) \cap \mathsf{cp}(\mathsf{G}_2) = \emptyset$. For $h \in \{1,2\}$, taking $\rho_h = \big(\mathsf{j} \mapsto \rho(\mathsf{j})\big)_{\mathsf{j} \in \operatorname{dom}\rho \cap \mathsf{cp}(\mathsf{G}_h)}$ we have that ρ_h is a resolution of G_h and $\rho = \rho_1[\mathsf{i} \mapsto \rho_2(\mathsf{i})]_{\mathsf{i} \in \operatorname{dom}\rho_2}$. □

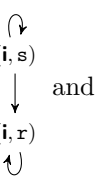
For a g-choreography $\mathsf{G} \in \mathcal{G}$ such that $[\![\mathsf{G}]\!] \neq \bot$ and $\{\!|\mathsf{G}|\!\} \neq \bot$ and a pomset $r \in [\![\mathsf{G}]\!]$, we say that $l_{[\mathsf{i}]} \in \{\!|\mathsf{G}|\!\}$ *corresponds to* $e \in \mathcal{E}_r$ (and vice versa) when $\lambda_r(e) = l$. Also, we say that a resolution $\rho \in \Re_{\mathsf{G}}$ and a pomset $r \in [\![\mathsf{G}]\!]$ are *equivalent via* η when η is a bijection between the events of r and the vertices of $\mathsf{G}(\rho)$ not in $\mathcal{K}$ such that for all $e \in r$, e corresponds to $\eta(e)$.

Theorem 4. *Given* $\mathsf{G} \in \mathcal{G}$ *for which* $[\![\mathsf{G}]\!] \neq \bot$ *and* $\{\!|\mathsf{G}|\!\} \neq \bot$*, if* $r \in [\![\mathsf{G}]\!]$ *and* $\rho \in \Re_{\mathsf{G}}$ *are equivalent via* η *then*

$$\forall e, e' \in \mathcal{E}_r : e \leq_r e' \iff \eta(e) \sqsubseteq_{\mathsf{G}(\rho)} \eta(e')$$

Proof. By induction on the syntax of G.
Case $\mathsf{G} = \mathbf{0}$**.** The thesis holds vacuously.
Case $\mathsf{G} = \mathbf{i}\colon \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$**.** The only resolution of G is an empty mapping, since $\mathsf{cp}(\mathsf{G}) \cap \mathcal{K} = \emptyset$, which yields

$\mathsf{G}\circledR\emptyset =$ 

and we set r to the only pomset in $[\![\mathsf{G}]\!]$: $\leq_r =$ 

and $\lambda_r = \begin{cases} (\mathsf{i},\mathtt{s}) \mapsto \mathsf{AB!m} \\ (\mathsf{i},\mathtt{r}) \mapsto \mathsf{AB?m} \end{cases}$

where the arrows in the diagram of $\leq_r$ represent the order relation. The thesis follows from the definition of $\sqsubseteq_{\mathsf{G}(\rho)}$.

Case $\mathsf{G} = \mathsf{i}{:}(\mathsf{G}_1|\mathsf{G}_2)$**.** We have $\mathsf{G}(\rho) = \mathsf{G}_1(\rho_1)|\mathsf{G}_2(\rho_2)$ with $\rho_1 = \rho|_{\mathsf{cp}(\mathsf{G}_1)}$ and $\rho_2 = \rho|_{\mathsf{cp}(\mathsf{G}_2)}$ (by Definition 20) and since $\{\!|\mathsf{G}|\!\} = \{\!|\mathsf{G}_1|\!\} \cup \{\!|\mathsf{G}_2|\!\} \cup H$ (where H is a hypergraph on the vertices of $\{\!|\mathsf{G}_1|\!\}$ and $\{\!|\mathsf{G}_2|\!\}$ as in Eq. (5) on page 29 and it is immaterial here) is defined by hypothesis, $\{\!|\mathsf{G}_1|\!\} \neq \bot$ and $\{\!|\mathsf{G}_2|\!\} \neq \bot$. Hence, both $[\![\mathsf{G}_1]\!] \neq \bot$ and $[\![\mathsf{G}_2]\!] \neq \bot$(otherwise $[\![\mathsf{G}]\!]$ would not be defined, contrary to our hypothesis). Also, $r = \mathsf{par}(r_1, r_2)$ for some $r_1 \in [\![\mathsf{G}_1]\!]$ and $r_2 \in [\![\mathsf{G}_2]\!]$. Let η be the bijection showing r equivalent to ρ. Since $\mathsf{cp}(\mathsf{G}_1) \cap \mathsf{cp}(\mathsf{G}_2) = \emptyset$, we have, for $h \in \{1, 2\}$, that $\eta_h : e \mapsto \eta(e, h)$ for each $e \in \mathcal{E}_{r_h}$ are bijections showing that r_h is equivalent to ρ_h. Finally, $(\!| v, v' |\!) \in \widehat{\{\!|\mathsf{G}|\!\}}^\star$ iff $(\!| v, v' |\!) \in \widehat{\{\!|\mathsf{G}_1|\!\}}^\star$ or $(\!| v, v' |\!) \in \widehat{\{\!|\mathsf{G}_2|\!\}}^\star$, and we consider the former case only since the other case is similar. By definition, $(\!| v, v' |\!) \in \widehat{\{\!|\mathsf{G}|\!\}}^\star$ is equivalent to $v \sqsubseteq_{\mathsf{G}_1(\rho_1)} v'$ which, by the inductive hypothesis, is equivalent to $\eta^{-1}(v) \leq_{r_1} \eta^{-1}(v')$ and the thesis follows since $\leq_{r_1} \times \mathbf{1} \subseteq \leq_r$.

Case $\mathsf{G} = \mathsf{G}_1; \mathsf{G}_2$**.** By Lemma 16, there are resolutions ρ_1 of G_1 and ρ_2 of G_2 with disjoint domains such that $\rho = \rho_1[\mathsf{i} \mapsto \rho_2(\mathsf{i})]_{\mathsf{i} \in \operatorname{dom} \rho_2}$. We have that both $[\![\mathsf{G}_1]\!] \neq \bot$ and $[\![\mathsf{G}_2]\!] \neq \bot$ (otherwise $[\![\mathsf{G}]\!] = \bot$ contrary to our hypothesis). Also, by inductive hypothesis and by Lemma 15,

- $[\![\mathsf{G}_1(\rho_1)]\!] = \{r_1\}$ and $[\![\mathsf{G}_2(\rho_2)]\!] = \{r_2\}$ with, for $h \in \{1, 2\}$, r_h equivalent to ρ_h, and
- $\sqsubseteq_{\mathsf{G}_h(\rho_h)}$ isomorphic to $\leq_{r_h}$ when restricting the two orders to communication events.

