Realizable causal-consistent reversible choreographies for systems with first-in-first-out communication channels

Monika Kapus-Kolar

Jožef Stefan Institute, Department of Communication Systems, Jamova 39, SI-1111 Ljubljana, Slovenia

Abstract

We re-engineer a pomset-based abstract semantics (and the associated semantic constraints) recently proposed for compositionally specified choreographies for systems of components communicating over first-in-first-out channels. We prove that the original semantics over-specifies components' behaviour and that for this, but not only this reason, the original semantic constraints are insufficient for the realizability of choreographies. We remove the problematic over-specification in the semantics, extend the semantics to explicitly specified interaction pomset sets, define an abstract syntax of choreographies and rephrase choreography semantics in terms of it, and newly provide a syntax-independent definition of choreography well-formedness. We prove that choreographies well-formed in the new sense are realizable and under a certain additional condition also causal-consistent reversible. Devising a set of rules for inferring well-formedness of choreographies compositionally, we correct the semantic constraints originally claimed sufficient for operands of individual composition operators. Our constraints and our definition of choreography well-formedness are in certain ways also less restrictive than the original ones. In particular, we newly allow also choreographies exploiting accidental event orderings.

Keywords: Choreography, Semantics, Realizability, Causal-consistent reversibility, Pomset, Communicating state machine

1. Introduction

When designing a distributed application supposed to run on a given distributed system, a possible way to proceed is to first conceive a choreography, i.e. a model of interactions among the system components from the global point of view and of (possibly ambiguous) causal relationships between the interactions. Ideally, the choreography is realizable, i.e. component processes for its correct implementation can be obtained simply by its projection. Furthermore, it is desirable that the choreography is causal-consistent reversible, i.e. that its so obtained implementation is at every point of execution in principle ready to undo any of those, and only those past events e for which the following is true for every other past event e': If among the by the choreography specified candidate causal interpretations of the past, there exist also such stating that e' has causally depended on e, e' has already been undone. Here 'in principle' means in case that every system component for each of its steps specified by the choreography projection implements also its inverse. The possibility of causal-consistent event undoing is important in, for example, system debugging [1, 2], system recovery [3], optimistic parallel discrete event simulation [4, 5] and reversible control of robots [6].

The paper corrects and generalizes the work of Tuosto and Guanciale [7] (for which proofs have been provided in [8]) on compositional construction of realizable choreographies for systems in which (1) every communication channel is between a pair of two different components, (2) from any component to any other component, there is exactly one communication channel, and (3) every channel is an initially empty, infinite-capacity buffer in which messages are queued in the order of arrival and exactly the first in the queue is available for reception. Actually, Tuosto and Guanciale advocate choreographies that are highly abstract, i.e. without any detailed assumptions about the target distributed system. Still, when demonstrating the applicability of their approach, they concentrate on the

Email address: `monika.kapus-kolar@ijs.si` (Monika Kapus-Kolar)

first-in-first-out (FIFO) channels case, and in the present paper, we discuss only this part of their work. For a discussion of earlier approaches to choreography semantics, the reader can refer to [7]. There one can find also a comparison explaining why it pays to go abstract, which we in certain aspects do to an even larger extent than [7].

The elementary choreographies considered in [7] are empty and singleton sets of interaction instances. To execute an interaction means that on a certain channel, a certain message is sent (i.e. appended to the end of the queue), by the source component of the channel, and at some later point received (i.e. removed from the queue after it has become its first element), by the sink component of the channel. In other words, every interaction instance consists of a transmission instance and the corresponding reception instance.

As, unlike [7], we study also possibilities for backward execution of choreographies, we additionally assume that any given channel supports also the following operations: (1) To delete the last element of its message queue, which is an instance of the inverse of transmitting the message on the channel and assumed to be executed by the source component of the channel, and (2) to add an instance of a given message to the start of the queue, which is an instance of the inverse of receiving the message on the channel and assumed to be executed by the sink component of the channel.

The choreography composition operators considered in [7] are parallel composition (the operator specifies concurrent execution of its operands), sequential composition at each component (the operator specifies that the operands are executed concurrently, except that at each component, transmission and reception instances belonging to the second operand are delayed until the component has completed its duties in the first operand), and choice (the operator specifies that either the first or the second operand is executed). Actually, iteration is also discussed, but only briefly and informally. The graphical syntax of the choreography specification language of [7] is presented in the Legend in Fig. 1, whereas the Figs. 1(a-f) present some example choreographies.

In [7], semantics and projection are defined only for those choreographies which the paper considers well-formed. For this, a choreography must be elementary or a composition of well-formed choreographies whose semantics satisfies the constraints which [7] defines for operands of the particular composition operator. The semantics of a well-formed choreography is in [7] defined basically as a set of partially ordered multisets (pomsets) of actions, i.e. transmissions and receptions. The component processes generated by the projection of a well-formed choreography are in [7] defined as (initialized, initially connected and deterministic) communicating state machines (CSMs) [9]. The CSM system of a given choreography is considered correct (i.e. the choreography is considered realizable) if, starting with every constituent CSM in the initial state (and every channel empty), it is (on the assumed channels) unable to reach a deadlock (i.e. a state in which it cannot proceed in spite of some channels non-empty or some constituent CSMs in a state with further actions specified) or execute a global action sequence not specified by the choreography semantics.

All choreographies well-formed in the sense of [7] are allegedly realizable and causally unambiguous [8]. As such, they have been employed as a basis for the conception of choreographies for automated recovery [10]. The choreographies in the Figs. 1(d-f) are well-formed in the sense of [7], whereas those in the Figs. 1(a-c) are not. Nevertheless, we applied (the in [8] defined semantics-based version of) the projection function of [7] to all the choreographies. The thereby obtained CSM systems are presented in the Figs. 1(a′-f′), respectively (for a given message m, $!m$ denotes transmission, and $?m$ reception). In the systems in the Figs. 1(d′-f′), black colour singles out those states which the constituent CSMs have after the system runs !z!w!x!y, !a?a!y?y and !z?z!b!x?x, respectively.

The main contribution of [7] is allegedly a relaxation of the usual constraints (such as defined, for example, in [11, 12, 13]) for operands of the choice operator. Consider, however, the choreography in Fig. 1(a). The operands of its choice operator satisfy the usual constraints, which suffices for the realizability of the choreography, but the choice constraints of [7] are not satisfied, meaning that they are not strictly weaker than the former. The choreographies in the Figs. 1(b,c) are examples of two further interesting kinds of choreographies which are realizable, but unacceptable for [7]. The one in Fig. 1(b) belongs to realizable choreographies comprising also unrealizable sub-choreographies, whereas the one in Fig. 1(c) belongs to choreographies exploiting also information available only thanks to accidental event orderings (see the Example 1 below), which makes it a choreography whose CSM system cannot be faithfully represented by a synchronous transition system and is therefore inaccessible to efficient verification approaches such as that of [14].

Example 1. Consider the choreography $\mathsf{G} = (\mathsf{G}_1 + \mathsf{G}_2) + \mathsf{G}_3$ in Fig. 1(c) and its CSM system in Fig. 1(c′). G_3 specifies that the interactions $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}$ and $\mathsf{B} \xrightarrow{\mathsf{y}} \mathsf{A}$ are concurrent and together enable $\mathsf{A} \xrightarrow{\mathsf{v}} \mathsf{B}$ and $\mathsf{B} \xrightarrow{\mathsf{u}} \mathsf{A}$. It is, however, possible that

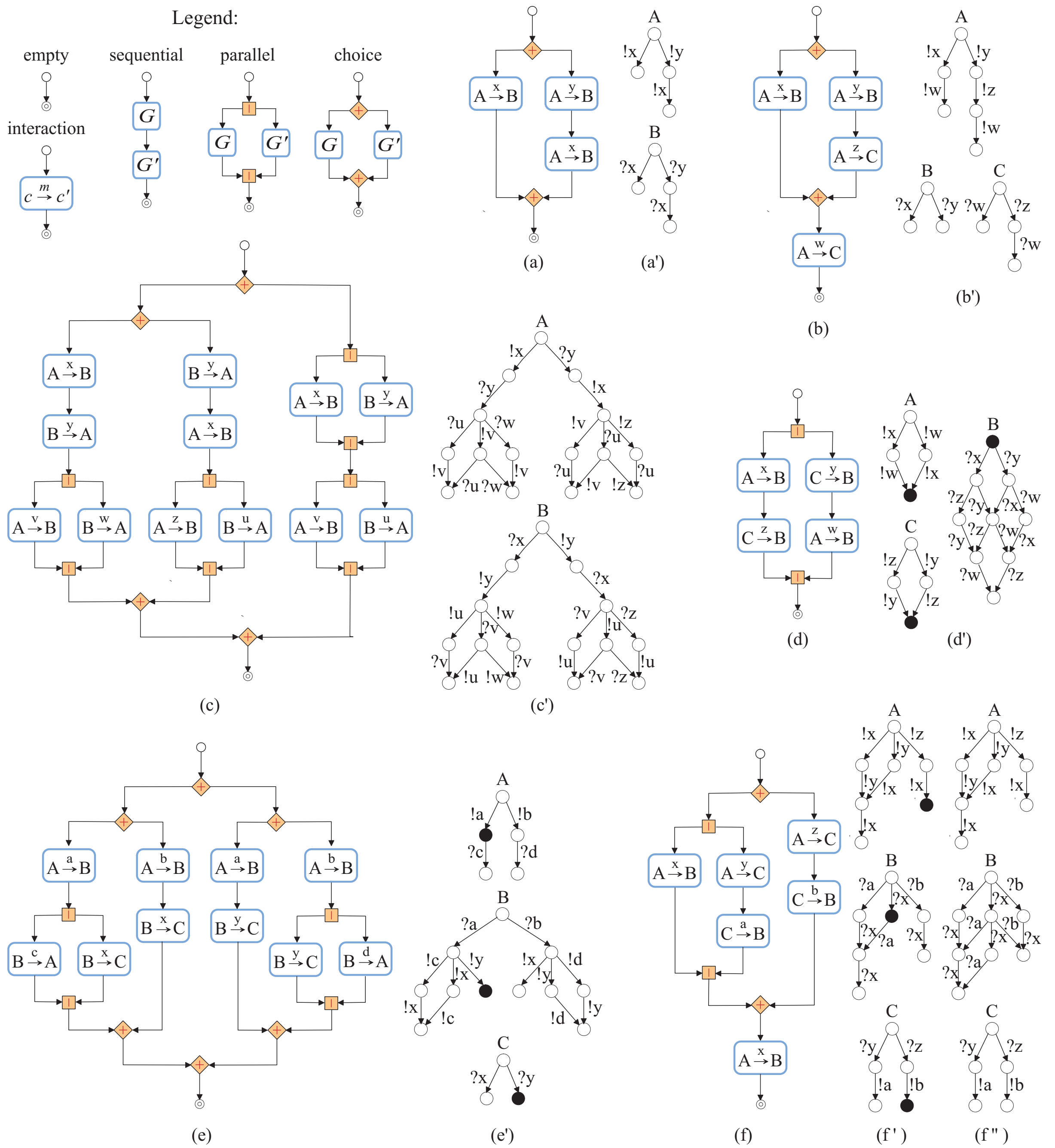

Figure 1: The choreography grammar of [7], (a-f) six choreographies, (a′-f′) the CSM systems obtained by projecting them as defined in [8], and (f′′) the proposed new CSM system for the choreography (f). In the CSM systems (d′-f′), black colour singles out those states which the CSMs have after the system runs !z!w!x!y, !a?a!y?y and !z?z!b!x?x, respectively.

their actions are executed in the order !x?x!y?y, i.e. as if $B \xrightarrow{y} A$ causally depended on $A \xrightarrow{x} B$, or in the order !y?y!x?x, i.e. as if $A \xrightarrow{x} B$ causally depended on $B \xrightarrow{y} A$. G_1 specifies that in case that the first alternative causal interpretation of the run-time actions' order is possible, it is legal that $B \xrightarrow{w} A$ follows instead of $B \xrightarrow{u} A$. G_2 specifies that in case that the second alternative causal interpretation of the run-time actions' order is possible, it is legal that $A \xrightarrow{z} B$ follows instead of $\mathsf{A}\xrightarrow{\mathsf{v}}\mathsf{B}$.

For the component A, for example, the above means that it might or might not receive the information in which order !y and ?x occur in the particular system run. If it accidentally executes !x before ?y, it cannot deduce the order, whereas in the opposite case, it can deduce that the order is !y?x. In the second case, A knows that it can continue more liberally than in the first case, namely be unready for the message w and freely choose whether to send v or z.

The choreographies in the Figs. 1(a-c) motivated us to reconsider the concept of well-formedness as defined in [7]. A detailed study of the constraints in [7] associated with individual composition operators revealed that they are more problematic than we expected. Namely, for none of the three considered operators, they are sufficient for the realizability of the composition. A proof of this are the choreographies in the Fig. 1(d-f): The operands of their top-most operator (parallel composition, choice or sequential composition, respectively) are realizable, whereas the choreographies as a whole are well-formed in the sense of [7], but not realizable. For example, each of the three system runs exposed in the Figs. 1(d′-f′) leads to a deadlock (what goes wrong in the three runs is described in Section 4, in the Examples 21, 20 and 22, respectively).

Identification of the above described problems is our first contribution. In the following, we gradually identify also their sources, among which an influential one is a problematic feature of the choreography semantics of [7]. As another contribution, we, in Section 3, redefine the semantics and well-formedness, for terms of the choreography grammar of [7] generalized by defining that interaction pomset sets are also elementary choreographies. Our definitions of the semantics, projection and well-formedness of (generalized) choreographies are independent from the concrete choreography syntax, which in [7] is not the case. We prove that choreographies well-formed in the new sense are realizable and under a certain additional condition also causal-consistent reversible. As the third contribution, we, in Section 4, prove some rules for inferring choreography well-formedness, and thereby virtually redefine constraints for operands of individual composition operators. Old and new choreography semantics and operand constraints are compared conceptually and on the example choreographies in the Figs. 1(a-f), for which (non-)well-formedness newly coincides with (un)realizability. First of all, however, we present, in Section 2, the basic concepts and notations used in the subsequent sections.

2. Basic concepts and notations

2.1. (Inter)actions and their instances

We assume that choreographies are designed for a system with component set C. Elements of C are ranged over by c. Communication channels are assumed to be as defined in Section 1, where for given different components c and c', the channel leading from c to c' is denoted as cc'. Messages are ranged over by m.

An interaction in which a given message m is passed over a given channel cc' is denoted as $c \xrightarrow{m} c'$, whereas its constituent transmission and reception are denoted as $cc'!m$ and $cc'?m$, respectively, with the channel identifier cc' possibly omitted in case of evident from the context.

The universes of transmissions and receptions are denoted as, respectively, $\mathcal{L}^!$ and $\mathcal{L}^?$. The universe $\mathcal{L}^! \cup \mathcal{L}^?$ of actions is denoted as $\mathcal{L}$. Actions are ranged over by a, action sets by $\mathcal{A}$, action sequences by α, and action sequence sets by A.

The universe $\{a^{-1}|a \in \mathcal{L}\}$ of action inverses (shortly i-actions) is denoted as $\mathcal{L}^{-1}$. The universe $\mathcal{L} \cup \mathcal{L}^{-1}$ of (i-)actions is ranged over by b. Sequences of (i-)actions are ranged over by β. An empty sequence is denoted as ϵ.

For a given action sequence α of the form $(cc'!m_i)_{i=1...k}$ or $(cc'?m_i)_{i=1...k}$, $\mathrm{ms}(\alpha)$ denotes the message sequence $(m_i)_{i=1...k}$.

For given action sequence α and action a of which it comprises an instance, $\mathrm{rlst}(\alpha, a)$ denotes the action sequence obtained from α by removing its last instance of a.

For a given action sequence α, $\mathrm{pf}(\alpha)$ denotes the set of all its prefixes. For a given action sequence set A, $\max(A)$ denotes the action sequence set $\{\alpha|(\alpha \in A) \wedge \nexists\alpha' \in (A \setminus \{\alpha\}) : (\alpha \in \mathrm{pf}(\alpha'))\}$.

(Inter)action instances are alternatively called events. Events are ranged over by e, event sets by $\mathcal{E}$, and event sequences by ε.

For a given event e, $\lambda(e)$ denotes its label, i.e. the (inter)action of which it is an instance. For a given action instance sequence $\varepsilon = (e_i)_{i=1...k}$, $\mathrm{asq}(\varepsilon)$ denotes the action sequence $(\lambda(e_i))_{i=1...k}$.

Interaction instances and their sets are alternatively (to expose their nature) ranged over by, respectively, g and $\mathcal{G}$. For a given interaction instance g, $e_g^!$ denotes the constituent transmission instance, and $e_g^?$ the constituent reception instance.

2.2. *Partially ordered sets of (inter)action instances*

A binary relation on a given event set $\mathcal{E}$ is a subset of $\mathcal{E} \times \mathcal{E}$. If it is reflexive, anti-symmetric and transitive, it is called partial order. The transitive closure of a given binary relation R is denoted as $R^\star$.

A partially ordered set of events (shortly poset) is an event set $\mathcal{E}$ endowed with a partial order $\leq$ and denoted as $(\mathcal{E}, \leq)$. Posets are ranged over by p, and their sets by $\mathcal{P}$.

If for given posets $p = (\mathcal{E}, \leq)$ and $p' = (\mathcal{E}', \leq')$ there exist bijections $\phi : \mathcal{E} \to \mathcal{E}'$ and $\phi' : \leq \to \leq'$ satisfying all the following:

(1) $\forall e \in \mathcal{E} : (\lambda(e) = \lambda(\phi(e)))$
(2) $\forall (e, e') \in \leq : (\phi'((e, e')) = (\phi(e), \phi(e')))$

then p and p' are isomorphic.

For a given poset $p = (\mathcal{E}, \leq)$, $\max(p)$ denotes the event set $\{e | (e \in \mathcal{E}) \wedge \nexists e' \in (\mathcal{E} \setminus \{e\}) : (e \leq e')\}$.

For a given poset $p = (\mathcal{E}, \leq)$, $\mathrm{esq}(p)$ denotes the set of all event sequences $(e_i)_{i=1...|\mathcal{E}|}$ that satisfy $(\mathcal{E} = \{e_i\}_{i=1...|\mathcal{E}|}) \wedge \forall 1 \leq i < j \leq |\mathcal{E}| : (e_j \nleq e_i)$.

For a given poset $p = (\mathcal{E}, \leq)$, $\mathrm{pf}(p)$ denotes the set of all its prefixes, i.e. the set of all posets $(\mathcal{E}', \leq')$ that satisfy $(\mathcal{E}' \subseteq \mathcal{E}) \wedge (\leq' = \leq \cap (\mathcal{E}' \times \mathcal{E}')) \wedge (\leq \cap ((\mathcal{E} \setminus \mathcal{E}') \times \mathcal{E}') = \emptyset)$.