Note that $\mathsf{G}(\rho) = \mathsf{G}_1(\rho_1); \mathsf{G}_2(\rho_2)$ (by Definition 20) and let $e, e' \in \mathcal{E}_r$ and let $v = \eta(e)$ and $v' = \eta(e')$. We have to prove the thesis when $v \in \{\!|\mathsf{G}_1(\rho_1)|\!\}$ and $v' \in \{\!|\mathsf{G}_2(\rho_2)|\!\}$ (since the other cases follow by induction similarly to the previous case). With no loss of generality, we can assume that v is maximal in $\{\!|\mathsf{G}_1(\rho_1)|\!\}$ and v' is minimal in $\{\!|\mathsf{G}_2(\rho_2)|\!\}$ (otherwise, using transitivity and induction the proof reduces to such case). More formally, we can assume that there are $(\!| \tilde{v}_1, \tilde{v}_2 |\!) \in \mathsf{lst}\,\{\!|\mathsf{G}_1(\rho_1)|\!\}$ and $(\!| \tilde{v}'_1, \tilde{v}'_2 |\!) \in \mathsf{fst}\,\mathsf{G}_2(\rho_2)$ such that $v \in \tilde{v}_1 \cup \tilde{v}_2$ and $v' \in \tilde{v}'_1 \cup \tilde{v}'_2$. Since $v \sqsubseteq_{\mathsf{G}(\rho)} v'$ by hypothesis, it must be $\mathsf{sbj}(v) = \mathsf{sbj}(v')$ by the definition of $\mathsf{seq}(\mathsf{G}_1(\rho_1), \mathsf{G}_2(\rho_2))$. Hence, we have that $(e, e') \in \mathsf{seq}(r_1, r_2)$ by definition. The proof ends noting that the other direction is similar.

Case $\mathsf{G} = \mathsf{i}{:}(\mathsf{G}_1 + \mathsf{G}_2)$**.** By Definition 20, $\mathsf{G}(\rho)$ yields either a resolution of G_1 or one of G_2. In either case, the thesis immediately follows by induction. □

For a pomset r, define $\mathtt{max}\ r = \{e \in \mathcal{E}_r \mid \nexists e' \in \mathcal{E}_r : e' \neq e \wedge e \leq_r e'\}$. The following lemma is a simple corollary of Theorem 4.

Lemma 17. *Given* $\mathsf{G} \in \mathcal{G}$ *for which* $[\![\mathsf{G}]\!] \neq \bot$ *and* $\{\!|\mathsf{G}|\!\} \neq \bot$*, if* $r \in [\![\mathsf{G}]\!]$ *and* $\rho \in \Re_{\mathsf{G}}$ *are equivalent via* η *then* η *establishes bijective correspondences between* $\mathtt{min}\ r$ *and* $\bigcup \mathsf{cs}(\mathsf{fst}\,\mathsf{G}(\rho))$ *and between* $\mathtt{max}\ r$ *and* $\bigcup \mathsf{ef}(\mathsf{lst}\,\mathsf{G}(\rho))$*.*

Proof sketch. By inspection of the proof of Theorem 4. □

We establish some useful properties of A-uniformity and reflectivity before proving the equivalence of our semantics.

Lemma 18. *Let $\mathsf{G} \in \mathcal{G}$ be such that $\{\!|\mathsf{G}|\!\} \neq \bot$. If $\tilde{v} \subseteq \mathsf{G}$ is A-uniform for a participant A of G and $\rho \in \mathfrak{R}_{\mathsf{G}}$ then $\tilde{v} \cap \mathsf{G}(\rho)$ contains at most one vertex.*

Proof. By construction, each vertex in $\mathsf{G}(\rho)$ has at most one outgoing hyperedge. Therefore, two vertices $v, v' \in \mathsf{G}(\rho)$ are not comparable with respect to $\sqsubseteq_{\mathsf{G}(\rho)}$ only if they are independent, hence they cannot both belong to $\tilde{v}$. □

It is useful to adopt the following terminology and notation. We say that $[\![\mathsf{G}]\!]$ *corresponds* to $\mathfrak{R}_{\mathsf{G}}$ (and viceversa) when there is a bijection $\beta : [\![\mathsf{G}]\!] \to \mathfrak{R}_{\mathsf{G}}$ such that r and $\beta(r)$ are equivalent for each $r \in [\![\mathsf{G}]\!]$; the identity of such bijection will often be immaterial, therefore, if $r \in [\![\mathsf{G}]\!]$ we write $\breve{r} \in \mathfrak{R}_{\mathsf{G}}$ to denote the equivalent resolution, likewise if $\rho \in \mathfrak{R}_{\mathsf{G}}$, we write $\breve{\rho} \in [\![\mathsf{G}]\!]$ to denote the pomset equivalent to ρ.

Lemma 19. *For $h \in \{1, 2\}$, given $\mathsf{G}_h \in \mathcal{G}$ such that $\{\!|\mathsf{G}_h|\!\} \neq \bot$ and $[\![\mathsf{G}_h]\!] \neq \bot$, $\mathsf{A} \in \mathcal{P}$, and V_h partition of a subset of vertices of $\{\!|\mathsf{G}_h|\!\}$ if V_2 A-reflects V_1 and $\mathfrak{R}_{\mathsf{G}_h}$ corresponds to $[\![\mathsf{G}_h]\!]$ then there is an A-prefix map of G_1 and G_2.*

Proof. Let $f : V_1 \to V_2$ be the bijection exhibiting the reflectivity relation between V_1 and V_2 and, moreover, for $h \in \{1, 2\}$,

- let $\rho \vdash_h \tilde{v}$ hold when $\mathsf{G}(\rho)$ contains at least vertices in $\tilde{v} \subseteq \{\!|\mathsf{G}_h|\!\}$
- let $\equiv_h$ be the equivalence relation of $\mathfrak{R}_{\mathsf{G}_h}$ defined by $\rho \equiv_h \rho' \iff (\forall \tilde{v} \in V_h : \rho \vdash_h \tilde{v} \iff \rho' \vdash_h \tilde{v})$.

Note that f induces a bijection between $\mathfrak{R}_{\mathsf{G}_1/\equiv_1}$ and $\mathfrak{R}_{\mathsf{G}_2/\equiv_2}$ by mapping $\big[\rho\big]_{\equiv_h}$ to $\big[f(\rho)\big]_{\equiv_h}$. Also, we can define an equivalence relation $\sim_h$ on $[\![\mathsf{G}_h]\!]$ corresponding to $\equiv_h$, namely $r \sim_h r' \iff \breve{r} \equiv_h \breve{r'}$. We can now exhibit an A-prefix map (ϕ, ψ) as follows:

$$\phi : \big(\big[\breve{\rho}\big]_{\sim_1}\big)\!\downarrow_{\mathsf{A}} \mapsto \big(\big[\breve{f}(\rho)\big]_{\sim_2}\big)\!\downarrow_{\mathsf{A}} \qquad \text{and} \qquad \psi : \big(\big[\breve{\rho}\big]_{\sim_1}\big)\!\downarrow_{\mathsf{A}} \mapsto [(V, \leq, \lambda)] \quad \text{where}$$

- $V = \{\tilde{v} \in V_1 \mid \forall r \in \big(\big[\breve{\rho}\big]_{\sim_1}\big)\!\downarrow_{\mathsf{A}} : \breve{r} \vdash_1 \tilde{v}\}$,
- $\lambda(\tilde{v}) = \mathsf{act}(v)$ if $v \in \tilde{v}$, and
- $\tilde{v} \leq \tilde{v}' \iff \exists v \in \tilde{v}, v' \in \tilde{v}' : v \sqsubseteq_{\mathsf{G}_1} v'$.

Note that ϕ is a bijection (because f is bijective), $\operatorname{dom}\phi = \big([\![\mathsf{G}_1]\!]_{/\sim_1}\big)\!\downarrow_{\mathsf{A}} = \operatorname{dom}\psi$, $\operatorname{cod}\phi = \big([\![\mathsf{G}_2]\!]_{/\sim_2}\big)\!\downarrow_{\mathsf{A}}$, and λ is well-defined because $\mathsf{act}(\tilde{v})$ is a singleton. Also, the relation $\leq$ is a partial order. To verify that for all $\mathsf{R} = \big(\big[r\big]_{\sim_1}\big)\!\downarrow_{\mathsf{A}} \in \operatorname{dom}\phi$ and $(r_1, r_2) \in \mathsf{R} \times \phi(\mathsf{R})$, the pomset $\psi(\mathsf{R})$ is a prefix pomset of r_h, for $h \in \{1, 2\}$, we note that for each $\tilde{v} \in V$ and belonging to $\breve{r_1}$ (by Lemma 18); hence, we can univocally map V in $\mathcal{E}_{r_1}$ (associating each $\tilde{v} \in V$ to the event e that corresponds to the vertex $v \in \breve{r_1} \cap \tilde{v}$). Finally, if $\tilde{v} \leq \tilde{v}'$ then there are $v \in \tilde{v}$ and $v' \in \tilde{v}'$ such that $v \sqsubseteq_{\mathsf{G}_1(\breve{r_1})}$, hence by Theorem 4 the event corresponding to v precedes the one corresponding to v' in $\leq_{r_1}$. □

Lemma 20. *Given* $\mathsf{A} \in \mathcal{P}$ *and, for* $h \in \{1,2\}$, $\mathsf{G}_h \in \mathcal{G}$ *such that* $[\![\mathsf{G}_h]\!] \neq \bot$ *and* $\{\!|\mathsf{G}_h|\!\} \neq \bot$, *if* $\Re_{\mathsf{G}_h}$ *corresponds to* $[\![\mathsf{G}_h]\!]$ *and there is an* A*-prefix map* (ϕ, ψ) *of* G_1 *and* G_2 *then there are* V_1 *and* V_2 *partition of a subset of vertices of* $\{\!|\mathsf{G}_1|\!\}$ *and* $\{\!|\mathsf{G}_2|\!\}$ *such that* V_2 A*-reflects* V_1.

Proof. Fix $\mathsf{R} \in \operatorname{dom}\phi$. We will write $e \in \mathsf{R}$ when $\exists r \in \mathsf{R} : e \in \mathcal{E}_r$. By definition of A-prefix map (cf. Definition 11), for $r \in \mathsf{R}$ and $r' \in \phi(\mathsf{R})$, is a prefix of both r and r', that is (cf. Definition 9) there is an order- and label-preserving bijection π^{R}_r from $\mathcal{E}_{\psi(\mathsf{R})}$ to a subset of $\mathcal{E}_r$ and similarly for $\mathcal{E}_{r'}$. Say that $e \in \mathsf{R}$ corresponds to $e' \in \phi(\mathsf{R})$ when $\exists \hat{e} \in \mathcal{E}_{\psi(\mathsf{R})} : \pi^{\mathsf{R}}_r(\hat{e}) = e \wedge \pi^{\phi(\mathsf{R})}_r(\hat{e}) = e'$; when e corresponds to e' we will write $\breve{e}$ to denote e' and $\breve{e'}$ to denote e.

The relations $\sim_{\mathsf{R}}$ on $\bigcup_{r\in\mathsf{R}} \pi^{\mathsf{R}}_r(\mathcal{E}_r)$, $\simeq_1$ on $\mathcal{E}_1 = \bigcup_{\mathsf{R}\in\operatorname{dom}\phi} \bigcup_{r\in\mathsf{R}} \pi^{\mathsf{R}}_r(\mathcal{E}_r)$, and $\simeq_2$ on $\mathcal{E}_2 = \bigcup_{\mathsf{R}\in\operatorname{cod}\phi} \bigcup_{r\in\mathsf{R}} \pi^{\mathsf{R}}_r(\mathcal{E}_r)$, are defined as:

$$
\begin{aligned}
e \sim_{\mathsf{R}} e' &\iff \pi^{\mathsf{R}}_r(e) = \pi^{\mathsf{R}}_r(e') \\
e \simeq_1 e' &\iff (\exists \mathsf{R} \in \operatorname{dom}\phi : e \sim_{\mathsf{R}} e') \vee \\
&\qquad (\exists \mathsf{R}, \mathsf{R}' \in \operatorname{dom}\phi : e \in \mathsf{R}, e' \in \mathsf{R}' \wedge [e]_{\sim_{\mathsf{R}}} \cap [e']_{\sim_{\mathsf{R}'}} \neq \emptyset) \\
e \simeq_2 e' &\iff (\exists \mathsf{R} \in \operatorname{cod}\phi : e \sim_{\mathsf{R}} e') \vee \\
&\qquad (\exists \mathsf{R}, \mathsf{R}' \in \operatorname{cod}\phi : e \in \mathsf{R}, e' \in \mathsf{R}' \wedge [e]_{\sim_{\mathsf{R}}} \cap [e']_{\sim_{\mathsf{R}'}} \neq \emptyset)
\end{aligned}
$$