For a given poset set $\mathcal{P}$, $\mathrm{pf}(\mathcal{P})$ denotes its prefix set $\{p | \exists p' \in \mathcal{P} : (p \in \mathrm{pf}(p'))\}$.

For a given poset set $\mathcal{P}$, $\max(\mathcal{P})$ denotes the poset set $\{p | (p \in \mathcal{P}) \wedge \nexists p' \in (\mathcal{P} \setminus \{p\}) : (p \in \mathrm{pf}(p'))\}$.

For a given poset set $\mathcal{P}$, $\mathrm{esq}(\mathcal{P})$ denotes the event sequence set $\{\varepsilon | \exists p \in \mathrm{pf}(\mathcal{P}) : (\varepsilon \in \mathrm{esq}(p))\}$.

For given poset p and action sequence α, $\mathrm{pf}(p, \alpha)$ denotes the set of all posets $p' \in \mathrm{pf}(p)$ whose $\mathrm{esq}(p')$ comprises an event sequence ε with $\mathrm{asq}(\varepsilon) = \alpha$.

For given poset set $\mathcal{P}$ and action sequence α, $\mathrm{pf}(\mathcal{P}, \alpha)$ denotes the poset set $\{p | \exists p' \in \mathcal{P} : (p \in \mathrm{pf}(p', \alpha))\}$.

For a given poset set $\mathcal{P}$, $\mathrm{asq}(\mathcal{P})$ denotes the action sequence set $\{\alpha | (\alpha \in \mathcal{L}^*) \wedge (\mathrm{pf}(\mathcal{P}, \alpha) \neq \emptyset)\}$

For a given poset p, $\mathrm{tri}(p)$ denotes the set of all triplets $((\mathcal{E}, \leq), \lambda(e), (\mathcal{E} \cup \{e\}, \leq'))$ that satisfy $((\mathcal{E} \cup \{e\}, \leq') \in \mathrm{pf}(p)) \wedge (e \notin \mathcal{E}) \wedge ((\mathcal{E}, \leq) \in \mathrm{pf}((\mathcal{E} \cup \{e\}, \leq')))$.

2.3. *Partially ordered multisets of (inter)actions*

An (inter)action pomset (shortly pomset) is an isomorphism class of posets. The isomorphism class to which a given poset $p = (\mathcal{E}, \leq)$ belongs is denoted as $[p]$ or $[\mathcal{E}, \leq]$. Pomsets are ranged over by r, and their sets by $\mathcal{R}$.

A natural way to discuss properties of a given pomset is to discuss properties of a representative of the class. Likewise, a natural way to define a composition operator for pomsets is to do it in terms of selected representatives of individual operands, taking care that the representatives are non-intersecting (inter)action sets. When discussing or combining pomset sets, one would proceed analogously. We therefore define the following families of pomset and pomset set representatives:

(1) For given pomset r and (possibly omitted) natural i, $\mathrm{po}_i(r) = (\mathcal{E}_{r,i}, \leq_{r,i})$ is the poset selected as the default representative of the class r (for the natural i), with $\mathcal{E}_{r,i} \cap \mathcal{E}_{r',i'} = \emptyset$ for every pomset r' and natural i' with $(r, i) \neq (r', i')$.
(2) For given pomset set $\mathcal{R}$ and (possibly omitted) natural i, $\mathrm{pos}_i(\mathcal{R})$ denotes the poset set $\{\mathrm{po}_i(r) | r \in \mathcal{R}\}$.

For a given pomset r, $\mathrm{pf}(r)$ denotes its prefix set $\{[p] | p \in \mathrm{pf}(\mathrm{po}(r))\}$.

For a given pomset set $\mathcal{R}$, $\mathrm{pf}(\mathcal{R})$ denotes its prefix set $\{[p] | p \in \mathrm{pf}(\mathrm{pos}(\mathcal{R}))\}$.

For given pomset set $\mathcal{R}$ and action sequence α, $\mathrm{pf}(\mathcal{R}, \alpha)$ denotes the pomset set $\{[p] | p \in \mathrm{pf}(\mathrm{pos}(\mathcal{R}), \alpha)\}$.

For a given action pomset set $\mathcal{R}$, $\mathrm{asq}(\mathcal{R})$ denotes the action sequence set $\mathrm{asq}(\mathrm{pos}(\mathcal{R}))$.

For a given action pomset set $\mathcal{R}$, $\lambda(\mathcal{R})$ denotes the action set $\{a | (a \in \mathcal{L}) \wedge \exists \alpha a \in \mathrm{asq}(\mathcal{R})\}$,

For a given action pomset set $\mathcal{R}$, $\mathrm{tri}(\mathcal{R})$ denotes the triplet set $\{([p], a, [p']) | \exists p'' \in \mathrm{pos}(\mathcal{R}) : ((p, a, p') \in \mathrm{tri}(p''))\}$.

2.4. State machines

An initialized and initially connected state automaton M whose individual transitions represent individual (i-)actions (shortly state machine) is defined by a triplet $(\mathcal{S}_M, s_M, \mathcal{T}_M)$ in which $\mathcal{S}_M$ is its state set, s_M its initial state, and $\mathcal{T}_M$ its transition set.

A state machine transition t is defined by a triplet (s_t, b_t, s'_t) in which s_t is the state in which it starts, s'_t is the state in which it ends, and b_t is the executed (i-)action.

If for given state machines M and M', there exist bijections $\phi : \mathcal{S}_M \to \mathcal{S}_{M'}$ and $\phi' : \mathcal{T}_M \to \mathcal{T}_{M'}$ satisfying all the following:

(1) $\phi(s_M) = s_{M'}$
(2) $\forall(s, a, s') \in \mathcal{T}_M : (\phi'((s, a, s')) = (\phi(s), a, \phi(s')))$

then M and M' are isomorphic.

For given state machine M, seq(M) denotes the (i-)action sequence set $\{(b_i)_{i=1\ldots k} | (k = 0) \vee \exists((s_i, b_i, s_{i+1}))_{i=1\ldots k} \in (\mathcal{T}_M)^+ : (s_1 = s_M)\}$.

For given deterministic state machine M and (i-)action sequence $\beta = (b_i)_{i=1\ldots k}$ in seq(M), $\delta_M(\beta)$ denotes the only member of the state set $\{s_{k+1} | ((k = 0) \wedge (s_{k+1} = s_M)) \vee \exists((s_i, b_i, s_{i+1}))_{i=1\ldots k} \in (\mathcal{T}_M)^+ : (s_1 = s_M)\}$.

For a given state machine M with no i-action transitions, rev(M) denotes the state machine $(\mathcal{S}_M, s_M, \mathcal{T}_M \cup \{(s', a^{-1}, s) | (s, a, s') \in \mathcal{T}_M\})$.

2.5. Projections

For a given component c, $\mathcal{L}_c$ denotes the set of all actions that are of the form $cc'!m$ or $c'c?m$.

For a given channel cc', $\mathcal{L}^!_{cc'}$ denotes the set of all actions of the form $cc'!m$, and $\mathcal{L}^?_{cc'}$ the set of all actions of the form $cc'?m$.

For given action sequence α and action set $\mathcal{A}$, $\alpha\lfloor_{\mathcal{A}}$ denotes α after the deletion of every element that is not in $\mathcal{A}$.

For given action sequence α and component c, $\alpha\lfloor_c$ denotes $\alpha\lfloor_{\mathcal{L}_c}$.

$\mathcal{F}$ denotes the universe of all action sequences α that for every channel cc' satisfy $\mathrm{ms}(\alpha\lfloor_{\mathcal{L}^?_{cc'}}) \in \mathrm{pf}(\mathrm{ms}(\alpha\lfloor_{\mathcal{L}^!_{cc'}}))$, i.e. respect the FIFO rule.

For given action instance set $\mathcal{E}$ and action set $\mathcal{A}$, $\mathcal{E}\lfloor_{\mathcal{A}}$ denotes the event set $\{e | (e \in \mathcal{E}) \wedge (\lambda(e) \in \mathcal{A})\}$.

For given action instance set $\mathcal{E}$ and component c, $\mathcal{E}\lfloor_c$ denotes $\mathcal{E}\lfloor_{\mathcal{L}_c}$.

For given action instance sequence ε and action set $\mathcal{A}$, $\varepsilon\lfloor_{\mathcal{A}}$ denotes ε after the deletion of every element e with $\lambda(e) \notin \mathcal{A}$.

For given action instance sequence ε and component c, $\varepsilon\lfloor_c$ denotes $\varepsilon\lfloor_{\mathcal{L}_c}$.

For given action instance poset $p = (\mathcal{E}, \leq)$ and component c, $p\lfloor_c$ denotes the poset $(\mathcal{E}\lfloor_c, \leq \cap((\mathcal{E}\lfloor_c) \times (\mathcal{E}\lfloor_c)))$.

For given action instance poset set $\mathcal{P}$ and component c, $\mathcal{P}\lfloor_c$ denotes the poset set $\{p\lfloor_c | p \in \mathcal{P}\}$.

For given poset set $\mathcal{P}$ and event e, $\mathcal{P} \setminus e$ denotes the poset set $\{(\mathcal{E} \setminus \{e\}, \leq \cap((\mathcal{E} \setminus \{e\}) \times (\mathcal{E} \setminus \{e\}))) | (\mathcal{E}, \leq) \in \mathcal{P}\}$.

For a given action pomset set $\mathcal{R}$, $\mathrm{asq}_{\mathcal{F}}(\mathcal{R})$ denotes the action sequence set $\mathrm{asq}(\mathcal{R}) \cap \mathcal{F}$.

For given action pomset set $\mathcal{R}$ and component c, $\mathcal{R}\lfloor_c$ denotes the pomset set $\{[p] | p \in \mathrm{pos}(\mathcal{R})\lfloor_c\}$.

For given action pomset set $\mathcal{R}$ and component c, $\mathrm{asq}_c(\mathcal{R})$ denotes the action sequence set $\mathrm{asq}(\mathcal{R}\lfloor_c)$.

2.6. Causal interpretations

This section is very important: As we later on define that the semantics of a given choreography G is an action pomset set $[\![G]\!]$, with the legal global action sequences of G those in $\mathrm{asq}([\![G]\!])$, here we actually define (see how the Definition 3 in Section 3.2 uses the below defined concepts) (1) how individual action sequences in $\mathrm{asq}([\![G]\!])$ inherit the causal relationships of the corresponding event sequences of the default representatives of individual pomsets in $[\![G]\!]$ and (2) the CSM that is the projection of G onto a given component c. In the definition of the latter, we take care that for every action sequence $\alpha \in \mathrm{asq}_c([\![G]\!])$, the resulting state of the CSM corresponds to what individual pomsets in $[\![G]\!]\lfloor_c$ specify about the causal relationships between individual action instances in α.

Note that every action a denotes the class of all action instances e with $\lambda(e) = a$. For given action a and natural i, let $e_{a,i}$ denote the default representative of the class a for the natural i.

Note that every action sequence α denotes the class of all action instance sequences ε with $\mathrm{asq}(\varepsilon) = \alpha$. A natural way to causally interpret a given action sequence α is to interpret the default representative of the class. For given action sequence $\alpha = (a_i)_{i=1\ldots k}$ and natural $1 \leq j \leq k$, let $\mathrm{ev}(\alpha, j)$ denote the event $e_{\alpha_j, |(a_i)_{i=1\ldots j}|_{\{a_j\}}|}$. For a given action sequence α, the default representative of the class, denoted as $\mathrm{esq}(\alpha)$, is the action instance sequence $(\mathrm{ev}(\alpha, i))_{i=1\ldots|\alpha|}$. Note that in the representative, any given i^{th} instance of a given action a is conveniently called $e_{a,i}$. For a given action sequence α, $\mathcal{E}_\alpha$ denotes the action instance set $\{\mathrm{ev}(\alpha, i)\}_{i=1\ldots|\alpha|}$.

For given action instance poset $p = (\mathcal{E}, \leq)$ and action instance sequence $\varepsilon = (e_i)_{1\ldots k}$ in $\mathrm{esq}(\mathrm{pf}(p))$, let $\mathrm{po}(p, \varepsilon)$ denote the action instance poset $(\mathcal{E}_{\mathrm{asq}(\varepsilon)}, \leq')$ with $\leq' = \{(\mathrm{ev}(\mathrm{asq}(\varepsilon), i), \mathrm{ev}(\mathrm{asq}(\varepsilon), j)) | (1 \leq i \leq j \leq k) \wedge (e_i \leq e_j)\}$. Note that $\mathrm{po}(p, \varepsilon)$ virtually refers to the events in ε and defines for them the same partial order as the corresponding prefix of p, but brings the convenience that for any given $1 \leq i \leq k$, the i^{th} event in ε is called by the name of the i^{th} event in the default representative $\mathrm{esq}(\mathrm{asq}(\varepsilon))$ of the class $\mathrm{asq}(\varepsilon)$, which makes $\leq'$ the partial order which the pair (p, ε) defines for the event set $\mathcal{E}_{\mathrm{asq}(\varepsilon)}$.

For given action instance poset p and action sequence α, let $\mathrm{ci}_p(\alpha)$ denote the action instance poset set $\{\mathrm{po}(p, \varepsilon) | (\varepsilon \in \mathrm{esq}(\mathrm{pf}(p))) \wedge (\{\mathrm{asq}(\varepsilon)\} = \max(\mathrm{pf}(\alpha) \cap \mathrm{asq}(\{p\})))\}$. Informally, to compute $\mathrm{ci}_p(\alpha)$, one would take the longest prefix of α that complies to p, find all the possible ways for interpreting the prefix as a legal event sequence ε of p, and for every such ε take the poset $\mathrm{po}(p, \varepsilon)$, because the latter can be regarded as one of the (partial) causal interpretations which p specifies for α (hence the name $\mathrm{ci}_p(\alpha)$).

The set of all causal interpretations which a given action pomset set $\mathcal{R}$ (e.g. the semantics of a given choreography) specifies for a given action sequence α, denoted as $\mathrm{ci}_{\mathcal{R}}(\alpha)$, is the action instance poset set $\{p | \exists r \in \mathcal{R} : (p \in \mathrm{ci}_{\mathrm{po}(r)}(\alpha))\}$.

If for a given action pomset set $\mathcal{R}$, there is an action sequence $\alpha \in \mathrm{asq}(\mathcal{R})$ satisfying $|\max(\mathrm{ci}_{\mathcal{R}}(\alpha))| > 1$, then $\mathcal{R}$ (and any choreography whose semantics it is) is called causally ambiguous.

If for a given action pomset set $\mathcal{R}$, there exist a component c and an action sequence $\alpha \in \mathrm{asq}_c(\mathcal{R})$ satisfying $|\max(\mathrm{ci}_{\mathcal{R}|_c}(\alpha))| > 1$, then $\mathcal{R}$ (and any choreography whose semantics it is) is called locally causally ambiguous.

The (minimally liberal) cumulative causal interpretation which a given action pomset set $\mathcal{R}$ specifies for a given action sequence α, denoted as $\mathrm{cci}_{\mathcal{R}}(\alpha)$, is the action instance poset
$(\{e | \exists(\mathcal{E}, \leq) \in \mathrm{ci}_{\mathcal{R}}(\alpha) : (e \in \mathcal{E})\}, \{(e, e') | \exists(\mathcal{E}, \leq) \in \mathrm{ci}_{\mathcal{R}}(\alpha) : ((e, e') \in \leq)\}^\star)$.

For given action pomset set $\mathcal{R}$ and action sequence α, $\max_{\mathcal{R}}(\alpha)$ denotes the event set $\max(\mathrm{cci}_{\mathcal{R}}(\alpha))$.

For given action pomset set $\mathcal{R}$ and action sequence α, $\mathrm{civ}_{\mathcal{R}}(\alpha)$ denotes the action instance poset set vector $(\mathrm{ci}_{\mathrm{po}(r)}(\alpha))_{r \in \mathcal{R}}$.

For given action pomset set $\mathcal{R}$ and component c, $\mathrm{sm}_c(\mathcal{R})$ denotes a deterministic state machine M satisfying $(\mathrm{seq}(M) = \mathrm{asq}_c(\mathcal{R})) \wedge \forall \alpha \in \mathrm{asq}_c(\mathcal{R}) : (\delta_M(\alpha) = \mathrm{civ}_{\mathcal{R}|_c}(\alpha))$.

3. Well-formed generalized choreographies

3.1. Generalized choreographies and their normal form

Definition 1 (Generalized choreographies). Generalized choreographies (shortly choreographies) are terms derived by the following grammar (parentheses not necessary for disambiguation can be omitted):

$G ::= \mathbf{0} \mid c \xrightarrow{m} c' \mid \mathcal{R} \mid (G_1 | G_2) \mid (G_1 ; G_2) \mid (G_1 + G_2)$

In the grammar, $\mathbf{0}$ denotes an empty choreography, $c \xrightarrow{m} c'$ is assumed to be an interaction, $\mathcal{R}$ is assumed to be a non-empty set of interaction pomsets, '|' is the parallel composition operator, ';' is the sequential composition operator, and '+' is the choice operator.

To abstract away from the concrete syntax of choreographies, we define for them a normal form:

Definition 2 (Normal form of choreographies). The normal form of a choreography G, denoted as $\langle\!\langle G \rangle\!\rangle$, is a non-empty set of interaction pomsets. For our six choreography types, it is defined, respectively, as follows:

$\langle\!\langle \mathbf{0} \rangle\!\rangle = \{[\{\}, \{\}]\}$

$\langle\!\langle c \xrightarrow{m} c' \rangle\!\rangle = \{[\{g\}, \{(g, g)\}]\}$ where g is an instance of $c \xrightarrow{m} c'$.

$\langle\!\langle \mathcal{R} \rangle\!\rangle = \mathcal{R}$.

$\langle\!\langle G_1 | G_2 \rangle\!\rangle = \{[\mathcal{G}_1 \cup \mathcal{G}_2, \leq_1 \cup \leq_2] | ((\mathcal{G}_1, \leq_1), (\mathcal{G}_2, \leq_2)) \in \mathrm{pos}_1(\langle\!\langle G_1 \rangle\!\rangle) \times \mathrm{pos}_2(\langle\!\langle G_2 \rangle\!\rangle)\}$

$\langle\!\langle G_1 ; G_2 \rangle\!\rangle = \{[\mathcal{G}_1 \cup \mathcal{G}_2, (\leq_1 \cup \leq_2 \cup (\mathcal{G}_1 \times \mathcal{G}_2))^\star] | ((\mathcal{G}_1, \leq_2), (\mathcal{G}_2, \leq_2)) \in \mathrm{pos}_1(\langle\!\langle G_1 \rangle\!\rangle) \times \mathrm{pos}_2(\langle\!\langle G_2 \rangle\!\rangle)\}$

$\langle\!\langle G_1 + G_2 \rangle\!\rangle = \langle\!\langle G_1 \rangle\!\rangle \cup \langle\!\langle G_2 \rangle\!\rangle$

The pomsets in the normal form of a given choreography G are the alternatives between which the system is supposed to choose when executing G. For an empty choreography, the only alternative is to execute no interaction instances. For a choreography $c \xrightarrow{m} c'$, the only alternative is to execute an instance of $c \xrightarrow{m} c'$. The alternatives of a $G_1 + G_2$ are the alternatives of G_1 and the alternatives of G_2. The alternatives of a $G_1|G_2$ are all those defined as parallel composition of an alternative of G_1 and an alternative of G_2. The alternatives of a $G_1; G_2$ are all those defined as strict sequential composition of an alternative of G_1 and an alternative of G_2 (at the high level of abstraction adopted in $\langle\!\langle G_1; G_2 \rangle\!\rangle$, it is not visible that strict sequencing of a pair of interaction instances in general does not preclude some concurrency of their constituent action instances). The alternatives of an interaction pomset set $\mathcal{R}$ are the pomsets in $\mathcal{R}$ (i.e., choreographies of this type are by definition already in the normal form, which reveals that the remaining five choreography types are just syntax sugar).

Example 2. For the choreography $\mathsf{G} = \mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}|(\mathsf{B} \xrightarrow{\mathsf{y}} \mathsf{C} + (\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}; (\mathsf{B} \xrightarrow{\mathsf{z}} \mathsf{A}|\mathsf{A} \xrightarrow{\mathsf{z}} \mathsf{C})))$, $\langle\!\langle \mathsf{G} \rangle\!\rangle$ is a set consisting of two pomsets. In the first one, there are an $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}$ and a $\mathsf{B} \xrightarrow{\mathsf{y}} \mathsf{C}$, unordered. In the second one, there are two $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}$, a $\mathsf{B} \xrightarrow{\mathsf{z}} \mathsf{A}$ and an $\mathsf{A} \xrightarrow{\mathsf{z}} \mathsf{C}$, where the only ordering is that one of the $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}$ is before the $\mathsf{B} \xrightarrow{\mathsf{z}} \mathsf{A}$ and the $\mathsf{A} \xrightarrow{\mathsf{z}} \mathsf{C}$.

3.2. Choreography semantics

The semantics of choreographies is in [7] defined as a function of their concrete syntax, whereas we define it as a function of their normal form. Namely, recall that in the normal form $\langle\!\langle G \rangle\!\rangle$ of a given choreography G, each of the constituent alternatives of G is represented by a pomset specifying the alternative very abstractly, in terms of interactions. To obtain the semantics of G, we refine every pomset r in $\langle\!\langle G \rangle\!\rangle$ into a pomset r' specifying the alternative of G less abstractly, in terms of actions. To obtain r', we take the interaction instance poset p that is the default representative of the isomorphism class r, refine it into the action instance poset $[\![p]\!]$ below (see Definition 4) defined as the semantics of p, and set r' to the isomorphism class to which p' belongs:

Definition 3 (Semantics of choreographies). The semantics of a given choreography G, denoted as $[\![G]\!]$, is the action pomset set $\{[[\![p]\!]] | p \in \mathrm{pos}(\langle\!\langle G \rangle\!\rangle)\}$. More precisely:

(1) The projection of G onto a given component c is the state machine $\mathrm{sm}_c([\![G]\!])$.
(2) The action sequences which the CSM system of G is allowed to execute are those in $\mathrm{asq}([\![G]\!])$.
(3) The causal interpretations supposed to be respected when undoing action instances of a given action sequence $\alpha \in \mathrm{asq}([\![G]\!])$, i.e. the candidate causal interpretations of α, are the posets in $\mathrm{ci}_{[\![G]\!]}(\alpha)$. In other words, one has to respect the cumulative causal interpretation $\mathrm{cci}_{[\![G]\!]}(\alpha)$. In other words, in the system state resulting from α, the undoing of the element of α in a given i^{th} position is allowed exactly if $\mathrm{ev}(\alpha, i) \in \max_{[\![G]\!]}(\alpha)$.

In our semantics of a given interaction instance poset, each of the interaction instances is represented by its constituent action instances, and in the resulting action instance set, two different members e and e' are considered directly ordered (there is also ordering because of the transitivity of the ordering relation) exactly if either (1) they are the transmission instance and the reception instance of the same interaction instance or (2) the interaction instances to which they belong are ordered and the ordering is inherited. The inheritance is assumed to exist exactly if a certain predicate *Ord* below (see Section 3.4) (re)defined on action pairs is true on the pair of the actions of which e and e' are instances:

Definition 4 (Semantics of interaction instance posets). The semantics of a given interaction instance poset $p = (\mathcal{G}, \leq)$, denoted as $[\![p]\!]$, is the action instance poset

$$(\bigcup_{g \in \mathcal{G}} \{e^!_g, e^?_g\}, ((\bigcup_{g \in \mathcal{G}} \{(e^!_g, e^!_g), (e^!_g, e^?_g), (e^?_g, e^?_g)\}) \cup \{(e, e') | \exists (g, g') \in \leq : ((g \neq g') \wedge ((e, e') \in \{e^!_g, e^?_g\} \times \{e^!_{g'}, e^?_{g'}\}) \wedge Ord(\lambda(e), \lambda(e')))\})^{\star}).$$

In different contexts, different definitions of the predicate *Ord* may be meaningful. If one defines that $Ord(a, a')$ for a given action pair (a, a') is true exactly if there is a component c with $(a, a') \in \mathcal{L}_c \times \mathcal{L}_c$, i.e. a component able to secure that a given instance of a' is delayed until after a given instance of a, our choreography semantics becomes that of [7] (except that we have written its definition in a more abstract style). In other words, [7] virtually assumes the above version of *Ord*. As for (the in [8] defined semantics-based version of) the choreography projection function

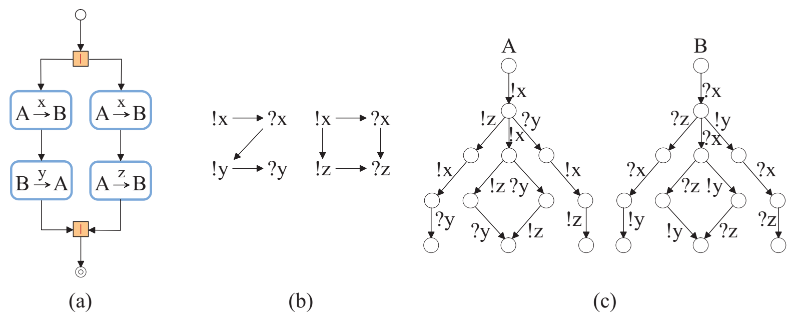

Figure 2: (a) A choreography, (b) the specified action pomset, and (c) the CSM system obtained by projection.

of Tuosto and Guanciale, it can be for our purposes considered identical to ours, because the CSM which it returns for a given component c is for locally causally unambiguous choreographies G ([7] allows no others) isomorphic to $\mathrm{sm}_c([\![G]\!])$.

Example 3. Consider the choreography $\mathsf{G} = (\mathsf{G}_1 + \mathsf{G}_2); \mathsf{G}_3$ in Fig. 1(f). $\langle\!\langle \mathsf{G} \rangle\!\rangle$ consists of two interaction pomsets, an r_1 representing the alternative $\mathsf{G}_1; \mathsf{G}_3$ and an r_2 representing the alternative $\mathsf{G}_2; \mathsf{G}_3$. r_2 specifies $\mathsf{A} \xrightarrow{\mathsf{z}} \mathsf{C}$ followed by $\mathsf{C} \xrightarrow{\mathsf{b}} \mathsf{B}$ followed by $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}$. In $[\![\mathsf{G}]\!]$, the pomset is refined into (the isomorphism class of) the action set $\{!\mathsf{z}, ?\mathsf{z}, !\mathsf{b}, ?\mathsf{b}, !\mathsf{x}, ?\mathsf{x}\}$ endowed with a partial order $\leq_2$ depending on the assumed version of *Ord*. For every version, $!\mathsf{z} \leq_2 ?\mathsf{z}$, $!\mathsf{b} \leq_2 ?\mathsf{b}$ and $!\mathsf{x} \leq_2 ?\mathsf{x}$. In case of the *Ord* of [7], also $!\mathsf{z} \leq_2 ?\mathsf{x}$, $?\mathsf{b} \leq_2 ?\mathsf{x}$ and $?\mathsf{z} \leq_2 !\mathsf{b}$, meaning that in this case, $!\mathsf{b}$ and $!\mathsf{x}$ are concurrent, whereas for $?\mathsf{x}$, $\leq_2$ specifies that B must delay it until after $?\mathsf{b}$, in spite of the fact that the message x from A possibly arrives to B before the message b from C. Similarly, the refinement of r_1 in case of the *Ord* of [7] specifies that in $\mathsf{G}_1; \mathsf{G}_3$, $!\mathsf{a}$ and the $!\mathsf{x}$ of G_3 are concurrent, but $?\mathsf{a}$ must nevertheless occur before the $?\mathsf{x}$ of G_3, in spite of the fact that the message x of G_3 from A possibly arrives to B before the message a from C.

Example 4. Consider the choreography $\mathsf{G} = \mathsf{G}_1 | \mathsf{G}_2$ in Fig. 2(a). The only action pomset in $[\![\mathsf{G}]\!]$ for the *Ord* of [7], presented in Fig. 2(b), has four ways to execute the action sequence $!\mathsf{x}!\mathsf{x}?\mathsf{x}$, i.e. the event sequence $e_{!\mathsf{x},1} e_{!\mathsf{x},2} e_{?\mathsf{x},1}$: As first, $e_{!\mathsf{x},1}$ can be either the left or the right instance of $!\mathsf{x}$ in Fig. 2(b) ($e_{!\mathsf{x},2}$ is in both cases the remaining instance). As second, $e_{?\mathsf{x},1}$ can be either the left or the right instance of $?\mathsf{x}$ in Fig. 2(b). In the (left,left) and in the (right,right) case, $e_{!\mathsf{x},1}$ causally precedes $e_{?\mathsf{x},1}$, and $e_{!\mathsf{x},2}$ does not. In the (left,right) and in the (right,left) case, $e_{!\mathsf{x},2}$ causally precedes $e_{?\mathsf{x},1}$, and $e_{!\mathsf{x},1}$ does not. The two partial orders on the event set $\{e_{!\mathsf{x},1}, e_{!\mathsf{x},2}, e_{?\mathsf{x},1}\}$ are the two candidate causal interpretations which $[\![G]\!]$ specifies for the sequence $!\mathsf{x}!\mathsf{x}?\mathsf{x}$. As for the cumulative causal interpretation of the latter, it specifies that $e_{!\mathsf{x},1}$ and $e_{!\mathsf{x},2}$ both causally precede $e_{?\mathsf{x},1}$.

Example 5. Consider the choreography $\mathsf{G} = (\mathsf{G}_1 + \mathsf{G}_2) + \mathsf{G}_3$ in Fig. 1(c) and assume the *Ord* of [7]. For the action sequence $?\mathsf{x}!\mathsf{y}$ (i.e. the event sequence $e_{?\mathsf{x},1} e_{!\mathsf{y},1}$) of B, the only member of $[\![\mathsf{G}_1]\!]\lfloor_{\mathsf{B}}$ specifies that $e_{?\mathsf{x},1}$ causally precedes $e_{!\mathsf{y},1}$, and the only member of $[\![\mathsf{G}_3]\!]\lfloor_{\mathsf{B}}$ specifies that $e_{?\mathsf{x},1}$ and $e_{!\mathsf{y},1}$ are concurrent. As for the only pomset in $[\![\mathsf{G}_2]\!]\lfloor_{\mathsf{B}}$, the longest prefix of $?\mathsf{x}!\mathsf{y}$ complying to it is the empty one. For this reason, the pomset specifies only how the events in the empty prefix of $e_{?\mathsf{x},1} e_{!\mathsf{y},1}$ are ordered, which is a causal interpretation already included in what the first or the second one tells about the sequence $?\mathsf{x}!\mathsf{y}$. As for the cumulative causal interpretation of the latter, it specifies that $e_{?\mathsf{x},1}$ causally precedes $e_{!\mathsf{y},1}$.

3.3. Reception-completeness

Example 6. Let us return to Example 3. In the CSM system of G, presented in Fig. 1(f′), it is clearly visible that every $\mathrm{sm}_c([\![\mathsf{G}]\!])$ chooses between its left branch, i.e. (a sub-machine isomorphic to) $\mathrm{sm}_c([\![\mathsf{G}_1; \mathsf{G}_3]\!])$, and its right branch, i.e. (a sub-machine isomorphic to) $\mathrm{sm}_c([\![\mathsf{G}_2; \mathsf{G}_3]\!])$. Because of the unnecessary reception delayings in the only members of $[\![\mathsf{G}_1; \mathsf{G}_3]\!]$ and $[\![\mathsf{G}_2; \mathsf{G}_3]\!]$, $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}_1; \mathsf{G}_3]\!])$ is without the action sequence $?\mathsf{x}?\mathsf{x}?\mathsf{a}$ and $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}_2; \mathsf{G}_3]\!])$ is without the action sequence $?\mathsf{x}?\mathsf{b}$. Consequently, $?\mathsf{x}$ fails to be specified in the initial state of $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}_2; \mathsf{G}_3]\!])$, in spite of the fact that it would be executable in this position within the CSM system of $\mathsf{G}_2; \mathsf{G}_3$. Consequently, $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}]\!])$ fails to comprise the information that an instance of x received in its initial state might as well have been sent by $sm_A([\![G_2;G_3]\!])$, and not only by $sm_A([\![G_1;G_3]\!])$. Consequently, the initial ?x in $sm_B([\![G]\!])$ unjustly cancels the execution of $sm_B([\![G_2;G_3]\!])$, which is the reason why the CSM system of G possibly deadlocks, in spite of the fact that $G_1;G_3$ and $G_2;G_3$ are both realizable. Similarly, ?x fails to be specified in the state of $sm_B([\![G_1;G_3]\!])$ resulting from ?x, in spite of the fact that it would be executable in this position within the CSM system of $G_1;G_3$. This, however, is not problematic, because a second instance of x can only be sent by $sm_A([\![G_1;G_3]\!])$.

The problem described in Example 6 originates in the fact that for the *Ord* of [7], the choreography $[\![G_2;G_3]\!]$, although realizable, is not reception-complete, where the latter choreography property is defined as follows:

Definition 5 (Reception-completeness). A given choreography is reception-complete if its CSM system cannot reach any state in which the message queue of a certain channel is non-empty, but the sink component of the channel is currently unable to receive the first message in the queue.

Example 3 and its continuation in Example 6 indicate that reception-incompleteness of a given realizable choreography G has at least two undesirable implications: First, in the implementation of G obtained by its projection, components' behaviour is constrained more than necessary. Second, such a G is a problematic operand of the choice operator. The latter is very inconvenient for the conception of choreography well-formedness, because abstractly, every choreography is choice between a certain set of interaction pomsets. On top of this, even an individual interaction pomset can be unrealizable for the sole reason of being a reception-incomplete choreography:

Example 7. The choreography G in Fig. 1(d) specifies a single interaction pomset, consisting of an $A \xrightarrow{x} B$, a $C \xrightarrow{y} B$, a $C \xrightarrow{b} Z$ and an $A \xrightarrow{w} B$, with $A \xrightarrow{x} B$ preceding $C \xrightarrow{z} B$, and $C \xrightarrow{y} B$ preceding $A \xrightarrow{w} B$. In case of the *Ord* of [7], $[\![G]\!]$ specifies that !x, !y, !z and !w are concurrent, whereas ?x precedes ?z, and ?y precedes ?w. The resulting CSM system of G, presented in Fig. 1(d′), has the run !z!w!x!y, leading to a state in which the message w is in the channel AB in front of x, and the message z is in the channel CB in front of y, but the only actions currently enabled by B are ?x and ?y, meaning that the messages waiting for reception will never be received, which makes G for the particular *Ord* not only reception-incomplete, but also unrealizable.

3.4. A different version of the predicate Ord

If a given realizable choreography G fails to be reception-complete for the assumed version of *Ord*, this originates in the properties of the particular *Ord*. As reception-incompleteness is undesirable, our version of *Ord* is such that for any action a and reception $c'c?m$, $Ord(a, c'c?m)$ is true only if a is also a reception on the channel $c'c$ (in all other aspects, the new default *Ord* is the same as that of [7]):

Definition 6 (Predicate *Ord*). For given actions a and a', $Ord(a,a')$ denotes that (a,a') is a $(cc'!m, cc''!m')$, a $(c'c?m, cc''!m')$ or a $(c'c?m, c'c?m')$.

The only consequence of our modification of the default *Ord* is that the members of the CSM system of a given choreography G newly comprise complete information on the local states in which individual kinds of messages are possibly available for reception in case that components consistently choose between the pomsets in $[\![G]\!]$. This increases the chances that messages are removed from channels in time and properly interpreted. It also simplifies safety assessment of candidate implementations of individual CSMs, and possibly helps CSM implementers to choose from a wider range of safe reduced implementations. In other words, while the new *Ord* possibly means more action instance concurrency in $[\![G]\!]$ and, hence, larger action sequence sets of individual CSMs, which possibly increases the need for implementing just their reduced versions, it on the other hand brings more implementation freedom:

Example 8. Let us return to Example 7. For the new *Ord*, ?x, ?y, ?z and ?w are concurrent in $[\![G]\!]$. Consequently, $sm_B([\![G]\!])$ is ready to execute them in any order, and G is both realizable and reception-complete.

Example 9. Let us return to Example 6. For the new *Ord*, the CSM system of G is the one presented in Fig. 1(f″). As in the only member of $[\![G_2;G_3]\!]$, ?x is no longer preceded by ?b, both $sm_B([\![G_1;G_3]\!])$ and $sm_B([\![G_2;G_3]\!])$ can start with ?x. Consequently, the initial ?x in $sm_B([\![G]\!])$ is no longer decisive.

Example 10. Let us return to Example 9. When implementing the new $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}]\!])$, it is not safe to omit the branch ?b following ?x, because the CSM comprises no guarantee that in case that the rest of the system decides to support the omitted branch, it will support also at least one of the other two actions possible in the CSM after ?x. On the other hand, it is safe to omit the initial ?x, because in case that the rest of the system decides to support the omitted branch, it thereby supports one of the CSM's action sequences ?x?a, ?x?x?a and ?x?b, thereby supporting (because in each of the sequences, the last reception is not on the same channel as any other) also initial execution of ?a or ?b. In $\mathrm{sm}_{\mathsf{A}}([\![\mathsf{G}]\!])$, the situation is less complicated, because in case that one or two of the initial actions are omitted, the remaining alternative is a transmission and as such non-blockable.