Note that if $e \simeq_1 e'$ then there is $r \in [\![\mathsf{G}_1]\!]$ such that $e, e' \in \mathcal{E}_r \iff e = e'$; in fact, e and e' could be on the same pomset only if $e \sim_{\mathsf{R}} e'$ for some $\mathsf{R} \in \operatorname{dom}\phi$ and $\pi^{\mathsf{R}}_r(e) = \pi^{\mathsf{R}}_r(e') \iff e = e'$ by the injectivity of π^{R}_r. (The same holds for $\simeq_2$.) It is also easy to verify that $\sim_{\mathsf{R}}$, $\simeq_1$, and $\simeq_2$ are equivalence relations and that, for $h \in \{1,2\}$,

$$
V_h = \bigcup_{e\in\mathcal{E}_h} \{v \in \{\!|\mathsf{G}_h|\!\} \mid \exists e' \in [e]_{\simeq_h} : e' \text{ corresponds to } v\}
$$

is a partition (induced by $\simeq_h$) of the vertices of $\{\!|\mathsf{G}_h|\!\}$ corresponding to the events in $\mathcal{E}_h$. Note that V_1 and V_2 have the same cardinality; in fact, $\mathcal{E}_{1/\simeq_1}$ bijectively corresponds to $\mathcal{E}_{2/\simeq_2}$ because ϕ is a bijection and, for all $\mathsf{R} \in \operatorname{dom}\phi$, $\psi(\mathsf{R})$ is a prefix of each pomset in R and in $\phi(\mathsf{R})$. Therefore,

$$
\forall e \in \pi^{\mathsf{R}}_r(\mathcal{E}_{\psi(\mathsf{R})}) : \breve{e} \in \pi^{\phi(\mathsf{R})}_{r'}(\mathcal{E}_{\psi(\mathsf{R})}) \quad \text{and} \quad \forall e' \in \pi^{\phi(\mathsf{R})}_{r'}(\mathcal{E}_{\psi(\mathsf{R})}) : \breve{e'} \in \pi^{\mathsf{R}}_r(\mathcal{E}_{\psi(\mathsf{R})}) \quad \text{(B.1)}
$$

Hence, $\forall \mathsf{R}, \mathsf{R}' \in \operatorname{dom}\phi : \mathsf{R} \cap \mathsf{R}' \neq \emptyset \iff \phi(\mathsf{R}) \cap \phi(\mathsf{R}') \neq \emptyset$.

We now show that V_2 A-reflects V_1. Let $f : V_1 \to V_2$ such that $f : \tilde{v}_1 \mapsto \tilde{v}_2$ iff, for $h \in \{1,2\}$ there are $v_h \in \tilde{v}_h$ and $e_h \in \mathcal{E}_h$ such that e_h corresponds to v_h and $\forall \mathsf{R} \in \operatorname{dom}\phi : e_1 \in \mathsf{R} \iff e_2 \in \phi(\mathsf{R})$. Then f is a bijection because of (B.1) and of the bijective correspondence between $\mathcal{E}_{1/\simeq_1}$ and $\mathcal{E}_{2/\simeq_2}$. Also, for each $\tilde{v} \in V_1$, both $\tilde{v}$ and $f(\tilde{v})$ are A-uniform, in fact: $\tilde{v} \cap \mathcal{K} = f(\tilde{v}) \cap \mathcal{K} = \emptyset$ (since each vertex in $\tilde{v} \cup f(\tilde{v})$ corresponds to an event in $[\![\mathsf{G}]\!]\lfloor_{\mathsf{A}}$) which implies that

- $\mathsf{sbj}(\tilde{v}) = \mathsf{sbj}(f(\tilde{v})) = \{\mathsf{A}\}$ as well as $\mathsf{act}(\tilde{v}) = \mathsf{act}(f(\tilde{v}))$ are singletons, and

- each $v \neq v' \in \tilde{v}$ (resp. $v \neq v' \in f(\tilde{v})$) are not independent (because their corresponding events, say e and e', belong to different pomsets, hence v and v' must be on different resolutions due to the bijective correspondence between $[\![\mathsf{G}_1]\!]$ and $\Re_{\mathsf{G}_1}$ (resp. $[\![\mathsf{G}_2]\!]$ and $\Re_{\mathsf{G}_2}$) and therefore $v \not\sqsubseteq_{\mathsf{G}_1} v'$ and $v' \not\sqsubseteq_{\mathsf{G}_1} v$ (resp. $v \not\sqsubseteq_{\mathsf{G}_2} v'$ and $v' \not\sqsubseteq_{\mathsf{G}_2} v$).

Let $\tilde{v}_2 \in V_1$ and $v_2 \in \tilde{v}_2$ be such that $v_1 \sqsubseteq_{\mathsf{G}_1} v_2$ for a vertex $v_1 \in \{\!|\mathsf{G}_1|\!\}$ with $\mathsf{sbj}(v_1) = \mathsf{A}$. Then, observing that $\operatorname{dom}\phi$ partitions $[\![\mathsf{G}_1]\!]\downarrow_{\mathsf{A}}$, there are $\mathsf{R} \in \operatorname{dom}\phi$ and $e_1 \in \mathsf{R}$ such that e_1 corresponds to v_1; this implies that $\exists \tilde{v}_1 \in V_1 : v_1 \in \tilde{v}_1$. Additionally, $\forall v \in \tilde{v}_2, v' \in v_1 : v \not\sqsubseteq_{\mathsf{G}_1} v'$ (otherwise we would have the corresponding events to be ordered according to $\leq_{\mathsf{G}_1}$) and that $\forall v'_2 \in f(\tilde{v}_2)\ \exists v'_1 \in f(\tilde{v}_1) : v'_1 \sqsubseteq_{\mathsf{G}1} v'_2$ by definition of f.