Formally, the reasoning employed in Example 10 is as follows:

Proposition 1. *If a given CSM system with every CSM deterministic and possessing no i-action transitions is a correct implementation of a given choreography, it remains its correct implementation also if one of the CSMs, a state machine M, is modified into a deterministic state machine M' for which in case of* $\mathrm{seq}(M) \neq \mathrm{seq}(M')$*, there is a pair* (α, a) *satisfying all the following:*

(1) $(a \in \mathcal{L}) \wedge (\alpha a \in \mathrm{seq}(M))$
(2) $\mathrm{seq}(M') = \{\alpha' | (\alpha' \in \mathrm{seq}(M)) \wedge (\alpha a \notin \mathrm{pf}(\alpha'))\}$
(3) *If there is no transmission a' with $\alpha a' \in \mathrm{seq}(M')$, there is an action sequence set $A \subseteq (\mathrm{seq}(M) \setminus \mathrm{seq}(M'))$ satisfying all the following:*
    (3.1) $\forall \alpha a \alpha' \in (\mathrm{seq}(M) \setminus \mathrm{seq}(M')) : \exists \alpha a \alpha'' \in A : ((\alpha' \in \mathrm{pf}(\alpha'')) \vee (\alpha'' \in \mathrm{pf}(\alpha')))$
    (3.2) *For every sequence $\alpha a \alpha' \in A$, there is a pair (α'', a') satisfying all the following:*
        (3.2.1) $(\alpha'' \in (\mathcal{L}^?)^*) \wedge (a' \in \mathcal{L}^?) \wedge (\alpha'' a' = a\alpha') \wedge (\alpha a' \in \mathrm{seq}(M'))$
        (3.2.2) *No member of α'' is a reception on the same channel as a'.*

Proof. Suppose that the premise is true. If $\mathrm{seq}(M) = \mathrm{seq}(M')$, then $\delta_{M'}(\alpha)$ of individual $\alpha \in \mathrm{seq}(M')$ is irrelevant for the correctness of the implementation. If $\mathrm{seq}(M) \neq \mathrm{seq}(M')$ and a pair (α, a) with the described properties exists, all the following is true:

(1) The only possible problem is that the M' executes α and then deadlocks.
(2) If there is a transmission a' with $\alpha a' \in \mathrm{seq}(M')$, M' is not blocked after α.
(3) Otherwise, the deadlock occurs because a after α is not an option for M', whereas all the alternative options are receptions never enabled by the rest of the system.
(4) In the latter case, the correctness of the original implementation implies that the deadlock occurs because the rest of the system decides that M' should execute an action sequence starting with an $\alpha a \alpha' \in A$.
(5) Consider the pair (α'', a') that presumably exists for such an $\alpha a \alpha'$. As the original implementation is correct, the rest of the system sends all the messages necessary for the execution of $\alpha'' a'$.
(6) As no reception in α'' is on the same channel as a', M' can, hence, execute a' also immediately after α, thereby avoiding the deadlock. □

3.5. Auto-concurrency

The following example shows that an individual interaction pomset can be unrealizable even for the new (in the rest of the paper the default) *Ord*:

Example 11. Consider the choreography $\mathsf{G} = \mathsf{G}_1 | \mathsf{G}_2$ in Fig. 2(a). The only action pomset in $[\![\mathsf{G}]\!]$, presented in Fig. 2(b), specifies that it is illegal to execute the four transmissions in the order !x!z!y!x. However, the CSM system of G, presented in Fig. 2(c), has also a run starting with !x!z?x!y!x, a run in which B misinterprets the message x of G_2 as the message x of G_1 and consequently sends y prematurely.

The above considered choreography G is unrealizable because of $Ac([\![\mathsf{G}]\!])$ with Ac the following predicate denoting the presence of auto-concurrency:

Definition 7 (Predicate Ac). For a given action pomset set $\mathcal{R}$, $Ac(\mathcal{R})$ denotes that for some poset $(\mathcal{E}, \leq) \in \mathrm{pos}(\mathcal{R})$ and event pair $(e, e') \in \mathcal{E} \times \mathcal{E}$ with $\lambda(e) = \lambda(e')$, neither $e \leq e'$ nor $e' \leq e$ is true.

Lemma 1. *If a given choreography G satisfies $(|\langle\langle G\rangle\rangle| = 1) \wedge \neg Ac([\![G]\!])$, it is realizable and reception-complete.*

Proof. Suppose that the premise is true. The only member of $\text{pos}(\langle\langle G\rangle\rangle)$ is a interaction instance poset $(\mathcal{G}, \leq')$. The only member of $\text{pos}([\![G]\!])$ is by definition an action instance poset $(\mathcal{E}, \leq)$ isomorphic to the poset $p = (\bigcup_{g\in\mathcal{G}}\{e^!_g, e^?_g\}, R^\star)$ with $R = (\bigcup_{g\in\mathcal{G}}\{(e^!_g, e^!_g), (e^!_g, e^?_g), (e^?_g, e^?_g)\}) \cup \{(e, e') | \exists (g, g') \in \leq' : ((g \neq g') \wedge ((e, e') \in \{e^!_g, e^?_g\} \times \{e^!_{g'}, e^?_{g'}\}) \wedge Ord(\lambda(e), \lambda(e')))\}$. Without loss of generality, we assume that $(\mathcal{E}, \leq) = p$. The CSM system of G satisfies all the following:

(1) For every component c, the action sequences α executable by $\text{sm}_c([\![G]\!])$ are exactly those satisfying $\text{pf}(p|_{\downarrow c}, \alpha) \neq \emptyset$.

(2) For every component c and action sequence α executable by $\text{sm}_c([\![G]\!])$, by $\neg Ac([\![G]\!])$, $\text{pf}(p|_{\downarrow c}, \alpha)$ comprises a single prefix of $p|_{\downarrow c}$, and the state of $\text{sm}_c([\![G]\!])$ after α is virtually denoted exactly by the prefix.

(3) By (2) and (3), every component c implements exactly the events in $\mathcal{E}|_{\downarrow c}$, and is ready to execute them exactly in the orders compatible with $\leq$.

(4) $\forall (e, e') \in R : ((\exists g \in \mathcal{G} : ((e, e') = (e^!_g, e^?_g))) \vee (\exists c \in C : ((\lambda(e), \lambda(e')) \in \mathcal{L}_c \times \mathcal{L}_c)))$
I.e., every ordering in $\leq$ which does not result from the remaining ones is either between a transmission instance and the corresponding reception instance, in which case it is implemented by the particular channel, or between two events implemented by the same component, in which case it is implemented by the component.

(5) If no deadlock ever occurs and no component ever misinterprets a received message (i.e. interprets its reception as a certain $e^?_g$ in spite of the fact that it is actually $e^?_{g'}$ of a $g' \in (\mathcal{G} \setminus \{g\})$), then, by (3) and (4), the system in every run executes exactly the events in $\mathcal{E}$, in an order compatible with $\leq$.

(6) $\forall g \in \mathcal{G}, (e, e^?_g) \in R : ((e = e^!_g) \vee$

$$\exists g' \in \mathcal{G}, c, c', m, m' : ((\lambda(g) = c \xrightarrow{m} c') \wedge (\lambda(g') = c \xrightarrow{m'} c') \wedge (e = e^?_{g'}) \wedge ((e^!_{g'}, e^!_g) \in \leq)))$$

I.e., for every reception instance $e^?_g \in \mathcal{E}$, every event $e \in \mathcal{E}$ that is one of its immediate preconditions is either the corresponding transmission instance $e^!_g$ or a reception instance whose corresponding transmission instance occurs on the same channel and is a precondition for $e^!_g$. In both cases, the enabling of $e^?_g$ by the channel is delayed until after e.

(7) By $\neg Ac([\![G]\!])$ and (6), no component ever misinterprets a received message.

(8) By (5)-(7), the system never reaches a state in which a component is on some of its currently non-empty incoming channels unable to receive the first message in the queue.

(9) By (5)-(8), the system in every run executes exactly the events in $\mathcal{E}$, in an order compatible with $\leq$. □

3.6. Local choice

Example 12. Let us return to Example 9. For the new *Ord*, $\mathsf{G}_1; \mathsf{G}_3$ and $\mathsf{G}_2; \mathsf{G}_3$ are realizable and reception-complete choreographies. In the new CSM system of G, components in every run consistently choose whether the system should execute (an action sequence complying to) $[\![\mathsf{G}_1; \mathsf{G}_3]\!]$ or (one complying to) $[\![\mathsf{G}_2; \mathsf{G}_3]\!]$. More precisely, all the following is true: Whenever a component c stops supporting one of the alternative action pomset sets, this is at a point at which both alternatives have further actions specified for c. The cancellation of support happens because c executes an action that is at the particular point legal only for the selected alternative. There is at most one component (in the particular case A) for which the decisive action can only be a transmission. For all the other components, it can only be a reception. In every system run, the choice between the two alternatives is, hence, made (if ever) locally, by the component that can only choose upon a transmission, and gradually communicated to (in the particular case all) the other components (for example, if $[\![\mathsf{G}_2; \mathsf{G}_3]\!]$ is selected instead of $[\![\mathsf{G}_1; \mathsf{G}_3]\!]$, C is informed of this upon ?z, and B upon ?b).

Speaking formally, the choice considered in Example 12 is local because the particular pair of action pomset sets satisfies the following predicate:

Definition 8 (Predicate *Lc*). For given non-empty action pomset sets $\mathcal{R}_1$ and $\mathcal{R}_2$, $Lc(\mathcal{R}_1, \mathcal{R}_2)$ denotes that there exists such a component set C' with $|C'| \leq 1$ that for every component c, action sequence $\alpha \in (\text{asq}_c(\mathcal{R}_1) \cap \text{asq}_c(\mathcal{R}_2))$, $i \in \{1, 2\}$ and action a with $\alpha a \in (\text{asq}_c(R_i) \setminus \text{asq}_c(\mathcal{R}_{3-i}))$, all the following is true:

(1) $\exists a' \in \mathcal{L} : (\alpha a' \in \text{asq}_c(\mathcal{R}_{3-i}))$

(2) $(c \in C') \Leftrightarrow (a \in \mathcal{L}^!)$

Lemma 2. *If given realizable and reception-complete choreographies G_1 and G_2 satisfy $Lc([\![G_1]\!], [\![G_2]\!])$, the choreography $G = G_1 + G_2$ is realizable and reception-complete.*

Proof. Suppose that the premise is true. For given $i \in \{1, 2\}$ and component c, let $A_{i,c}$ denote $\mathrm{asq}_c([\![G_i]\!])$. By $Lc([\![G_1]\!], [\![G_2]\!])$, there is a component set C' for which all the following is true:

(1) $|C'| \leq 1$
(2) $\forall c \in C, \alpha \in (A_{1,c} \cap A_{2,c}), i \in \{1, 2\}, a \in \mathcal{L} : ((\alpha a \in (A_{i,c} \setminus A_{3-i,c})) \Rightarrow \exists a' \in \mathcal{L} : (\alpha a' \in A_{3-i,c}))$
(3) $\forall c \in C, \alpha \in (A_{1,c} \cap A_{2,c}), i \in \{1, 2\}, a \in \mathcal{L} : ((\alpha a \in (A_{i,c} \setminus A_{3-i,c})) \Rightarrow ((c \in C') \Leftrightarrow (a \in \mathcal{L}^!)))$

The CSM system of G satisfies all the following:

(4) For every $c \in C$, $\mathrm{sm}_c([\![G]\!])$ chooses between (a sub-machine isomorphic to) $\mathrm{sm}_c([\![G_1]\!])$ and (a sub-machine isomorphic to) $\mathrm{sm}_c([\![G_2]\!])$.
(5) The moment when an $\mathrm{sm}_c([\![G_i]\!])$ is selected is when after executing an $\alpha \in (A_{1,c} \cap A_{2,c})$, c executes an action a with $\alpha a \in (A_{i,c} \setminus A_{3-i,c})$.
(6) If a is a $c'c?m$, then the message received results from a transmission instance impossible in $\mathrm{sm}_{c'}([\![G_{3-i}]\!])$, for otherwise, by the reception-completeness of G_{3-i}, $\mathrm{sm}_c([\![G_{3-i}]\!])$ would also be ready for $c'c?m$ immediately after α, which would contradict $\alpha a \notin A_{3-i,c}$.
(7) If a is a $c'c?m$, then, by (6), the choice at c is made after c' has selected $\mathrm{sm}_{c'}([\![G_i]\!])$.
(8) By (3)-(7), only a member of C' can be the first component c to make the choice between $\mathrm{sm}_c([\![G_1]\!])$ and $\mathrm{sm}_c([\![G_2]\!])$, and thereby the choice between (a CSM system component-wise isomorphic to) the CSM system of G_1 and (a CSM system component-wise isomorphic to) the CSM system of G_2.
(9) By (1) and (3)-(8), the choice between the two alternative CSM systems is made, if ever, by the only member of C', and every other component follows it consistently.
(10) By (9), every run of the CSM system of G is virtually a run of one of the alternative CSM systems.
(11) By (2), (10) and the realizability and reception-completeness of G_1 and G_2, G is also realizable and reception-complete. □

At this point, it is interesting to note three things: As first, $Lc([\![G_1]\!], [\![G_2]\!])$ in Lemma 2 is a sufficient constraint for individually acceptable G_1 and G_2 in the role of operands of the choice operator. As second, in comparison to the corresponding constraint in [7] (partially presented in Section 4.2), ours is much simpler. The latter is possible because we newly expect operands to be reception-complete, which is crucial for the step (6) in the proof of Lemma 2. More precisely, the reception-completeness of G_1 and G_2 helps because it removes the need for considering also non-initial actions in the futures of individual decision points of individual members of the CSM system of $G_1 + G_2$ (see the Example 13 below). As third, our constraint drops the usual assumption, adopted also in [7], that in individual local decision points, *all* initial actions of the possible futures must be decisive (see the Example 14 below).

Example 13. Let us return to Example 12, more precisely to the old CSM system of G. To correctly answer the question whether it is acceptable that B possibly selects its left alternative already upon an initial ?x, one has to consider also the non-initial ?x in the right alternative, for only then one can answer the question whether the latter instance of ?x is possibly enabled by the rest of the system already before ?b. As we know, the answer to the latter question is positive, which in the new CSM of B is evident simply from the initial ?x in the right alternative.

Example 14. Consider the choreography $\mathsf{G} = \mathsf{G}_1 + \mathsf{G}_2$ with G_1 and G_2 the realizable and reception-complete choreographies $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B} | \mathsf{A} \xrightarrow{\mathsf{y}} \mathsf{B}$ and $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B} | \mathsf{A} \xrightarrow{\mathsf{z}} \mathsf{B}$, respectively. The choice between G_1 and G_2 is made by A, possibly already upon its first action, but only if the action is !y or !z. If the action is !x, the choice is delayed until the next decision point. Nevertheless, $Lc([\![\mathsf{G}_1]\!], [\![\mathsf{G}_2]\!])$, whereas for [7], G is unacceptable.

3.7. Well-formedness

Well-formedness of choreographies is in [7] defined in terms of their topmost operator and the semantics of its operands, whereas we define it in terms of the semantics of choreographies themselves, with the help of the following recursively defined predicate denoting well-branchedness of a given action pomset set:

Definition 9 (Predicate *Wb*). For a given action pomset set $\mathcal{R}$, $Wb(\mathcal{R})$ denotes that either $|\mathcal{R}| = 1$ or there exist non-empty pomset sets $\mathcal{R}_1 \subset \mathcal{R}$ and $\mathcal{R}_2 \subset \mathcal{R}$ satisfying $(\mathcal{R}_1 \cup \mathcal{R}_2 = \mathcal{R}) \wedge Wb(\mathcal{R}_1) \wedge Wb(\mathcal{R}_2) \wedge Lc(\mathcal{R}_1, \mathcal{R}_2)$.

Definition 10 (Well-formedness). A given choreography G is well-formed, denoted as $Wf(G)$, if $\neg Ac([\![G]\!]) \wedge Wb([\![G]\!])$.

Proposition 2. *If a given choreography G satisfies $Wf(G)$, it is realizable and reception-complete.*

Proof. The proof is by induction, assuming $Wf(G)$ and that every choreography G' with $Wf(G') \wedge (|\langle\!\langle G' \rangle\!\rangle| < |\langle\!\langle G \rangle\!\rangle|)$ is realizable and reception-complete:

(1) By $Wf(G)$, there is a non-empty interaction pomset set (i.e. a choreography) $\mathcal{R} \subseteq \langle\!\langle G \rangle\!\rangle$ with $([\![\mathcal{R}]\!] = [\![G]\!]) \wedge (|\mathcal{R}| = |[\![\mathcal{R}]\!]|) \wedge Wf(\mathcal{R})$.
(2) By $[\![\mathcal{R}]\!] = [\![G]\!]$, it suffices to prove that $\mathcal{R}$ is realizable and reception-complete.
(3) If $|\mathcal{R}| = 1$, then $|\langle\!\langle \mathcal{R} \rangle\!\rangle| = 1$ and, by $Wf(\mathcal{R})$, $\neg Ac([\![\mathcal{R}]\!])$ and, by Lemma 1, $\mathcal{R}$ is realizable and reception-complete.
(4) If $|\mathcal{R}| > 1$, then, by $(|\mathcal{R}| = |[\![\mathcal{R}]\!]|) \wedge Wf(\mathcal{R})$, there exist non-empty interaction pomset sets (i.e. choreographies) $\mathcal{R}_1$ and $\mathcal{R}_2$ satisfying $(\mathcal{R}_1 \subset \mathcal{R}) \wedge (\mathcal{R}_2 \subset \mathcal{R}) \wedge (\mathcal{R}_1 \cup \mathcal{R}_2 = \mathcal{R}) \wedge Wf(\mathcal{R}_1) \wedge Wf(\mathcal{R}_2) \wedge Lc([\![\mathcal{R}_1]\!], [\![\mathcal{R}_2]\!])$.
(5) By $(|\langle\!\langle \mathcal{R}_1 \rangle\!\rangle| < |\langle\!\langle G \rangle\!\rangle|) \wedge (|\langle\!\langle \mathcal{R}_2 \rangle\!\rangle| < |\langle\!\langle G \rangle\!\rangle|) \wedge Wf(\mathcal{R}_1) \wedge Wf(\mathcal{R}_2)$, $\mathcal{R}_1$ and $\mathcal{R}_2$ are realizable and reception-complete.
(6) By (5), $Lc([\![\mathcal{R}_1]\!], [\![\mathcal{R}_2]\!])$ and Lemma 2, the choreography $G' = \mathcal{R}_1 + \mathcal{R}_2$ is realizable and reception-complete.
(7) By $[\![G']\!] = [\![G]\!]$ and (6), $\mathcal{R}$ is realizable and reception-complete. □

It seems that when Tuosto and Guanciale defined well-formedness in [7], their plan, though not optimally executed, was the same as ours, namely to enforce 'no auto-concurrency and only local choice'. For the assumed kind of channels, avoiding specification of auto-concurrency is a reasonable decision, for note the following: Even if two instances of a given interaction are specified as concurrent in a given choreography G, so that the corresponding transmission instances are concurrent in $[\![G]\!]$, the corresponding message instances are ordered by the channel on which they are sent. It is just that their order is decided not earlier than at run-time, but this can be more faithfully specified as the possibility of two alternative orderings of the two interaction instances. Analogously, it can be helpful to explicitly specify that the recipient of the two message instances has two different possibilities for attributing them to the two interaction instances:

Example 15. The choreographies G_1 and G_2 in the Figs. 3(a) and 3(b), respectively, for every component c satisfy $\mathrm{seq}(\mathrm{sm}_c([\![\mathsf{G}_1]\!])) = \mathrm{seq}(\mathrm{sm}_c([\![\mathsf{G}_2]\!]))$, but have different semantics. Only $[\![\mathsf{G}_2]\!]$ faithfully represents the fact that in the system, the two instances of y are ordered by the channel, and that A freely chooses whether the first received instance of y should be attributed to the left or to the right instance of $\mathsf{B} \xrightarrow{\mathsf{y}} \mathsf{A}$. Consequently, G_2 is realizable, whereas its abstraction G_1 is not. In the latter, the two instances of $\mathsf{B} \xrightarrow{\mathsf{y}} \mathsf{A}$ are presented as concurrent and each guard exactly one of the interactions $\mathsf{A} \xrightarrow{\mathsf{a}} \mathsf{B}$ and $\mathsf{A} \xrightarrow{\mathsf{b}} \mathsf{B}$. According to $[\![\mathsf{G}_1]\!]$ it is therefore illegal that the CSM system of G_1 possibly executes the action sequence $\mathsf{!x?z?z!y?y!a}$, in which the message of the right instance of $\mathsf{B} \xrightarrow{\mathsf{y}} \mathsf{A}$ is interpreted by A as the message of the left instance of $\mathsf{B} \xrightarrow{\mathsf{y}} \mathsf{A}$.

3.8. Causal-consistent reversibility

In this section we assume that the CSM which a given choreography G defines for a given component c is $\mathrm{rev}(\mathrm{sm}_c([\![G]\!]))$. Note that for a given well-formed choreography G, a given action sequence $\alpha \in \mathrm{asq}([\![G]\!])$ is executable by the CSM system of G exactly if it is in $\mathrm{asq}_{\mathcal{F}}([\![G]\!])$. In the following we prove that a given well-formed choreography G is causal-consistent reversible exactly if there is no action sequence $\alpha \in \mathrm{asq}_{\mathcal{F}}([\![G]\!])$ for which the cumulative causal interpretation $\mathrm{cci}_{[\![G]\!]}(\alpha)$ would specify some intra-channel concurrency of transmissions or receptions, i.e. exactly if $\neg Ic([\![G]\!])$ with Ic the following predicate:

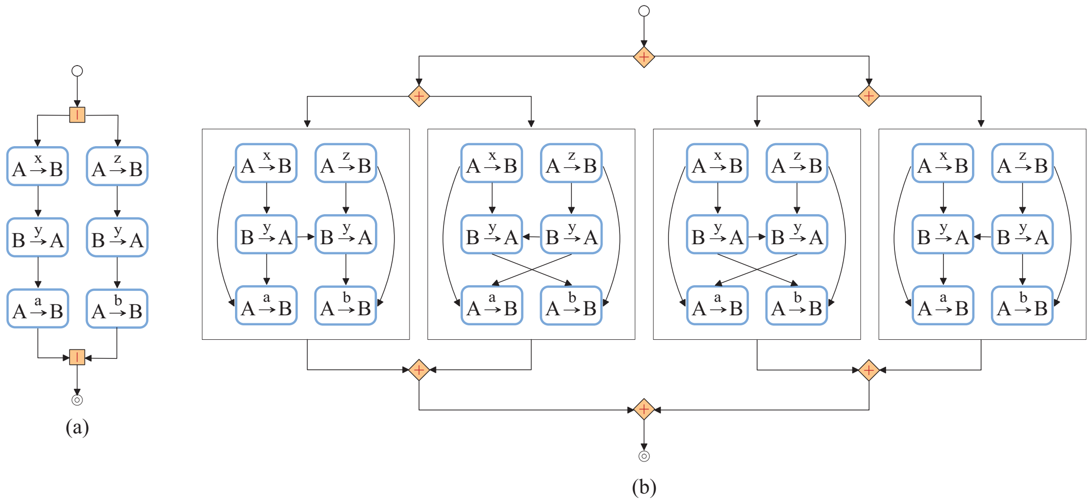

Figure 3: Two choreographies.

Definition 11 (Predicate Ic). For a given action pomset set $\mathcal{R}$, $Ic(\mathcal{R})$ denotes that there exist an action sequence $\alpha \in \mathrm{asq}_{\mathcal{F}}(\mathcal{R})$, a channel cc' and different events $e_{a,i}$ and $e_{a',i'}$ in $\max_{\mathcal{R}}(\alpha)$ with $(a, a') \in ((\mathcal{L}^{!}_{cc'} \times \mathcal{L}^{!}_{cc'}) \cup (\mathcal{L}^{?}_{cc'} \times \mathcal{L}^{?}_{cc'}))$.

For given well-formed choreography G and (i-)action sequence β executable by its CSM system, let $\mathrm{st}_G(\beta)$ denote the system state after β.

Lemma 3. *For any given choreography G satisfying $Wf(G) \wedge \neg Ic(\llbracket G \rrbracket)$, action sequence $\alpha \in \mathrm{asq}_{\mathcal{F}}(\llbracket G \rrbracket)$ and action instance $e_{a,i} \in \max_{\llbracket G \rrbracket}(\alpha)$, $\mathrm{st}_G(\alpha)$ is a state in which the CSM system of G can execute a^{-1}.*

Proof. The action a is a $cc'!m$ or a $c'c?m$. Let M denote the CSM $\mathrm{sm}_c(\llbracket G \rrbracket)$.

(1) By $e_{a,i} \in \max_{\llbracket G \rrbracket}(\alpha)$: $e_{a,i} \in \max_{\llbracket G \rrbracket|_c}(\alpha|_c)$
(2) By (1), there is an action sequence $\alpha' a \in \mathrm{seq}(M)$ with $\delta_M(\alpha' a) = \delta_M(\alpha|_c)$.
(3) By (2): $(\delta_M(\alpha' a), a^{-1}, \delta_M(\alpha')) \in \mathcal{T}_{\mathrm{rev}(M)}$
(4) By (3), $\alpha|_c a^{-1} \in \mathrm{seq}(\mathrm{rev}(M))$ and, hence, in $\mathrm{st}_G(\alpha)$ the component c is ready for a^{-1}.
(5) If a is a $c'c?m$, the channel $c'c$ is always ready for a^{-1}.
(6) If a is a $cc'!m$, then in $\mathrm{st}_G(\alpha)$, by $(e_{a,i} \in \max_{\llbracket G \rrbracket}(\alpha)) \wedge \neg Ic(\llbracket G \rrbracket)$, the last element of the message queue of the channel cc' is an instance of m, implying that the channel is ready for a^{-1}. □

Next we prove that in case of $Wf(G) \wedge \neg Ic(\llbracket G \rrbracket)$, any executed action inverse removes from the action history the last instance of the action and thereby transforms the action sequence into one that is also executable by the system. Moreover, we prove that the undone action instance is one that is allowed to be undone at the point, and that the undoing transforms the system state into the one resulting from the resulting action sequence.

Lemma 4. *For any given choreography G satisfying $Wf(G) \wedge \neg Ic(\llbracket G \rrbracket)$, action sequence $\alpha \in \mathrm{asq}_{\mathcal{F}}(\llbracket G \rrbracket)$ and action a for which the CSM system of G is able to execute the (i-)action sequence αa^{-1}, all the following is true:*

(a) $\max_{\llbracket G \rrbracket}(\alpha)|_a = \{e_{a,|\alpha|_{\{a\}}|}\}$
(b) $\mathrm{rlst}(\alpha, a) \in \mathrm{asq}_{\mathcal{F}}(\llbracket G \rrbracket)$
(c) $\mathrm{st}_G(\alpha a^{-1}) = \mathrm{st}_G(\mathrm{rlst}(\alpha, a))$

Proof. The action a is a $cc'!m$ or a $c'c?m$. Let M denote the CSM $\mathrm{sm}_c(\llbracket G \rrbracket)$.

(1) As αa^{-1} is executable by the system, there is an action sequence $\alpha' a \in \mathrm{seq}(M)$ with $\delta_M(\alpha' a) = \delta_M(\alpha|_c)$.
(2) As the last element of the action sequence $\alpha' a$ is a, $\max_{\llbracket G \rrbracket|_c}(\alpha' a)$ comprises at least $e_{a,|(\alpha' a)|_{\{a\}}|}$.

(3) If a is a $c'c?m$, then, by $\neg Ic(\llbracket G \rrbracket)$, $e_{a,|(\alpha' a)\lfloor_{\{a\}}|}$ is for every action sequence $\alpha'' \in \mathrm{seq}(M)$ with $\delta_M(\alpha'') = \delta_M(\alpha' a)$ the only event in $\max_{\llbracket G \rrbracket\lfloor_c}(\alpha'')$ that is an instance of a reception on the channel $c'c$.

(4) If a is a $cc'!m$, then, by $\neg Ic(\llbracket G \rrbracket)$, $e_{a,|(\alpha' a)\lfloor_{\{a\}}|}$ is for every action sequence $\alpha'' \in \mathrm{seq}(M)$ with $\delta_M(\alpha'') = \delta_M(\alpha' a)$ the only event in $\max_{\llbracket G \rrbracket\lfloor_c}(\alpha'')$ that is an instance of a transmission on the channel $c'c$.

(5) By (3), (4) and $\delta_M(\alpha' a) = \delta_M(\alpha\lfloor_c)$, $\mathrm{rev}(M)$ has exactly one state s with $(\delta_M(\alpha\lfloor_c), a^{-1}, s) \in \mathcal{T}_{\mathrm{rev}(M)}$, namely the state $(\mathrm{ci}_{\mathrm{po}(r)}(\alpha\lfloor_c) \setminus e_{a,|(\alpha' a)\lfloor_{\{a\}}|})_{r \in \llbracket G \rrbracket\lfloor_c}$.

(6) If a is a $c'c?m$, $e_{a,|(\alpha' a)\lfloor_{\{a\}}|}$ is, by (3), for every action sequence $\alpha'' \in \mathrm{asq}_{\mathcal{F}}(\llbracket G \rrbracket)$ with $\delta_M(\alpha''\lfloor_c) = \delta_M(\alpha' a)$ the only event in $\max_{\llbracket G \rrbracket}(\alpha'')$ that is an instance of a reception on the channel $c'c$.

(7) If a is a $cc'!m$, $e_{a,|(\alpha' a)\lfloor_{\{a\}}|}$ is, by (4), for every action sequence $\alpha'' \in \mathrm{asq}_{\mathcal{F}}(\llbracket G \rrbracket)$ with $\delta_M(\alpha''\lfloor_c) = \delta_M(\alpha' a)$ and $\alpha'' a^{-1}$ executable by the system the only event in $\max_{\llbracket G \rrbracket}(\alpha'')$ that is an instance of a transmission on the channel cc'.

(8) If a is a $c'c?m$, then, by $\delta_M(\alpha' a) = \delta_M(\alpha\lfloor_c)$ and (6), $e_{a,|(\alpha' a)\lfloor_{\{a\}}|}$ is the only instance of a in $\max_{\llbracket G \rrbracket}(\alpha)$ and in α denotes the last reception on the channel $c'c$.

(9) If a is a $cc'!m$, then, by $\delta_M(\alpha' a) = \delta_M(\alpha\lfloor_c)$ and (7), $e_{a,|(\alpha' a)\lfloor_{\{a\}}|}$ is the only instance of a in $\max_{\llbracket G \rrbracket}(\alpha)$ and in α denotes the last transmission on the channel cc'.

(10) By (8) and (9): $e_{a,|(\alpha' a)\lfloor_{\{a\}}|} = e_{a,|\alpha\lfloor_{\{a\}}|}$

(11) By (8)-(10): $(\max_{\llbracket G \rrbracket}(\alpha)\lfloor_a = \{e_{a,|\alpha\lfloor_{\{a\}}|}\}) \wedge (\mathrm{rlst}(\alpha, a) \in \mathrm{asq}_{\mathcal{F}}(\llbracket G \rrbracket))$

(12) By (5) and (8)-(11), $\mathrm{st}_G(\alpha a^{-1}) = \mathrm{st}_G(\mathrm{rlst}(\alpha, a))$. □

From the Lemmas 3 and 4 it is already evident that in case of $Wf(G) \wedge \neg Ic(\llbracket G \rrbracket)$, the CSM system of G allows event undoing exactly where appropriate, and in the only form appropriate. For completeness, however, we next prove causal-consistent reversibility of the default implementation of G also formally. Here note that our definition of causal-consistent reversibility of the implementation in no way differs from the classical one, which is (see, for example, the overview paper [15]) that any action instance can be undone, provided that all its consequences (if any, with respect to the employed causal interpretation function, in our case $\mathrm{cci}_{\llbracket G \rrbracket}$) are undone beforehand. To prove the property, it suffices to prove (as we do below) that the implementation is a state machine possessing the following properties [15]:

(1) For every transition (s, a, s') with $a \in \mathcal{L}$, there is also the inverse transition (s', a^{-1}, s), and for every transition (s, a^{-1}, s') with $a^{-1} \in \mathcal{L}^{-1}$, there is also the inverse transition (s', a, s).

(2) Any two transition sequences with a common starting state have a common ending state exactly if they are causally equivalent, i.e. if they have the same effect on the causal history of the system.

Proposition 3. *If a given choreography G satisfies $Wf(G) \wedge \neg Ic(\llbracket G \rrbracket)$, it is causal-consistent reversible.*

Proof. Suppose that the premise is true. Hence, all the following is true:

(1) The CSM system of G is a state machine M.

(2) By Lemma 4, M is deterministic.

(3) By $Wf(G)$ and Lemma 4: $\mathrm{seq}(M) \cap \mathcal{L}^* = \mathrm{asq}_{\mathcal{F}}(\llbracket G \rrbracket)$

(4) By Lemma 4: $\mathcal{S}_M = \{\mathrm{st}_G(\alpha) | \alpha \in \mathrm{asq}_{\mathcal{F}}(\llbracket G \rrbracket)\}$

(5) For any action sequence α and action a with $\alpha a^{-1} \in \mathrm{seq}(M)$, there is, by Lemma 4, an $e_{\alpha,i} \in \max_{\llbracket G \rrbracket}(\alpha)$.

(6) For any action sequences $\alpha \in \mathrm{seq}(M)$ and $e_{a,i} \in \max_{\llbracket G \rrbracket}(\alpha)$, there is an action sequence $\alpha' a \in \mathrm{seq}(M)$ with $\delta_M(\alpha) = \delta_M(\alpha' a)$.

(7) For any action sequences α and action a with $\alpha a \in \mathrm{seq}(M)$, $e_{a,|\alpha a\lfloor_{\{a\}}|} \in \max_{\llbracket G \rrbracket}(\alpha a)$.

(8) For any action sequence α and action a with $\alpha a \in \mathrm{seq}(M)$, the transition $(\delta_M(\alpha), a, \delta_M(\alpha a)) \in \mathcal{T}_M$ is, by (7) and the Lemmas 3 and 4, in $\mathcal{T}_M$ accompanied by the transition $(\delta_M(\alpha a), a^{-1}, \delta_M(\alpha))$.

(9) Besides the inverse transitions, M has no other i-action transition, for otherwise, by (5)-(8), (a) of Lemma 4 is contradicted.

(10) Let M' denote the state machine
$(\{\mathrm{st}_G(\alpha) | \alpha \in \mathrm{asq}_{\mathcal{F}}(\llbracket G \rrbracket)\}, \mathrm{st}_G(\epsilon), \{(\mathrm{st}_G(\alpha), a, \mathrm{st}_G(\alpha a)) | (a \in \mathcal{L}) \wedge (\alpha a \in \mathrm{asq}_{\mathcal{F}}(\llbracket G \rrbracket))\})$.

(11) By (1)-(9): $M = \mathrm{rev}(M')$

(12) For any two action sequences α and α' in $\mathrm{asq}_{\mathcal{F}}([\![G]\!])$, by $\neg Ic([\![G]\!])$:
$(\mathrm{st}_G(\alpha) = \mathrm{st}_G(\alpha')) \Leftrightarrow ((\mathrm{civ}_{[\![G]\!]\downarrow_c}(\alpha\lfloor_c))_{c \in C} = (\mathrm{civ}_{[\![G]\!]\downarrow_c}(\alpha'\lfloor_c))_{c \in C})$

(13) For any two action sequences α and α' in $\mathrm{asq}_{\mathcal{F}}([\![G]\!])$, because components, by $Wf(G)$, consistently choose between pomsets in $[\![G]\!]$: $((\mathrm{civ}_{[\![G]\!]\downarrow_c}(\alpha\lfloor_c))_{c \in C} = (\mathrm{civ}_{[\![G]\!]\downarrow_c}(\alpha'\lfloor_c))_{c \in C}) \Leftrightarrow (\mathrm{civ}_{[\![G]\!]}(\alpha) = \mathrm{civ}_{[\![G]\!]}(\alpha'))$

(14) By $\neg Ac([\![G]\!])$: $(\mathrm{civ}_{[\![G]\!]}(\alpha) = \mathrm{civ}_{[\![G]\!]}(\alpha')) \Leftrightarrow (\mathrm{cci}_{[\![G]\!]}(\alpha) = \mathrm{cci}_{[\![G]\!]}(\alpha'))$

(15) Let M'' denote the state machine
$(\{\mathrm{cci}_{[\![G]\!]}(\alpha) | \alpha \in \mathrm{asq}_{\mathcal{F}}([\![G]\!])\}, \mathrm{cci}_{[\![G]\!]}(\epsilon), \{(\mathrm{cci}_{[\![G]\!]}(\alpha), a, \mathrm{cci}_{[\![G]\!]}(\alpha a)) | (a \in \mathcal{L}) \wedge (\alpha a \in \mathrm{asq}_{\mathcal{F}}([\![G]\!]))\})$.