The proof of the last condition of reflectivity is omitted because it is similar to the proof of the second condition. □

The following theorem establishes the equivalence of our semantics.

Theorem 3 (Correctness and completeness)**.** *For each $\mathsf{G} \in \mathcal{G}$, $\{\!|\mathsf{G}|\!\} \neq \bot$ iff $[\![\mathsf{G}]\!] \neq \bot$. Moreover, for each $\rho \in \Re_{\mathsf{G}}$ there is a pomset $r \in [\![\mathsf{G}]\!]$ equivalent to ρ and vice versa.*

Proof. By induction on the syntax of G.
Case $\mathsf{G} = \mathbf{0}$. The thesis holds vacuously.
Case $\mathsf{G} = \mathsf{i}\colon \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B}$. By inspection of the proof for the same case of Theorem 4.
Case $\mathsf{G} = \mathsf{i}\!:\!(\mathsf{G}_1|\mathsf{G}_2)$. We have $\mathsf{G}(\rho) = \mathsf{G}_1(\rho_1)|\mathsf{G}_2(\rho_2)$ with $\rho_1 = \rho|_{\mathsf{cp}(\mathsf{G}_1)}$ and $\rho_2 = \rho|_{\mathsf{cp}(\mathsf{G}_2)}$ (by Definition 20) and since $\{\!|\mathsf{G}|\!\} = \{\!|\mathsf{G}_1|\!\} \cup \{\!|\mathsf{G}_2|\!\} \cup H$ is defined by hypothesis, $(\{\!|\mathsf{G}_1|\!\} \sqcap \{\!|\mathsf{G}_2|\!\}) \cap \mathcal{L}^? = \emptyset$ and both $\{\!|\mathsf{G}_1|\!\} \neq \bot$ and $\{\!|\mathsf{G}_2|\!\} \neq \bot$ (here H is an hypergraph on the vertices of $\{\!|\mathsf{G}_1|\!\}$ and $\{\!|\mathsf{G}_2|\!\}$ as in Eq. (5) on page 29 and it is immaterial here). Hence, both $[\![\mathsf{G}_1]\!] \neq \bot$ and $[\![\mathsf{G}_2]\!] \neq \bot$ and, for each $h \in \{1,2\}$ there is a pomset $r_h \in [\![\mathsf{G}_h]\!]$ equivalent to ρ_h via some η_h (by the inductive hypothesis). Since the control points of G_1 and G_2 are disjoint, the function

$$\eta : e \mapsto \begin{cases} \eta_1(e) & \text{if } e \in r_1 \\ \eta_2(e) & \text{if } e \in r_2 \end{cases}$$

shows that ρ is equivalent to $\mathsf{par}(r_1, r_2) \in [\![\mathsf{G}]\!]$. Also, $wf(r_1, r_2)$ otherwise condition $(\{\!|\mathsf{G}_1|\!\} \sqcap \{\!|\mathsf{G}_2|\!\}) \cap \mathcal{L}^? = \emptyset$ would be violated.
Case $\mathsf{G} = \mathsf{G}_1; \mathsf{G}_2$. We first prove the $\Longrightarrow$ direction. If $\{\!|\mathsf{G}|\!\} \neq \bot$ then, $\{\!|\mathsf{G}_h|\!\} \neq \bot$, for $h \in \{1,2\}$, and therefore by the inductive hypothesis, $[\![\mathsf{G}_h]\!] \neq \bot$ and there is r_h equivalent to $\rho|_{\mathsf{cp}(\mathsf{G}_h)}$ via some η_h. Let $e \in [\![\mathsf{G}_1]\!]$ be an $(\mathsf{i}, \mathtt{s})$-event and $e' \in [\![\mathsf{G}_2]\!]$ be an $(\mathsf{j}, \mathtt{r})$-event. By Lemma 14, there are $v = \big(\lambda_{r_1}(e)\big)_{[\mathsf{i}]} \in \{\!|\mathsf{G}_1|\!\}$ and $v' = \big(\lambda_{r_2}(e')\big)_{[\mathsf{i}]} \in \{\!|\mathsf{G}_2|\!\}$ that correspond to e and e' respectively. Hence, we can find two resolutions ρ_1 and ρ_2 such that $v \in \mathsf{G}_1(\rho_1)$ and $v' \in \mathsf{G}_2(\rho_2)$. Then there are $v_1 \in \mathsf{lst}\{\!|\mathsf{G}_1(\rho_1)|\!\}$, $v_2 \in \mathsf{fst}\{\!|\mathsf{G}_2(\rho_2)|\!\}$, and for $h \in \{1,2\}$, $\breve{v}_h \in \{\!|\mathsf{G}_1(\rho_h)|\!\}$ with $\mathsf{sbj}(\breve{v}_1) = \mathsf{sbj}(\breve{v}_2)$ such that

$$v \sqsubseteq_{\mathsf{G}_1(\rho_1)} \breve{v}_1 \sqsubseteq_{\mathsf{G}_1(\rho_1)} v_1 \qquad \text{and} \qquad v_2 \sqsubseteq_{\mathsf{G}_2(\rho_2)} \breve{v}_2 \sqsubseteq_{\mathsf{G}_2(\rho_2)} v'$$

otherwise the condition for $\{\!|\mathsf{G}|\!\}$ to be defined could not hold, contradicting our hypothesis. Therefore, for $h \in \{1,2\}$, the events $e_h \in r_h$ corresponding to e_h (which exist by inductive hypothesis) we have

$$e \ \leq_{r_1} \ e_1 \qquad \text{and} \qquad (e_1,1) \ \leq_{\mathsf{seq}(r_1,r_2)} \ (e_2,2) \qquad \text{and} \qquad e_2 \ \leq_{r_1} \ e'$$

in fact, the relation in the middle holds by construction (since e_1 and e_2 have the same subject) and the other relations hold by inductive hypothesis. This proves that

$$\leq_{\mathsf{seq}(r_1,r_2)} \quad \supseteq \quad \{(e,1) \in \mathcal{E}_{r_1} \mid \lambda_{r_1}(e) \in \mathcal{L}^!\} \times \{(e,2) \in \mathcal{E}_{r_2} \mid \lambda_{r_2}(e) \in \mathcal{L}^?\}$$

namely, that $[\![\mathsf{G}]\!] \neq \bot$.