(16) By (10)-(15), M is isomorphic to $\mathrm{rev}(M'')$.

(17) M'' is an implementation of G in which the current state is denoted exactly by the current causal history.

(18) By (16) and (17), M is a state machine in which given transition sequences starting in the same state end in the same state exactly if they have the same effect on the causal history, i.e., if they are causally equivalent.

(19) By (16)-(18) and [15], the CSM system of G is its causal-consistent reversible implementation. □

Proposition 4. *If a given choreography G satisfies $Wf(G) \wedge Ic([\![G]\!])$, it is not causal-consistent reversible.*

Proof. Suppose that the premise is true. Hence, all the following is true:

(1) For any action sequence $\alpha \in \mathrm{asq}_{\mathcal{F}}([\![G]\!])$, channel cc' and different events $e_{a,i}$ and $e_{a',i'}$ in $\max_{[\![G]\!]}(\alpha)$ satisfying $(a, a') \in \mathcal{L}^?_{cc'} \times \mathcal{L}^?_{cc'}$, those elements of α that are the transmission instance corresponding to $e_{a,i}$ and $e_{a',i'}$ are concurrent from the aspect of every poset $(\mathcal{E}, \leq) \in \mathrm{ci}_{[\![G]\!]}(\alpha)$ with $\{e_{a,i}, e_{a',i'}\} \subseteq \mathcal{E}$.

(2) By $Ic([\![G]\!])$ and (1), there exist such action sequence $\alpha \in \mathrm{asq}_{\mathcal{F}}([\![G]\!])$, channel cc' and different events $e_{a,i}$ and $e_{a',i'}$ in $\max_{[\![G]\!]}(\alpha)$ that $(a, a') \in \mathcal{L}^!_{cc'} \times \mathcal{L}^!_{cc'}$.

(3) For any action sequence $\alpha \in \mathrm{asq}_{\mathcal{F}}([\![G]\!])$ and different events $e_{cc'!m,i}$ and $e_{cc'!m',i'}$ in $\max_{[\![G]\!]}(\alpha)$, the system state after α should allow both $cc'!m^{-1}$ and $cc'!m'^{-1}$, which is possible only if the last element of the message queue of the channel cc' is an instance of both m and m', but as, by $\neg Ac([\![G]\!])$, $m \neq m'$, this is impossible. □

4. Inference of choreography well-formedness

Having proved that all well-formed choreographies G are realizable, reception-complete and in case of $\neg Ic([\![G]\!])$ causal-consistent reversible, we in this section prove a set of inference rules that can be useful in proving that a given choreography is well-formed. In Section 4.1, we point out some direct consequences of our observations in previous sections, among them well-formedness of elementary choreographies. In the Sections 4.2-4.4, we study well-formedness inference for choice, parallel composition and sequential composition, respectively. For each of the composition operators, we prove that in case that its operands are well-formed and satisfy certain additional constraints, the composition is also well-formed. The constraints suggested for operands of individual composition operators are compared with those suggested in [7], both conceptually and on examples. Thereby we reveal in which points the old constraints are inadequate.

4.1. Four simple inference rules

Proposition 5. *If a choreography G satisfies $(|[\![G]\!]| = 1) \wedge \neg Ac([\![G]\!])$, then $Wf(G)$.*

Proof. By $|[\![G]\!]| = 1$, $Wb([\![G]\!])$. Hence, by $\neg Ac([\![G]\!])$, $Wf(G)$. □

Proposition 6. *If a choreography G is of the form $\mathbf{0}$ or $c \xrightarrow{m} c'$, then $Wf(G)$.*

Proof. Every such G satisfies the premise of Proposition 5. □

Proposition 7. *If given choreographies G and G' satisfy $([\![G]\!] = [\![G']\!]) \wedge Wf(G)$, then $Wf(G')$.*

Proof. For each of the choreographies, well-formedness is defined exclusively in terms of its semantics. □

Example 16. Consider the choreography $\mathsf{G} = (\mathsf{G}_1 + \mathsf{G}_2); \mathsf{G}_3$ in Fig. 1(b) and the choreography $\mathsf{G}_4 = (\mathsf{G}_1; \mathsf{G}_3) + (\mathsf{G}_2; \mathsf{G}_3)$. A compositional proof of well-formedness exists only for G_4, because in G, the sub-choreography $\mathsf{G}_1 + \mathsf{G}_2$ is not well-formed. By $([\![\mathsf{G}]\!] = [\![\mathsf{G}_4]\!]) \wedge Wf(\mathsf{G}_4)$, however, G is also well-formed.

Proposition 8. *If for a given choreography G, there exist non-empty pomsets $\mathcal{R}_1$ and $\mathcal{R}_2$ with $(\mathcal{R}_1 \cup \mathcal{R}_2 = \langle\!\langle G \rangle\!\rangle) \wedge Wf(\mathcal{R}_1 + \mathcal{R}_2)$, then $Wf(G)$.*

Proof. By $[\![G]\!] = [\![\mathcal{R}_1 + \mathcal{R}_2]\!]$ and Proposition 7. □

4.2. Well-formedness inference for choice

Our constraint for given well-formed choreographies G_1 and G_2 in the role of operands of the choice operator (shortly choice constraint) is simply $Lc([\![G_1]\!], [\![G_2]\!])$.

Proposition 9. *If given choreographies G_1 and G_2 satisfy $Wf(G_1) \wedge Wf(G_2) \wedge Lc([\![G_1]\!], [\![G_2]\!])$, then $Wf(G_1 + G_2)$.*

Proof. Suppose that the premise is true. Hence, all the following is true:

(1) For every $i \in \{1, 2\}$, by $Wf(G_i)$: $\neg Ac([\![G_i]\!]) \wedge Wb([\![G_i]\!])$
(2) By $\neg Ac([\![G_1]\!]) \wedge \neg Ac([\![G_2]\!]) : \neg Ac([\![G_1 + G_2]\!])$
(3) By $Wb([\![G_1]\!]) \wedge Wb([\![G_2]\!]) \wedge Lc([\![G_1]\!], [\![G_2]\!]) : Wb([\![G_1 + G_2]\!])$
(4) By (2) and (3): $Wf(G_1 + G_2)$ □

From Definition 8 it is evident that our choice constraint $Lc([\![G_1]\!], [\![G_2]\!])$ is defined in terms of action sequences executable by individual components, in no way considering the candidate causal interpretations of the sequences. As such, the constraint comprises no restrictions of causal ambiguity, whereas the choice constraint of [7] has virtually been conceived with the intent to rule out the possibility that in $G_1 + G_2$, there is causal ambiguity other than that in G_1 or G_2 individually.

Example 17. Consider the choreography $\mathsf{G} = (\mathsf{G}_1 + \mathsf{G}_2) + \mathsf{G}_3$ in Fig. 1(c). For each of the choreographies G_1, G_2 and G_3, the old semantics is the same as the new one, and all are well-formed both in the old and in the new sense. The choice $\mathsf{G}_1 + \mathsf{G}_2$ is non-local, whereas the choices $\mathsf{G}_1 + \mathsf{G}_3$ and $\mathsf{G}_2 + \mathsf{G}_3$ introduce causal ambiguity (recall Example 5). Consequently, none of the choices satisfies the old version of the choice constraint. With the new one, however, it is possible to compositionally prove $Wf(\mathsf{G})$, as follows:

By $Wf(\mathsf{G}_2) \wedge Wf(\mathsf{G}_3) \wedge Lc([\![\mathsf{G}_2]\!], [\![\mathsf{G}_3]\!])$, the choreography $\mathsf{G}_4 = \mathsf{G}_2 + \mathsf{G}_3$ is well-formed (the component responsible for the choice between G_2 and G_3 is A). By $Wf(\mathsf{G}_1) \wedge Wf(\mathsf{G}_4) \wedge Lc([\![\mathsf{G}_1]\!], [\![\mathsf{G}_4]\!])$, the choreography $\mathsf{G}_5 = \mathsf{G}_1 + \mathsf{G}_4$ is well-formed (the component responsible for the choice between G_1 and G_4 is B). By $([\![\mathsf{G}]\!] = [\![\mathsf{G}_5]\!]) \wedge Wf(\mathsf{G}_5)$, G is also well-formed.

The old choice constraint has been defined in terms of the candidate causal interpretations of action sequences executable by individual components. If in $G_1 + G_2$, there is a lot of parallelism and not much causal ambiguity, the approach can be very convenient. For this reason, we also provide a pomset-based choice constraint. Like [7], we intend it for operands G_i without auto-concurrency whose semantics satisfies the following predicate implying their local causal unambiguity:

Definition 12 (Predicate Lu). For a given action pomset set $\mathcal{R}$, $Lu(\mathcal{R})$ denotes that for every component c and action sequence $\alpha \in \mathrm{asq}(\mathcal{R}|_c)$, $|\,\mathrm{pf}(\mathcal{R}|_c, \alpha)| = 1$.

Example 18. Let us return to Example 5. Among the prefixes of the action pomsets in $[\![G]\!]|_{\mathsf{B}}$, there are also a pomset specifying concurrent execution of ?x and !y and a pomset specifying their sequential execution. The action sequence ?x!y $\in \mathrm{asq}([\![G]\!]|_{\mathsf{B}})$ is a legal permutation of both pomsets. Hence, $\neg Lu([\![\mathsf{G}]\!])$.

In the constraint $Lc([\![G_1]\!], [\![G_2]\!])$, every CSM $\mathrm{sm}_c([\![G_i]\!])$ is regarded as an executor of action sequences, i.e. as a machine that in each step takes the sequence α of its past actions and, by executing an action a, enhances it into the action sequence $\alpha' = \alpha a$. Formally, such a step is the triplet (α, a, α'). In case of $Lu([\![G_1]\!]) \wedge Lu([\![G_2]\!])$, the triplet can alternatively be regarded as the triplet (r, a, r') with r and r' the action pomsets that are the unique maximal candidate causal interpretations of α and α', respectively. In the alternative view, the step set of individual $\mathrm{sm}_c([\![G_i]\!])$ is, hence, the triplet set $\mathrm{tri}([\![G_i]\!]|_c)$, whereas the corresponding rephrasing of (the relevant specialization of) the predicate Lc is the following predicate Lc', with $Lc'([\![G_1]\!], [\![G_2]\!])$ our pomset-based choice constraint for locally causally unambiguous operands:

Definition 13 (Predicate Lc'). For given non-empty action pomset sets $\mathcal{R}_1$ and $\mathcal{R}_2$, $Lc'(\mathcal{R}_1, \mathcal{R}_2)$ denotes that $Lu(\mathcal{R}_1) \wedge Lu(\mathcal{R}_2)$ and that there exists such a component set C' with $|C'| \leq 1$ that for every component c, pomset $r \in (\mathrm{pf}(\mathcal{R}_1|_c) \cap \mathrm{pf}(\mathcal{R}_2|_c))$, $i \in \{1, 2\}$ and $(r, a, r') \in \mathrm{tri}(\mathcal{R}_i|_c)$ with $r' \notin \mathrm{pf}(\mathcal{R}_{3-i}|_c)$, all the following is true:

(1) $\nexists(r,a,r_1)\in \mathrm{tri}(\mathcal{R}_{3-i}\lfloor_c)$
(2) $\exists(r,a',r_2)\in \mathrm{tri}(\mathcal{R}_{3-i}\lfloor_c)$
(3) $(c\in C')\Leftrightarrow(a\in\mathcal{L}^!)$

Lemma 5. *If given non-empty action pomset sets $\mathcal{R}_1$ and $\mathcal{R}_2$ satisfy $Lc'(\mathcal{R}_1,\mathcal{R}_2)$, then $Lc(\mathcal{R}_1,\mathcal{R}_2)\wedge Lu(\mathcal{R}_1\cup\mathcal{R}_2)$.*

Proof. Suppose that the C' required in Definition 13 actually exists. For given $i\in\{1,2\}$ and component c, let $\mathcal{R}_{i,c}$ denote $\mathrm{pf}(\mathcal{R}_i\lfloor_c)$. For every component c, action sequence $\alpha\in(\mathrm{asq}(\mathcal{R}_{1,c})\cap\mathrm{asq}(\mathcal{R}_{2,c}))$ with $\exists r:(\mathrm{pf}(\mathcal{R}_{1,c},\alpha)=\mathrm{pf}(\mathcal{R}_{2,c},\alpha)=\{r\})$, $i\in\{1,2\}$ and action a with $\alpha a\in\mathrm{asq}(\mathcal{R}_{i,c})$, all the following is true:

(1) $\forall i\in\{1,2\},c\in C:(\mathrm{asq}_c(\mathcal{R}_i)=\mathrm{asq}(\mathcal{R}_{i,c}))$
(2) By $Lu(\mathcal{R}_i)$, $\mathrm{pf}(\mathcal{R}_{i,c},\alpha a)$ is an $\{r'\}$ with $(r,a,r')\in\mathrm{tri}(\mathcal{R}_{i,c})$.
(3) If $(\alpha a\in\mathrm{asq}(\mathcal{R}_{3-i,c}))\wedge(r'\in\mathcal{R}_{3-i,c})$ then, by $(r'\in\mathrm{pf}(\{r'\},\alpha a))\wedge Lu(\mathcal{R}_{3-i})$, $\mathrm{pf}(\mathcal{R}_{3-i,c},\alpha a)=\{r'\}$.
(4) If $(\alpha a\in\mathrm{asq}(\mathcal{R}_{3-i,c}))\wedge(r'\notin\mathcal{R}_{3-i,c})$ then, by $Lu(\mathcal{R}_{3-i})$, there is an $(r,a,r'')\in\mathrm{tri}(\mathcal{R}_{3-i,c})$, contradicting $Lc'(\mathcal{R}_1,\mathcal{R}_2)$.
(5) If $\alpha a\notin\mathrm{asq}(\mathcal{R}_{3-i,c})$ then $r'\notin\mathcal{R}_{3-i,c}$ and, by $Lc'(\mathcal{R}_1,\mathcal{R}_2)$, there is an action a' with $\alpha a'\in\mathrm{asq}(\mathcal{R}_{3-i,c})$ and $(c\in C')\Leftrightarrow(a\in\mathcal{L}^!)$.

From the above, satisfaction of the constraints which $Lc(\mathcal{R}_1,\mathcal{R}_2)$ and $Lu(\mathcal{R}_1\cup\mathcal{R}_2)$ impose for individual components c follows by induction on increasingly longer $\alpha\in(\mathrm{asq}(\mathcal{R}_{1,c})\cap\mathrm{asq}(\mathcal{R}_{2,c}))$. □

Proposition 10. *If given choreographies G_1 and G_2 satisfy $Wf(G_1)\wedge Wf(G_2)\wedge Lc'([\![G_1]\!],[\![G_2]\!])$, then $Wf(G_1+G_2)\wedge Lu([\![G_1+G_2]\!])$.*

Proof. Suppose that the premise is true.

(1) By Lemma 5: $Lc([\![G_1]\!],[\![G_2]\!])\wedge Lu([\![G_1]\!]\cup[\![G_2]\!])$
(2) By $Wf(G_1)\wedge Wf(G_2)\wedge Lc([\![G_1]\!],[\![G_2]\!])$ and Proposition 9: $Wf(G_1+G_2)$
(3) By $Lu([\![G_1]\!]\cup[\![G_2]\!])\wedge([\![G_1+G_2]\!]=[\![G_1]\!]\cup[\![G_2]\!])$: $Lu([\![G_1+G_2]\!])$ □

The papers [7, 8] suggest that the plan for the old choice constraint was to define it as an $Lc_{\mathsf{old}}([\![G_1]\!],[\![G_2]\!])$ with Lc_{old} a certain specialization of the following specialization Lc'' of the predicate Lc':

Definition 14 (Predicate Lc''). For given non-empty action pomset sets $\mathcal{R}_1$ and $\mathcal{R}_2$, $Lc''(\mathcal{R}_1,\mathcal{R}_2)$ denotes that $Lu(\mathcal{R}_1)\wedge Lu(\mathcal{R}_2)$ and that there exist such component set C' with $|C'|\le 1$ and pomset array $[q_{i,c,r}]_{i\in\{1,2\},c\in C,r\in\mathcal{R}_i\lfloor_c}$ satisfying $\forall i\in\{1,2\},c\in C,r\in\mathcal{R}_i\lfloor_c:((q_{i,c,r}\in\mathrm{pf}(r))\wedge\exists r'\in\mathcal{R}_{3-i}\lfloor_c:(q_{i,c,r}=q_{3-i,c,r'}))$ that for every component c, pomset $r\in(\mathrm{pf}(\mathcal{R}_1\lfloor_c)\cap\mathrm{pf}(\mathcal{R}_2\lfloor_c))$, $i\in\{1,2\}$ and $(r,a,r')\in\mathrm{tri}(\mathcal{R}_i\lfloor_c)$ with $r'\notin\mathrm{pf}(\mathcal{R}_{3-i}\lfloor_c)$, all the following is true:

(0) $\exists r_0\in\mathcal{R}_i\lfloor_c:(r\in\mathrm{pf}(q_{i,c,r_0}))$
(1) $\nexists(r,a,r_1)\in\mathrm{tri}(\mathcal{R}_{3-i}\lfloor_c)$
(2) $\exists(r,a',r_2)\in\mathrm{tri}(\mathcal{R}_{3-i}\lfloor_c)$
(3) $(c\in C')\Leftrightarrow(a\in\mathcal{L}^!)$
(4) $(a\in\mathcal{L}^?)\Rightarrow(\nexists(r_3,a,r_4)\in\mathrm{tri}(\mathcal{R}_{3-i}\lfloor_c):(r\in\mathrm{pf}(r_3)))$

The predicate Lc'' specializes the predicate Lc' by the additional constraints (0) and (4). In the more restrictive choice constraint $Lc''([\![G_1]\!],[\![G_2]\!])$, the additional (0) serves no particular purpose, whereas the additional (4) removes the need for the reception-completeness of G_1 and G_2. With the new versions of the choreography semantics and well-formedness, however, the latter is irrelevant, because $Wf(G_1)\wedge Wf(G_2)$ is sufficient for the reception-completeness of G_1 and G_2. There are cases where removing the redundant (4) can actually help:

Example 19. Consider the choreography $\mathsf{G}=\mathsf{G}_1+\mathsf{G}_2$ in Fig. 1(a). For each of the choreographies G_1 and G_2, the old semantics is the same as the new one, and the choreographies are well-formed both in the old and in the new sense. Hence, G_1 and G_2 are reception-complete, but $Lc''([\![\mathsf{G}_1]\!],[\![\mathsf{G}_2]\!])$ is nevertheless not satisfied. On the other hand, if one removes the redundant (4) in Definition 14, $Lc''([\![\mathsf{G}_1]\!],[\![\mathsf{G}_2]\!])$ can be satisfied, by setting C' to $\{\mathsf{A}\}$ and every $q_{i,c,r}$ to the empty pomset.