We now show that each $\rho \in \Re_{\mathsf{G}}$ has a correspondent equivalent $r \in [\![\mathsf{G}]\!]$. By Definition 20, $\mathsf{G}(\rho) = \mathsf{G}_1(\rho_1); \mathsf{G}_2(\rho_2)$ where for $h \in \{1,2\}$ $\rho_h = \rho|_{\mathsf{cp}(\mathsf{G}_h)}$; hence, by the inductive hypothesis, there are $r_1 \in [\![\mathsf{G}_1]\!]$ and $r_2 \in [\![\mathsf{G}_2]\!]$ respectively equivalent to ρ_1 and ρ_2. Finally, the function η defined as in the previous case shows that $\mathsf{seq}(r_1, r_2)$ is equivalent to ρ.

We now show the $\Longleftarrow$ direction. If $[\![\mathsf{G}]\!] \neq \bot$ then, for $[\![\mathsf{G}_h]\!] \neq \bot$, for $h \in \{1,2\}$, and therefore by the inductive hypothesis, $\{\!|\mathsf{G}_h|\!\} \neq \bot$ and for each $r_h \in [\![\mathsf{G}_h]\!]$ there is $\rho|_{\mathsf{cp}(\mathsf{G}_h)} \in \{\!|\mathsf{G}_h|\!\}$ equivalent to r_h via some η_h. To show that $\mathrm{ws}(\mathsf{G}_1, \mathsf{G}_2)$ consider $(v, v') \in \mathsf{cs}(\mathsf{lst}\,\mathsf{G}_1) \times \mathsf{ef}(\mathsf{fst}\,\mathsf{G}_2)$, let $e \in [\![\mathsf{G}_1]\!]$ and $e' \in [\![\mathsf{G}_2]\!]$ be the events that correspond to v and v' respectively (which exist by the inductive hypothesis) and assume $e \in r$ and $e' \in r'$. Also, the inductive hypothesis ensures the existence of $\rho \in \{\!|\mathsf{G}_1|\!\}$ and $\rho' \in \{\!|\mathsf{G}_2|\!\}$ equivalent to r and r' respectively. By construction, $v \in \mathcal{L}^! \times \mathcal{K}$ and $v' \in \mathcal{L}^? \times \mathcal{K}$ (the proof is easy by induction of the structure of the graph). Since $ws(r, r')$ holds, we have that

$$e \ \leq_{r} \ e_1 \qquad \text{and} \qquad (e_1,1) \ \leq_{\mathsf{seq}(r,r')} \ (e_2,2) \qquad \text{and} \qquad e_2 \ \leq_{r'} \ e'$$

for some $e_1 \in r$ and $e_2 \in r'$ with $\mathsf{sbj}(\lambda_r(e)) = \mathsf{sbj}(\lambda_{r'}(e))$. Hence, $v \sqsubseteq_{\mathsf{G}_1(\rho)} v_1$ and $v_2 \sqsubseteq_{\mathsf{G}_2(\rho')} v'$ by the inductive hypothesis, and the proof ends by noting that $(v_1, v_2) \in \mathsf{seq}(\{\!|\mathsf{G}|\!\}, \{\!|\mathsf{G}'|\!\})$ by the definition of $\mathsf{seq}(\_,\_)$.

Case $\mathsf{G} = \mathsf{i}{:}(\mathsf{G}_1 + \mathsf{G}_2)$**.** We first show the $\Longrightarrow$ direction. Since, $\{\!|\mathsf{G}|\!\} \neq \bot$ for $h \in \{1,2\}$ we have that $\{\!|\mathsf{G}_h|\!\} \neq \bot$ and, applying the inductive hypothesis, $[\![\mathsf{G}_h]\!] \neq \bot$ and there is bijective correspondence between $[\![\mathsf{G}_h]\!]$ an $\Re_{\mathsf{G}_h}$. To show that $[\![\mathsf{G}]\!] \neq \bot$, we have to prove that $wb(\mathsf{G}_1, \mathsf{G}_2)$ holds. By hypothesis, $\mathrm{wb}(\mathsf{G}_1, \mathsf{G}_2)$ holds, hence for each participant A of G there are there are V_1 and V_2 partitions of subsets of vertices of G_1 and G_2, respectively, such that V_2 A-reflects V_1 and the A-branching pair $(\tilde{v}_1, \tilde{v}_2) = \mathtt{div}_{\mathsf{A}}^{V_1,V_2}(\mathsf{G}_1, \mathsf{G}_2)$ witnesses that A is either active or passive according to Definitions 15 and 16. By definition,

$$\tilde{v}_1 = \bigcup \mathsf{cs}(\mathsf{fst}\,(\{\!|\mathsf{G}_1|\!\}_{@\mathsf{A}})) \setminus \bigcup V_1 \qquad \text{and} \qquad \tilde{v}_2 = \bigcup \mathsf{cs}(\mathsf{fst}\,(\{\!|\mathsf{G}_2|\!\}_{@\mathsf{A}})) \setminus \bigcup V_2$$

We show that the A-prefix map (ϕ, ψ) built in the proof of Lemma 19 yields a bijective correspondence between $\mathtt{div}_{\mathsf{A}}^{V_1,V_2}(\mathsf{G}_1, \mathsf{G}_2)$ and $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\tilde{l}_1, \tilde{l}_2)$.

We have (cf. Definition 12)

$$\tilde{l}_1 = \bigcup_{\substack{\mathsf{R} \in \operatorname{dom}\phi,\\ r \in \mathsf{R}}} \lambda_r\big(\mathtt{min}\ (r - \psi(\mathsf{R}))\big) \quad \text{and} \quad \tilde{l}_2 = \bigcup_{\substack{\mathsf{R} \in \operatorname{cod}\phi,\\ r \in \mathsf{R}}} \lambda_r\big(\mathtt{min}\ (r - \psi(\phi^{-1}(\mathsf{R})))\big)$$