Unfortunately, there is a detail in which the old choice constraint fails to be a specialization of $Lc''([\![G_1]\!],[\![G_2]\!])$. In other words, the actual Lc_{old} is not an implementation of Lc''. Namely, instead of implementing the constraint (2) in Definition 14, Lc_{old} virtually implements the less restrictive constraint $\exists r_5 \in \mathcal{R}_{3-i}\lfloor_c, r_6 \in \mathrm{pf}(q_{3-i,c,r_5}), (r_6, a', r_7) \in \mathrm{tri}(\mathcal{R}_{3-i}\lfloor_c)$, which is a problem:

Example 20. Consider the choreography $\mathsf{G} = \mathsf{G}_1 + \mathsf{G}_2$ in Fig. 1(e). For each of the choreographies $\mathsf{G}_1 = \mathsf{G}_{1,1} + \mathsf{G}_{1,2}$ and $\mathsf{G}_2 = \mathsf{G}_{2,1} + \mathsf{G}_{2,2}$, the old semantics is the same as the new one, and the choreographies are well-formed both in the old and in the new sense. G as a whole is unrealizable, for example because in $\mathrm{sm}_{\mathsf{A}}(\mathsf{G})$, !a executed in both (the sub-machine isomorphic to) $\mathrm{sm}_{\mathsf{A}}(\mathsf{G}_{1,1})$ and (the sub-machine isomorphic to) $\mathrm{sm}_{\mathsf{A}}(\mathsf{G}_{2,1})$ is not necessarily followed by arrival of the expected c, for B possibly executes !y in $\mathrm{sm}_{\mathsf{B}}(\mathsf{G}_{2,1})$, an event cancelling $\mathrm{sm}_{\mathsf{B}}(\mathsf{G}_{1,1})$ and thereby !c.

Because of the deadlock possibility in G, neither $Lc(\mathcal{R}_1,\mathcal{R}_2)$ for $(\mathcal{R}_1,\mathcal{R}_2) = ([\![\mathsf{G}_1]\!],[\![\mathsf{G}_2]\!])$ nor its specialization $Lc''([\![\mathsf{G}_1]\!],[\![\mathsf{G}_2]\!])$ is satisfied, because, respectively, the constraint (1) in Definition 8 or the corresponding constraint (2) in Definition 14 is not satisfied wherever required. On the other hand, $Lc_{\mathsf{old}}([\![\mathsf{G}_1]\!],[\![\mathsf{G}_2]\!])$ is satisfied, because the subsumed weaker version of the constraint (2) in Definition 14 is satisfied wherever required. Consequently, G is well-formed in the old sense, which is incompatible with its unrealizability.

4.3. Well-formedness inference for parallel composition

Our constraint for given well-formed choreographies G_1 and G_2 in the role of operands of the parallel composition operator is the same as that of [7], prescribing that G_1 and G_2 use different actions, i.e. that $\lambda([\![G_1]\!]) \cap \lambda([\![G_2]\!]) = \emptyset$. This suffices for correct interpretation of received messages, but not for timely reception on shared channels. For the latter, we rely on the reception-completeness of G_1 and G_2, which in case of the old *Ord* is not secured:

Example 21. Consider the choreography $\mathsf{G} = \mathsf{G}_1|\mathsf{G}_2$ in Fig. 1(d). The choreographies G_1 and G_2 are well-formed both in the old and in the new sense, actually realizable for both the old and the new *Ord*, and use different actions. For every component c, $\mathrm{sm}_c([\![\mathsf{G}]\!])$ is (isomorphic to) the parallel composition of (a state machine isomorphic to) $\mathrm{sm}_c([\![\mathsf{G}_1]\!])$ and (a state machine isomorphic to) $\mathrm{sm}_c([\![\mathsf{G}_2]\!])$. Thus, A can execute !x and !w in any order, and C can execute !y and !z in any order, whereas the behaviour of B depends on whether one assumes the old or the new choreography semantics. In case of the old one, $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}_1]\!])$ can execute ?z only after ?x, and $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}_2]\!])$ can execute ?w only after ?y, so that in case that the CSM system of G, presented in Fig.1(d′), executes the transmission sequence !z!w!x!y, the resulting system state is one in which the message w is in the channel AB in front of x, and the message z is in the channel CB in front of y, but the only action currently enabled by $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}_1]\!])$ is ?x, and the only action currently enabled by $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}_2]\!])$ is ?y, meaning that the messages waiting for reception will never be received. In case of the new choreography semantics, $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}_1]\!])$ can receive x and z in any order, and $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}_2]\!])$ can receive y and w in any order, meaning that $\mathrm{sm}_{\mathsf{B}}([\![\mathsf{G}]\!])$ can receive x, y, z and w in any order.

Lemma 6. *If given choreographies G_1, G_2 and G_3 satisfy $Lc([\![G_1]\!],[\![G_2]\!]) \wedge \forall i \in \{1,2\} : (\lambda([\![G_i]\!]) \cap \lambda([\![G_3]\!]) = \emptyset)$, then $Lc([\![G_1|G_3]\!],[\![G_2|G_3]\!])$.*

Proof. Suppose that the premise is true. For given $i \in \{1,2\}$ and component c, let $A_{i,c}$ and $A'_{i,c}$ denote $\mathrm{asq}_c([\![G_i]\!])$ and $\mathrm{asq}_c([\![G_i|G_3]\!])$, respectively. By $Lc([\![G_1]\!],[\![G_2]\!])$, there is a component set C' for which all the following is true:

(1) $|C'| \leq 1$
(2) $\forall c \in C, \alpha \in (A_{1,c} \cap A_{2,c}), i \in \{1,2\}, a \in \mathcal{L} : ((\alpha a \in (A_{i,c} \setminus A_{3-i,c})) \Rightarrow \exists a' \in \mathcal{L} : (\alpha a' \in A_{3-i,c}))$
(3) $\forall c \in C, \alpha \in (A_{1,c} \cap A_{2,c}), i \in \{1,2\}, a \in \mathcal{L} : ((\alpha a \in (A_{i,c} \setminus A_{3-i,c})) \Rightarrow ((c \in C') \Leftrightarrow (a \in \mathcal{L}^!)))$

Hence, all the following is true:

(4) For every $i \in \{1,2\}$ and component c, by $\lambda([\![G_i]\!]) \cap \lambda([\![G_3]\!]) = \emptyset$:
$A'_{i,c} = \{\alpha | (\alpha \in (L_c)^*) \wedge \exists \alpha' \in A_{i,c}, \alpha'' \in A_{3,c} : ((\alpha\lfloor_{\lambda([\![G_i]\!])} = \alpha') \wedge (\alpha\lfloor_{\lambda([\![G_3]\!])} = \alpha'') \wedge (|\alpha'| + |\alpha''| = |\alpha|))\}$
(5) By (2) and (4): $\forall c \in C, \alpha \in (A'_{1,c} \cap A'_{2,c}), i \in \{1,2\}, a \in \mathcal{L} : ((\alpha a \in (A'_{i,c} \setminus A'_{3-i,c})) \Rightarrow \exists a' \in \mathcal{L} : (\alpha a' \in A'_{3-i,c}))$
(6) By (3) and (4): $\forall c \in C, \alpha \in (A'_{1,c} \cap A'_{2,c}), i \in \{1,2\}, a \in \mathcal{L} : ((\alpha a \in (A'_{i,c} \setminus A'_{3-i,c})) \Rightarrow ((c \in C') \Leftrightarrow (a \in \mathcal{L}^!)))$ □

Proposition 11. *If given choreographies G_1 and G_2 satisfy $Wf(G_1) \wedge Wf(G_2) \wedge (\lambda([\![G_1]\!]) \cap \lambda([\![G_2]\!]) = \emptyset)$, then $Wf(G_1|G_2)$.*

Proof. The proof is by induction, assuming that for any choreographies G'_1 and G'_2 with $(|\langle\!\langle G'_1 \rangle\!\rangle| < |\langle\!\langle G_1 \rangle\!\rangle|) \vee (|\langle\!\langle G'_2 \rangle\!\rangle| < |\langle\!\langle G_2 \rangle\!\rangle|)$, $Wf(G'_1) \wedge Wf(G'_2) \wedge (\lambda([\![G'_1]\!]) \cap \lambda([\![G'_2]\!]) = \emptyset)$ is sufficient for $Wf(G'_1|G'_2)$. For a given $i \in \{1,2\}$, let $\mathcal{R}_i$ denote $\langle\!\langle G_i \rangle\!\rangle$. Suppose that the premise is true. Hence, $Wf(\mathcal{R}_1) \wedge Wf(\mathcal{R}_2) \wedge (\lambda([\![\mathcal{R}_1]\!]) \cap \lambda([\![\mathcal{R}_2]\!]) = \emptyset)$. By $[\![\mathcal{R}_1|\mathcal{R}_2]\!] = [\![G_1|G_2]\!]$, it suffices to prove $Wf(\mathcal{R}_1|\mathcal{R}_2)$.

If there is an $i \in \{1,2\}$ with $|\mathcal{R}_i| > 1$, all the following is true:

(1) By $Wf(R_i)$, there exist non-empty pomsets $\mathcal{R}_{i,1} \subset \mathcal{R}_i$ and $\mathcal{R}_{i,2} \subset \mathcal{R}_i$ satisfying $(\mathcal{R}_{i,1} \cup \mathcal{R}_{i,2} = \mathcal{R}_i) \wedge Wf(\mathcal{R}_{i,1}) \wedge Wf(\mathcal{R}_{i,2}) \wedge Lc([\![\mathcal{R}_{i,1}]\!], [\![\mathcal{R}_{i,2}]\!])$.
(2) By $\lambda([\![\mathcal{R}_1]\!]) \cap \lambda([\![\mathcal{R}_2]\!]) = \emptyset$: $(\lambda([\![\mathcal{R}_{i,1}]\!]) \cap \lambda([\![\mathcal{R}_{3-i}]\!]) = \emptyset) \wedge (\lambda([\![\mathcal{R}_{i,2}]\!]) \cap \lambda([\![\mathcal{R}_{3-i}]\!]) = \emptyset)$
(3) For every $j \in \{1,2\}$, by $(|\langle\!\langle \mathcal{R}_{i,j} \rangle\!\rangle| < |\mathcal{R}_i|) \wedge Wf(\mathcal{R}_{i,j}) \wedge Wf(\mathcal{R}_{3-i}) \wedge (\lambda([\![\mathcal{R}_{i,j}]\!]) \cap \lambda([\![\mathcal{R}_{3-i}]\!]) = \emptyset)$: $Wf(\mathcal{R}_{i,j}|\mathcal{R}_{3-i})$
(4) By $Lc([\![\mathcal{R}_{i,1}]\!], [\![\mathcal{R}_{i,2}]\!]) \wedge \forall j \in \{1,2\} : (\lambda([\![\mathcal{R}_{i,j}]\!]) \cap \lambda([\![\mathcal{R}_{3-i}]\!]) = \emptyset)$ and Lemma 6: $Lc([\![\mathcal{R}_{i,1}|\mathcal{R}_{3-i}]\!], [\![\mathcal{R}_{i,2}|\mathcal{R}_{3-i}]\!])$
(5) By (3), (4) and Proposition 9: $Wf((\mathcal{R}_{i,1}|\mathcal{R}_{3-i}) + (\mathcal{R}_{i,2}|\mathcal{R}_{3-i}))$
(6) By (5) and $[\![\mathcal{R}_1|\mathcal{R}_2]\!] = [\![(\mathcal{R}_{i,1}|\mathcal{R}_{3-i}) + (\mathcal{R}_{i,2}|\mathcal{R}_{3-i})]\!]$: $Wf(\mathcal{R}_1|\mathcal{R}_2)$

If $|\mathcal{R}_1| = |\mathcal{R}_2| = 1$, all the following is true:

(1) By $Wf(\mathcal{R}_1) \wedge Wf(\mathcal{R}_2)$: $\neg Ac([\![\mathcal{R}_1]\!]) \wedge \neg Ac([\![\mathcal{R}_2]\!])$
(2) By (1) and $\lambda([\![\mathcal{R}_1]\!]) \cap \lambda([\![\mathcal{R}_2]\!]) = \emptyset$: $\neg Ac([\![\mathcal{R}_1|\mathcal{R}_2]\!])$
(3) By $|[\![\mathcal{R}_1|\mathcal{R}_2]\!]| = 1$: $Wb([\![\mathcal{R}_1|\mathcal{R}_2]\!])$ □

4.4. Well-formedness inference for sequential composition

Our constraint for given well-formed choreographies G_1 and G_2 in the role of operands of the sequential composition operator is $Ls(G_1, G_2)$ with Ls the following predicate:

Definition 15 (Predicate *Ls*). For given choreographies G_1 and G_2, $Ls(G_1, G_2)$ denotes that they are locally strictly sequenced, i.e. that for every interaction instance poset pair $((\mathcal{G}_1, \leq_1), (\mathcal{G}_2, \leq_2)) \in \mathrm{pos}_1(\langle\!\langle G_1 \rangle\!\rangle) \times \mathrm{pos}_2(\langle\!\langle G_2 \rangle\!\rangle)$, the action instance poset $[\![(\mathcal{G}_1 \cup \mathcal{G}_2, (\leq_1 \cup \leq_2 \cup (\mathcal{G}_1 \times \mathcal{G}_2))^\star)]\!]$ is an $(\mathcal{E}, \leq)$ with $\leq \supseteq \bigcup_{c \in C}((\bigcup_{g \in \mathcal{G}_1} \{e^!_g, e^?_g\}\downharpoonright_c) \times (\bigcup_{g \in \mathcal{G}_2} \{e^!_g, e^?_g\}\downharpoonright_c))$.

We see that the constraint $Ls(G_1, G_2)$ is virtually defined in terms of constraints on individual action instance posets in $\mathrm{pos}([\![G_1; G_2]\!])$. For each of them it requires that every event e belonging to G_2 is delayed until its executor c has executed all its events belonging to G_1, where in case that e is an instance of a reception, the assumed choreography semantics must allow that 'delayed' is interpreted as 'not enabled by the channel', for otherwise the constraint is in general not sufficient for the realizability of $G_1; G_2$. Such interpretation is allowed provided that the semantics specifies reception-completeness of choreographies, which is in general true only for the new semantics. In [7], 'delayed' virtually means 'delayed by c' (and is specified already in the semantics of $G_1; G_2$). In this sense, [7] is less restrictive, which is a problem, in spite of the fact that the old constraint for operands of the sequential composition operator additionally requires that reception instances in G_2 are delayed with respect to every transmission instance in the selected alternative of G_1.

Example 22. Consider the choreography $\mathsf{G} = \mathsf{G}_1; \mathsf{G}_2$ in Fig. 1(f). For each of the choreographies $\mathsf{G}_1 = \mathsf{G}_{1,1} + \mathsf{G}_{1,2}$ and G_2, the old semantics is the same as the new one, and the choreographies are well-formed both in the old and in the new sense.

In case of the old semantics, the CSM system of G, presented in Fig.1(f′), possibly executes the action sequence !z?z!b!x?x, in which the components A and C choose $\mathsf{G}_{1,2}$, but the message x of G_2 is transmitted so early that the component B executes its reception as its first event. Consequently, B erroneously interprets the reception as an event in $\mathsf{G}_{1,1}$, concludes that $\mathsf{G}_{1,1}$ is the selected alternative, and becomes permanently unready for the message b of $\mathsf{G}_{1,2}$. Nevertheless, G_1 and G_2 satisfy the constraint of [7] for operands of the sequential composition operator, because in both alternatives $\mathsf{G}_{1,1}; \mathsf{G}_2$ and $\mathsf{G}_{1,2}; \mathsf{G}_2$ of $\mathsf{G}_1; \mathsf{G}_2$, the only reception instance in G_2 is delayed with respect to every transmission instance in $\mathsf{G}_{1,1}$ or $\mathsf{G}_{1,2}$, respectively. Consequently, G is well-formed in the old sense, in spite of the fact that for the old semantics, it is unrealizable.

G is well-formed also in the new sense, but this is fine, because for the new semantics, G is actually realizable. The reason is that the semantics corrects its CSM system to the one presented in Fig 1(f″). To prove $Wf(G)$ compositionally is, however, impossible with our inference rules only, because $Ls(\mathsf{G}_1, \mathsf{G}_2)$ is not satisfied. The reason is that $Ls(\mathsf{G}_{1,1}; \mathsf{G}_2)$ and $Ls(\mathsf{G}_{1,2}; \mathsf{G}_2)$ are not satisfied. Namely, in the new semantics of $\mathsf{G}_{1,1}; \mathsf{G}_2$ and $\mathsf{G}_{1,2}; \mathsf{G}_2$, the instance of AB?x in G_2 is not delayed until B has executed CB?a or CB?b, respectively.

It seems that the requirement of [7] that every reception instance in G_2 must be delayed with respect to every transmission instance in the selected alternative of G_1 has been introduced to secure that messages belonging to G_1 are not received until the recipient knows how many instances of the particular message G_1 will transmit on the particular channel. After the last transmission in a realizable G_1, the number is indeed known, but this does not mean that the recipient knows it. On the other hand, the recipient sometimes knows the number already before the last transmission in G_1, for example if it is the same in every alternative of G_1:

Example 23. The choreography $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{C} \xrightarrow{\mathsf{y}} \mathsf{D}$ is realizable for both the old and the new *Ord* (in both cases, it has the same semantics, namely the same as the choreography $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B} | \mathsf{C} \xrightarrow{\mathsf{y}} \mathsf{D}$), but only the new constraint for operands of the sequential composition operator is satisfied in it.