Given $\mathsf{R} = \big[r\big]_{\sim_1} \in \operatorname{dom}\phi$, let $V_\mathsf{R} = \{v \in \bigcup V_1 \mid \exists e \in \mathcal{E}_{\psi(\mathsf{R})} : v \text{ corresponds to } e\}$. Then $\bigcup_{\mathsf{R}\in\operatorname{dom}\phi} V_\mathsf{R} = \bigcup V_1$; in fact, if $v \in \bigcup V_1$ then $v \in \tilde{v}$ for some $\tilde{v} \in V_1$, therefore, for each $\rho \in \Re_{\mathsf{G}_1}$ such that $v \in \big\{\!\!\big\{\mathsf{G}_1(\rho)\big\}\!\!\big\}$ and we have $v \in \big[\breve{\rho}\big]_{\sim_1}$, hence $v \in \bigcup_{\mathsf{R}\in\operatorname{dom}\phi} V_\mathsf{R}$ (and trivially $\bigcup_{\mathsf{R}\in\operatorname{dom}\phi} V_\mathsf{R} \subseteq \bigcup V_1$ because each $V_\mathsf{R} \subseteq V_1$ for each $\mathsf{R} \in \operatorname{dom}\phi$). Finally, we exhibit a bijective label-preserving correspondence between $\tilde{v}_1$ and $\tilde{l}_1$. For $v \in \tilde{v}_1$ there must be a resolution $\rho \in \Re_{\mathsf{G}_1}$ such that $v \in \big\{\!\!\big\{\mathsf{G}_1(\rho)\big\}\!\!\big\} \setminus \bigcup V_1$ and no other vertex with a communication of A preceding v. Therefore there is $e \in \breve{\rho}$ that corresponds to v; moreover, by inductive hypothesis, $e \notin \psi(\big[\breve{\rho}\big]_{\sim_1})$, hence by Theorem 4, $\lambda_e \in \tilde{l}_1$. For the converse, let $r \in [\![\mathsf{G}_1]\!]$ and $e \in \mathcal{E}_r$ such that $\lambda_r(e) \in \tilde{l}_1$ and let v be the vertex of $\big\{\!\!\big\{\mathsf{G}_1(\breve{r})\big\}\!\!\big\}$ that corresponds to e then, $v \notin \bigcup V_1$ since $e \notin \psi(\big[\breve{\rho}\big]_{\sim_1})$ and again by Theorem 4, it must be $v \in v_1$. By the bijectivity of ϕ and using a similar argument, we can find a label-preserving bijection between $\tilde{v}_2$ and $\tilde{l}_2$.

If A is active in the sense of Definition 15 then

$$\tilde{v}_1 \cup \tilde{v}_2 \subseteq (\mathcal{L}^! \times \mathcal{K}) \qquad \tilde{v}_1 \sqcap \tilde{v}_2 = \emptyset \qquad \tilde{v}_1 \neq \emptyset \qquad \tilde{v}_2 \neq \emptyset$$

hence, by the bijective correspondences shown above, we have

$$\tilde{l}_1 \cup \tilde{l}_2 \subseteq \mathcal{L}^! \qquad \tilde{l}_1 \cap \tilde{l}_2 = \emptyset \qquad \tilde{l}_1 \neq \emptyset \qquad \tilde{l}_2 \neq \emptyset$$

proving that A is active in the sense of Definition 13. If A is passive in the sense of Definition 16 then

$$\begin{array}{l|ll} \tilde{v}_1 \sqcap \{v \in \mathsf{G}'\} & \nexists v' \in \tilde{v}_2 : v \sqsubseteq_{\mathsf{G}'} v'\} = \emptyset & \tilde{v}_1 \cup \tilde{v}_2 \subseteq (\mathcal{L}^? \times \mathcal{K}) \\ \tilde{v}_2 \sqcap \{v \in \mathsf{G}\} & \nexists v' \in \tilde{v}_1 : v \sqsubseteq_{\mathsf{G}} v'\} = \emptyset & \tilde{v}_1 = \emptyset \iff \tilde{v}_2 = \emptyset \end{array}$$

and to show that A is passive in the sense of Definition 14) we have to show that

- $\tilde{l}_1 \cup \tilde{l}_2 \subseteq \mathcal{L}^?$ and $\tilde{l}_1 = \emptyset \iff \tilde{l}_2 = \emptyset$
- $\forall \mathsf{R} \in \operatorname{dom}\phi, r \in \mathsf{R} : \tilde{l}_2 \cap \lambda_r(\mathcal{E}_{r-\psi(\mathsf{R})}) = \emptyset$
- $\forall \mathsf{R} \in \operatorname{cod}\phi, r \in \mathsf{R} : \tilde{l}_1 \cap \lambda_r(\mathcal{E}_{r-\psi(\phi^{-1}(\mathsf{R}))}) = \emptyset$

The first condition above immediately follow from the bijective correspondence between $\mathtt{div}_\mathsf{A}^{V_1,V_2}(\mathsf{G}_1,\mathsf{G}_2)$ and $\mathtt{div}_\mathsf{A}^{\phi,\psi}(\mathsf{G}_1,\mathsf{G}_2)$. For the second condition, we proceed by contradiction; assume there are $\hat{\mathsf{R}} \in \operatorname{dom}\phi$ and $\hat{r} \in \hat{\mathsf{R}}$ such that $\tilde{l}_2 \cap \lambda_{\hat{r}}(\mathcal{E}_{\hat{r}-\psi(\hat{\mathsf{R}})}) \neq \emptyset$, then there are $e \in \bigcup_{\substack{\mathsf{R} \in \operatorname{cod}\phi,\\ r \in \mathsf{R}}} \lambda_r\big(\mathtt{min}\ (r - \psi(\phi^{-1}(\mathsf{R})))\big)$ and $e' \in \mathcal{E}_{\hat{r}-\psi(\hat{\mathsf{R}})}$ with the same label. Then the vertices $v \in \tilde{v}_2$ and $v' \in \tilde{v}_1$

respectively corresponding to e and e' should have the same label, contrary to the fact that $\tilde{v}_1 \sqcap \tilde{v}_2 = \emptyset$. The proof of the third condition is similar and therefore omitted.

We now show the $\Longleftarrow$ direction. Since, $[\![\mathsf{G}]\!] \neq \bot$ for $h \in \{1,2\}$ we have that $[\![\mathsf{G}_h]\!] \neq \bot$; hence, applying the inductive hypothesis, $\{\!|\mathsf{G}_h|\!\} \neq \bot$ and there is bijective correspondence between $[\![\mathsf{G}_h]\!]$ an $\Re_{\mathsf{G}_h}$. To show that $\{\!|\mathsf{G}|\!\} \neq \bot$, we have to prove that $\mathrm{wb}(\mathsf{G}_1, \mathsf{G}_2)$ holds. By hypothesis, $wb(\mathsf{G}_1, \mathsf{G}_2)$ holds, hence for each participant A of G there is an A-prefix map (ϕ, ψ) such that the point of divergence $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2) = (\tilde{l}_1, \tilde{l}_2)$ witnesses that A is either active or passive according to Definitions 13 and 14. By definition

$$\tilde{l}_1 = \bigcup_{\substack{\mathsf{R} \in \operatorname{dom}\phi, \\ r \in \mathsf{R}}} \lambda_r\big(\mathtt{min}\,\big(r - \psi(\mathsf{R})\big)\big) \quad \text{and} \quad \tilde{l}_2 = \bigcup_{\substack{\mathsf{R} \in \operatorname{cod}\phi, \\ r \in \mathsf{R}}} \lambda_r\big(\mathtt{min}\,\big(r - \psi(\phi^{-1}(\mathsf{R}))\big)\big)$$

Let V_1 and V_2 partitions of subsets of vertexes of G_1 and G_2 as in the proof of Lemma 20 and $\mathsf{div}_{\mathsf{A}}^{V_1,V_2}(\mathsf{G}_1, \mathsf{G}_2) = (\tilde{v}_1, \tilde{v}_2)$. We show that $\tilde{v}_h$ is in a bijective label-preserving correspondence with $\tilde{l}_h$ for $h \in \{1,2\}$. (Recall that, by definition of A-reflectivity, $V_1 \cap \mathcal{K} = \emptyset$.) By definition

$$\tilde{v}_1 = \bigcup \mathsf{cs}(\mathsf{fst}\,(\{\!|\mathsf{G}_1|\!\}_{@\mathsf{A}})) \setminus \bigcup V_1 \qquad \text{and} \qquad \tilde{v}_2 = \bigcup \mathsf{cs}(\mathsf{fst}\,(\{\!|\mathsf{G}_2|\!\}_{@\mathsf{A}})) \setminus \bigcup V_2$$

For each $l \in \tilde{l}_1$ there must be $\mathsf{R} \in \operatorname{dom}\phi$, $r \in \mathsf{R}$, and $e \in r - \psi(\mathsf{R})$ such that $\lambda_r(e) = l$ and no other event preceding e in $r - \psi(\mathsf{R})$ is a communication of A. Therefore, there is a vertex v in $\{\!|\breve{r}|\!\}$ that corresponds to e; moreover, V_1 is dijoint from $\{\!|\mathsf{G}_1(\breve{r})|\!\}$, hence by Theorem 4, $v \in \tilde{v}_1$. For the converse, let $\rho \in \Re_{\mathsf{G}_1}$ be such that $v \in \{\!|\mathsf{G}_1(\rho)|\!\}$ and $v \in \tilde{v}_1$. Also let $\mathsf{R} \in \operatorname{dom}\phi$ be such that $\breve{\rho} \in \mathsf{R}$ and let e be the vertex of $\breve{\rho}$ that corresponds to v. Then, $e \notin \psi(\mathsf{R})$ since $v \notin \bigcup V_1$ and again by Theorem 4, it must be $\lambda_r(e) \in \tilde{l}_1$. By the bijectivity of ϕ and A-reflectivity and using a similar argument, we can find a label-preserving bijection between $\tilde{v}_2$ and $\tilde{l}_2$. The proof that if A is active in the sense of Definition 13 then A is active in the sense of Definition 15 uses the bijection correspondence and follows the same strategy of the $\Longrightarrow$ directions. By Definition 14, A is passive iff

1. $\tilde{l}_1 \cup \tilde{l}_2 \subseteq \mathcal{L}^?$ and $\tilde{l}_1 = \emptyset \iff \tilde{l}_2 = \emptyset$
2. $\forall \mathsf{R} \in \operatorname{dom}\phi, r \in \mathsf{R} : \tilde{l}_2 \cap \lambda_r(\mathcal{E}_{r-\psi(\mathsf{R})}) = \emptyset$
3. $\forall \mathsf{R} \in \operatorname{cod}\phi, r \in \mathsf{R} : \tilde{l}_1 \cap \lambda_r(\mathcal{E}_{r-\psi(\phi^{-1}(\mathsf{R}))}) = \emptyset$

while A is passive in the sense of Definition 16 iff

(a) $\tilde{v}_1 \sqcap \{v \in \mathsf{G}' \mid \nexists v' \in \tilde{v}_2 : v \sqsubseteq_{\mathsf{G}'} v'\} = \emptyset$
(b) $\tilde{v}_2 \sqcap \{v \in \mathsf{G} \mid \nexists v' \in \tilde{v}_1 : v \sqsubseteq_{\mathsf{G}} v'\} = \emptyset$
(c) $\tilde{v}_1 \cup \tilde{v}_2 \subseteq (\mathcal{L}^? \times \mathcal{K})$
(d) $\tilde{v}_1 = \emptyset \iff \tilde{v}_2 = \emptyset$

Conditions (c) and (d) immediately follow from (1) bijective correspondence between $\mathsf{div}_{\mathsf{A}}^{V_1,V_2}(\mathsf{G}_1, \mathsf{G}_2)$ and $\mathsf{div}_{\mathsf{A}}^{\phi,\psi}(\mathsf{G}_1, \mathsf{G}_2)$. The proof of condition (a) proceed by contradiction. If (a) is false then there is $\hat{\rho} \in \Re_{\mathsf{G}_1}$ such that $\tilde{v}_2 \sqcap \{v \in \hat{\rho} \mid \nexists v' \in \tilde{v}_1 : v \sqsubseteq_{\mathsf{G}_1} v'\} \neq \emptyset$. Then there are $v \in \bigcup \mathsf{cs}(\mathsf{fst}\,(\{\!|\mathsf{G}_2|\!\}_{@\mathsf{A}})) \setminus \bigcup V_2$ and

$v' \in \{v \in \mathsf{G}_1 \mid \nexists v' \in \tilde{v}_1 : v \sqsubseteq_{\mathsf{G}_1} v'\}$ with the same label. By construction, the events e and e' that corresponds to v and v' have labels in $\tilde{l}_2$ and $\tilde{l}_1$. This contradict to the fact that $\tilde{l}_1 \cap \tilde{l}_2 = \emptyset$. The proof of (b) is similar and therefore omitted. □