Lemma 7. *If given choreographies G_1, G_2 and G_3 satisfy $Ls(G_1, G_3) \wedge Ls(G_2, G_3) \wedge Lc([\![G_1]\!], [\![G_2]\!])$, then $Lc([\![G_1; G_3]\!], [\![G_2; G_3]\!])$.*

Proof. Suppose that the premise is true. For given $i \in \{1, 2\}$ and component c, let $A_{i,c}$ and $A'_{i,c}$ denote $\text{asq}_c([\![G_i]\!])$ and $\text{asq}_c([\![G_i; G_3]\!])$, respectively. By $Lc([\![G_1]\!], [\![G_2]\!])$, there is a component set C' for which all the following is true:

(1) $|C'| \leq 1$

(2) $\forall c \in C, \alpha \in (A_{1,c} \cap A_{2,c}), i \in \{1, 2\}, a \in \mathcal{L} : ((\alpha a \in (A_{i,c} \setminus A_{3-i,c})) \Rightarrow \exists a' \in \mathcal{L} : (\alpha a' \in A_{3-i,c}))$

(3) $\forall c \in C, \alpha \in (A_{1,c} \cap A_{2,c}), i \in \{1, 2\}, a \in \mathcal{L} : ((\alpha a \in (A_{i,c} \setminus A_{3-i,c})) \Rightarrow ((c \in C') \Leftrightarrow (a \in \mathcal{L}^!)))$

Hence, all the following is true:

(4) By (2) and $Ls(G_1, G_3) \wedge Ls(G_2, G_3)$:
$\forall c \in C, \alpha \in (A'_{1,c} \cap A'_{2,c}), i \in \{1, 2\}, a \in \mathcal{L} :$
$((\alpha a \in (A'_{i,c} \setminus A'_{3-i,c})) \Rightarrow ((\alpha \in (A_{1,c} \cap A_{2,c})) \wedge (\alpha a \in (A_{i,c} \setminus A_{3-i,c})) \wedge \exists a' \in \mathcal{L} : (\alpha a' \in A'_{3-i,c})))$

(5) By (3) and (4): $\forall c \in C, \alpha \in (A'_{1,c} \cap A'_{2,c}), i \in \{1, 2\}, a \in \mathcal{L} : ((\alpha a \in (A'_{i,c} \setminus A'_{3-i,c})) \Rightarrow ((c \in C') \Leftrightarrow (a \in \mathcal{L}^!)))$ ☐

Lemma 8. *If given choreographies G_1, G_2 and G_3 satisfy $(|\langle\!\langle G_3 \rangle\!\rangle| = 1) \wedge Ls(G_3, G_1) \wedge Ls(G_3, G_2) \wedge Lc([\![G_1]\!], [\![G_2]\!])$, then $Lc([\![G_3; G_1]\!], [\![G_3; G_2]\!])$.*

Proof. Suppose that the premise is true. For given $i \in \{1, 2\}$ and component c, let $A_{i,c}$ and $A'_{i,c}$ denote $\text{asq}_c([\![G_i]\!])$ and $\text{asq}_c([\![G_3; G_i]\!])$, respectively. By $Lc([\![G_1]\!], [\![G_2]\!])$, there is a component set C' for which all the following is true:

(1) $|C'| \leq 1$

(2) $\forall c \in C, \alpha \in (A_{1,c} \cap A_{2,c}), i \in \{1, 2\}, a \in \mathcal{L} : ((\alpha a \in (A_{i,c} \setminus A_{3-i,c})) \Rightarrow \exists a' \in \mathcal{L} : (\alpha a' \in A_{3-i,c}))$

(3) $\forall c \in C, \alpha \in (A_{1,c} \cap A_{2,c}), i \in \{1, 2\}, a \in \mathcal{L} : ((\alpha a \in (A_{i,c} \setminus A_{3-i,c})) \Rightarrow ((c \in C') \Leftrightarrow (a \in \mathcal{L}^!)))$

Hence, all the following is true:

(4) For every $i \in \{1, 2\}$ and component c, by $Ls(G_3, G_i)$:
$A'_{i,c} = \text{asq}_c([\![G_3]\!]) \cup \{\alpha\alpha' | (\alpha \in \text{asq}_c([\![G_3]\!])) \wedge (\alpha' \in A_{i,c}) \wedge \nexists \alpha'' \in \text{asq}_c([\![G_3]\!]) : (|\alpha''| > |\alpha|)\}$

(5) By (2) and (4): $\forall c \in C, \alpha \in (A'_{1,c} \cap A'_{2,c}), i \in \{1, 2\}, a \in \mathcal{L} : ((\alpha a \in (A'_{i,c} \setminus A'_{3-i,c})) \Rightarrow \exists a' \in \mathcal{L} : (\alpha a' \in A'_{3-i,c}))$

(6) By (3) and (4): $\forall c \in C, \alpha \in (A'_{1,c} \cap A'_{2,c}), i \in \{1, 2\}, a \in \mathcal{L} : ((\alpha a \in (A'_{i,c} \setminus A'_{3-i,c})) \Rightarrow ((c \in C') \Leftrightarrow (a \in \mathcal{L}^!)))$ ☐

Proposition 12. *If given choreographies G_1 and G_2 satisfy $Wf(G_1) \wedge Wf(G_2) \wedge Ls(G_1, G_2)$, then $Wf(G_1; G_2)$.*

Proof. The proof is by induction, assuming that for any choreographies G'_1 and G'_2 with $(|\langle\!\langle G'_1 \rangle\!\rangle| < |\langle\!\langle G_1 \rangle\!\rangle|) \vee (|\langle\!\langle G'_2 \rangle\!\rangle| < |\langle\!\langle G_2 \rangle\!\rangle|)$, $Wf(G'_1) \wedge Wf(G'_2) \wedge Ls(G'_1, G'_2)$ is sufficient for $Wf(G'_1; G'_2)$. For a given $i \in \{1, 2\}$, let $\mathcal{R}_i$ denote $\langle\!\langle G_i \rangle\!\rangle$. Suppose that the premise is true. Hence, $Wf(\mathcal{R}_1) \wedge Wf(\mathcal{R}_2) \wedge Ls(\mathcal{R}_1, \mathcal{R}_2)$. By $[\![\mathcal{R}_1; \mathcal{R}_2]\!] = [\![G_1; G_2]\!]$, it suffices to prove $Wf(\mathcal{R}_1; \mathcal{R}_2)$.

If $|\mathcal{R}_1| > 1$, all the following is true:

(1) By $Wf(R_1)$, there exist non-empty pomsets $\mathcal{R}_{1,1} \subset \mathcal{R}_1$ and $\mathcal{R}_{1,2} \subset \mathcal{R}_1$ satisfying $(\mathcal{R}_{1,1} \cup \mathcal{R}_{1,2} = \mathcal{R}_1) \wedge Wf(\mathcal{R}_{1,1}) \wedge Wf(\mathcal{R}_{1,2}) \wedge Lc([\![\mathcal{R}_{1,1}]\!], [\![\mathcal{R}_{1,2}]\!])$.

(2) By $Ls(\mathcal{R}_1, \mathcal{R}_2)$: $Ls(\mathcal{R}_{1,1}, \mathcal{R}_2) \wedge Ls(\mathcal{R}_{1,2}, \mathcal{R}_2)$
(3) For every $i \in \{1, 2\}$, by $(|\langle\!\langle \mathcal{R}_{1,i} \rangle\!\rangle| < |\mathcal{R}_1|) \wedge Wf(\mathcal{R}_{1,i}) \wedge Wf(\mathcal{R}_2) \wedge Ls(\mathcal{R}_{1,i}, \mathcal{R}_2)$: $Wf(\mathcal{R}_{1,i}; \mathcal{R}_2)$
(4) By $Ls(\mathcal{R}_{1,1}, \mathcal{R}_2) \wedge Ls(\mathcal{R}_{1,2}, \mathcal{R}_2) \wedge Lc([\![\mathcal{R}_{1,1}]\!], [\![\mathcal{R}_{1,2}]\!])$ and Lemma 7: $Lc([\![\mathcal{R}_{1,1}; \mathcal{R}_2]\!], [\![\mathcal{R}_{1,2}; \mathcal{R}_2]\!])$
(5) By (3), (4) and Proposition 9: $Wf((\mathcal{R}_{1,1}; \mathcal{R}_2) + (\mathcal{R}_{1,2}; \mathcal{R}_2))$
(6) By (5) and $[\![\mathcal{R}_1; \mathcal{R}_2]\!] = [\![(\mathcal{R}_{1,1}; \mathcal{R}_2) + (\mathcal{R}_{1,2}; \mathcal{R}_2)]\!]$: $Wf(\mathcal{R}_1; \mathcal{R}_2)$

If $|\mathcal{R}_1| = |\mathcal{R}_2| = 1$, all the following is true:

(1) By $Wf(\mathcal{R}_1) \wedge Wf(\mathcal{R}_2)$: $\neg Ac([\![\mathcal{R}_1]\!]) \wedge \neg Ac([\![\mathcal{R}_2]\!])$
(2) By (1) and $Ls(\mathcal{R}_1, \mathcal{R}_2)$: $\neg Ac([\![\mathcal{R}_1; \mathcal{R}_2]\!])$
(3) By $|[\![\mathcal{R}_1; \mathcal{R}_2]\!]| = 1$: $Wb([\![\mathcal{R}_1; \mathcal{R}_2]\!])$

If $(|\mathcal{R}_1| = 1) \wedge (|\mathcal{R}_2| > 1)$, all the following is true:

(1) By $Wf(R_2)$, there exist non-empty pomsets $\mathcal{R}_{2,1} \subset \mathcal{R}_2$ and $\mathcal{R}_{2,2} \subset \mathcal{R}_2$ satisfying $(\mathcal{R}_{2,1} \cup \mathcal{R}_{2,2} = \mathcal{R}_2) \wedge Wf(\mathcal{R}_{2,1}) \wedge Wf(\mathcal{R}_{2,2}) \wedge Lc([\![\mathcal{R}_{2,1}]\!], [\![\mathcal{R}_{2,2}]\!])$.
(2) By $Ls(\mathcal{R}_1, \mathcal{R}_2)$: $Ls(\mathcal{R}_1, \mathcal{R}_{2,1}) \wedge Ls(\mathcal{R}_1, \mathcal{R}_{2,2})$
(3) For every $i \in \{1, 2\}$, by $(|\langle\!\langle \mathcal{R}_{2,i} \rangle\!\rangle| < |\mathcal{R}_2|) \wedge Wf(\mathcal{R}_1) \wedge Wf(\mathcal{R}_{2,i}) \wedge Ls(\mathcal{R}_1, \mathcal{R}_{2,i})$: $Wf(\mathcal{R}_1; \mathcal{R}_{2,i})$
(4) By $(|\langle\!\langle \mathcal{R}_1 \rangle\!\rangle| = 1) \wedge Ls(\mathcal{R}_1, \mathcal{R}_{2,1}) \wedge Ls(\mathcal{R}_1, \mathcal{R}_{2,2}) \wedge Lc([\![\mathcal{R}_{2,1}]\!], [\![\mathcal{R}_{2,2}]\!])$ and Lemma 8: $Lc([\![\mathcal{R}_1; \mathcal{R}_{2,1}]\!], [\![\mathcal{R}_1; \mathcal{R}_{2,2}]\!])$
(5) By (3), (4) and Proposition 9: $Wf((\mathcal{R}_1; \mathcal{R}_{2,1}) + (\mathcal{R}_1; \mathcal{R}_{2,2}))$
(6) By (5) and $[\![\mathcal{R}_1; \mathcal{R}_2]\!] = [\![(\mathcal{R}_1; \mathcal{R}_{2,1}) + (\mathcal{R}_1; \mathcal{R}_{2,2})]\!]$: $Wf(\mathcal{R}_1; \mathcal{R}_2)$ □

5. Concluding remarks

We re-engineered the pomset-based abstract semantics and semantic constraints which Tuosto and Guanciale [7, 8] recently proposed for compositionally specified choreographies for distributed systems with FIFO channels. Our primary aim has been to remove the flaws which originally prevented well-formed choreographies from being realizable in the general case. We achieved this mainly by securing that the default implementation of well-formed choreographies is reception-complete, taking care also that the CSMs obtained by choreography projection have no inexecutable transitions. Thanks to the latter, each of the CSMs comprises more precise information of what the component is to expect from the rest of the system, which brings an additional degree of freedom in the construction of safe reduced versions of the CSMs and in the conception of constraints for choreographies in the role of operands of the choice operator. By less restrictive constraints for operands of the operator, we newly allowed branching points in which only some continuations are decisive, and choreographies exploiting accidental event orderings.

Devising a set of rules for inferring well-formedness of choreographies compositionally, we corrected and in certain ways relaxed also constraints for operands of the sequential composition operator. An item for future work is to conceive even less restrictive reasonably simple sufficient constraints for the operands, in particular such allowing that in the default implementation of the first operand, some system components are termination-unaware. The possibility of (context-compensated) termination-unawareness has already been foreseen in [16], the paper in which Tuosto and Guanciale took their abstract handling of choreographies even further than in [7], considering the realizability of general action pomset sets, though assuming that communication buffers are used not as queues, but as bags of messages, and without considering compositionality aspects, which in [7] and in our paper are the primary concern.

Besides rules for using as building blocks also choreographies which are not well-formed, our framework for compositional conception of well-formed choreographies lacks also operators and inference rules for compositional conception of infinite well-formed choreographies. The latter, however, have virtually already been foreseen in the paper, because for choreographies specified as an interaction pomset set, we do not prescribe that the set or its elements must be finite.

The new version of choreography well-formedness still does not cover all realizable choreographies. In particular, it fails to cover realizable choreographies with auto-concurrency, for example the choreography $\mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B} | \mathsf{A} \xrightarrow{\mathsf{x}} \mathsf{B}$. For the considered kind of channels, however, choreographies comprising intra-channel concurrency are less interesting, because they fail to capture channels' FIFO policy and as such typically fail to have on the channels a causal-consistent reversible default implementation. We say this because, unlike [10], we refrain from assuming that intra-channel concurrency, if specified, is an irrelevant degree of freedom in the choreography semantics. On the other hand, recall our assumption that when undoing past events, the implementation must respect all specified candidate causal interpretations of the past. We chose this restrictive approach because with it, one can have causal-consistent reversibility in the strict sense defined in [15] without sacrificing the specified exploitation of accidental event orderings. In the future, we plan to study also choreography projections implementing more liberal event undoing, which the community of system components could use, for example, to achieve that the resulting global action sequence is one of those specified by the choreography which are not executable on FIFO channels.

Our study has shown that in the conception of choreographies, it indeed pays to go abstract, but not as abstract as to ignore useful or problematic constraints which channels put on the communication actions and their undoings.

Acknowledgement

This work was supported by the Slovenian Research Agency [research programme P2-0095] and partially by the EU COST Action IC1405.

References

[1] I. Lanese, From reversible semantics to reversible debugging, in: Reversible Computation - 10th International Conference, RC 2018, Leicester, UK, September 12-14, 2018, Proceedings, 2018, pp. 34–46. doi:10.1007/978-3-319-99498-7_2.
URL `https://doi.org/10.1007/978-3-319-99498-7_2`

[2] I. Lanese, N. Nishida, A. Palacios, G. Vidal, Cauder: A causal-consistent reversible debugger for erlang, in: Functional and Logic Programming - 14th International Symposium, FLOPS 2018, Nagoya, Japan, May 9-11, 2018, Proceedings, 2018, pp. 247–263. doi:10.1007/978-3-319-90686-7_16.
URL `https://doi.org/10.1007/978-3-319-90686-7_16`

[3] R. Neykova, N. Yoshida, Let it recover: multiparty protocol-induced recovery, in: Proceedings of the 26th International Conference on Compiler Construction, Austin, TX, USA, February 5-6, 2017, 2017, pp. 98–108.
URL `http://dl.acm.org/citation.cfm?id=3033031`

[4] C. D. Carothers, K. S. Perumalla, R. Fujimoto, Efficient optimistic parallel simulations using reverse computation, ACM Trans. Model. Comput. Simul. 9 (3) (1999) 224–253. doi:10.1145/347823.347828.
URL `https://doi.org/10.1145/347823.347828`

[5] M. Schordan, T. Oppelstrup, D. R. Jefferson, P. D. B. Jr., Generation of reversible C++ code for optimistic parallel discrete event simulation, New Generation Comput. 36 (3) (2018) 257–280. doi:10.1007/s00354-018-0038-2.
URL `https://doi.org/10.1007/s00354-018-0038-2`

[6] J. S. Laursen, L. Ellekilde, U. P. Schultz, Modelling reversible execution of robotic assembly, Robotica 36 (5) (2018) 625–654. doi:10.1017/S0263574717000613.
URL `https://doi.org/10.1017/S0263574717000613`

[7] E. Tuosto, R. Guanciale, Semantics of global view of choreographies, Journal of Logical and Algebraic Methods in Programming 95 (2018) 17 – 40. doi:https://doi.org/10.1016/j.jlamp.2017.11.002.
URL `http://www.sciencedirect.com/science/article/pii/S2352220816301754`

[8] E. Tuosto, R. Guanciale, Semantics of global view of choreographies: Proofs, Tech. rep. (2017).
URL `http://www.cs.le.ac.uk/people/et52/jlamp-with-proofs.pdf`

[9] D. Brand, P. Zafiropulo, On communicating finite-state machines, J. ACM 30 (2) (1983) 323–342. doi:10.1145/322374.322380.
URL `http://doi.acm.org/10.1145/322374.322380`

[10] C. A. Mezzina, E. Tuosto, Choreographies for automatic recovery, CoRR abs/1705.09525 (2017).
URL `http://arxiv.org/abs/1705.09525`

[11] M. Bravetti, G. Zavattaro, Towards a unifying theory for choreography conformance and contract compliance, in: Proceedings of the 6th International Conference on Software Composition, SC'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 34–50.
URL `http://dl.acm.org/citation.cfm?id=1785051.1785057`

[12] M. Carbone, K. Honda, N. Yoshida, A calculus of global interaction based on session types, Electr. Notes Theor. Comput. Sci. 171 (3) (2007) 127–151. doi:10.1016/j.entcs.2006.12.041.
URL `https://doi.org/10.1016/j.entcs.2006.12.041`

[13] I. Lanese, C. Guidi, F. Montesi, G. Zavattaro, Bridging the gap between interaction- and process-oriented choreographies, in: Sixth IEEE International Conference on Software Engineering and Formal Methods, SEFM 2008, Cape Town, South Africa, 10-14 November 2008, 2008, pp. 323–332. doi:10.1109/SEFM.2008.11.
URL `https://doi.org/10.1109/SEFM.2008.11`

[14] J. Lange, E. Tuosto, N. Yoshida, From communicating machines to graphical choreographies, in: Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015, 2015, pp. 221–232. doi:10.1145/2676726.2676964.
URL `https://doi.org/10.1145/2676726.2676964`

[15] I. Lanese, C. A. Mezzina, F. Tiezzi, Causal-consistent reversibility, Bulletin of the EATCS 114 (2014).
URL `http://eatcs.org/beatcs/index.php/beatcs/article/view/305`

[16] R. Guanciale, E. Tuosto, Realisability of pomsets, J. Log. Algebr. Meth. Program. 108 (2019) 69–89. doi:10.1016/j.jlamp.2019.06.003. URL https://doi.org/10.1016/j.jlamp.2019.06.003