Formal Choreographic Languages

Franco Barbanera[1], Ivan Lanese[2,3], and Emilio Tuosto[4(✉)]

[1] Department of Mathematics and Computer Science, University of Catania, Catania, Italy
barba@dmi.unict.it
[2] Focus Team, University of Bologna, Bologna, Italy
ivan.lanese@gmail.com
[3] Focus Team, INRIA, Sophia Antipolis, France
[4] Gran Sasso Science Institute, L'Aquila, Italy
emilio.tuosto@gssi.it

Abstract. We introduce a meta-model based on formal languages, dubbed *formal choreographic languages*, to study message-passing systems. Our main motivation is to establish a framework for the comparison and generalisation of standard constructions and properties from the literature. In particular, we consider notions such as global view, local view, and projections from the former to the latter. The correctness of local views projected from global views is characterised in terms of a closure property. A condition is also devised to guarantee relevant communication properties such as (dead)lock-freedom. Formal choreographic languages capture existing formalisms for message-passing systems; we detail the cases of multiparty session types and choreography automata. Unlike many other models, formal choreographic languages can naturally model systems exhibiting non-regular behaviour.

1 Introduction

Choreographic models of message-passing systems are gaining momentum both in academia [8,12,13] and industry [10,27,34]. These models envisage the so-called *global* and *local* views of communicating systems. The former can be thought of as holistic descriptions of protocols that a number of participants should realise through some communication actions, the latter as descriptions of the contribution of single participants.

Research partly supported by the EU H2020 RISE programme under the Marie Skłodowska-Curie grant agreement No 778233. Work partially funded by MIUR project PRIN 2017FTXR7S *IT MATTERS* (Methods and Tools for Trustworthy Smart Systems). The first and second authors have also been partially supported by INdAM as members of GNCS (Gruppo Nazionale per il Calcolo Scientifico). The first author has also been partially supported by Progetto di Ateneo UNICT PIACERI. The authors thank the anonymous reviewers for their helpful comments, in particular one reviewer of a previous submission for suggesting the relation with Galois connections. The authors also thank Mariangiola Dezani-Ciancaglini for her support.

M. H. ter Beek and M. Sirjani (Eds.): COORDINATION 2022, LNCS 13271, pp. 121–139, 2022.
https://doi.org/10.1007/978-3-031-08143-9_8

We propose *formal choreographic languages* (FCL) as a general framework to formalise message-passing systems; existing choreographic models can be conceived as specifications of FCLs. Specifically, we introduce *global* and *local* languages. Global languages (g-languages for short) are made of words built out of *interactions* of the form A→B:m, representing the fact that participant A sends message m to participant B, and participant B receives it. Local languages (l-languages for short) consist of words of *actions* of the forms AB?m and AB!m, respectively representing that participant B receives message m from A and that participant A sends message m to B.

Abstractly such languages consist of runs of a system described in terms of sequences of interactions at the global level and executed through message-passing at the local level. A word w in a global language represents then a possible run expected of a communicating system inducing an expected "local" behaviour on each participant A: the projection of w on A yields the sequence of *output* or *input* actions performed by A along the run w.

Our language-theoretic treatment is motivated mainly by the need for a general setting immune to syntactic restrictions. This naturally leads us to consider e.g., context-free choreographies (cf. Example 3.11). In fact, we strive for generality; basically *prefix-closure* is the only requirement we impose on FCL. The gist is that, if a sequence of interactions or of communications is an observable behaviour of a system, any prefix of the sequence should be observable as well. (We discuss some implications of relaxing prefix-closure in Sect. 8.) This allows us to consider partial executions as well as "complete" ones. We admit infinite words to account for diverging computations, ubiquitous in communication protocols.

Some g-languages cannot be faithfully executed by distributed components; consider $\{\, \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m},\ \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{C}{\to}\mathsf{D}{:}\mathsf{n} \,\}$ that specifies a system where, if occurring, the interaction between C and D has to follow the one between A and B. Clearly, this is not possible if the participants act concurrently because C and D are not aware of when the interaction between A and B takes place.

Contributions and Structure. We summarise below our main contributions. (Proofs and further material can be found in [7].)

Section 2 introduces FCL (g-languages in Definition 2.1, l-languages in Definition 2.2) and adapts standard constructions from the literature. We consider synchronous interactions; the asynchronous case, albeit interesting, is scope for future work (cf. Sect. 8). In particular, we render communicating systems as sets of l-languages (Definition 2.3), while we borrow projections from choreographies and multiparty session types.

Section 3 considers correctness and completeness. An immediate consequence of our constructions is the completeness of systems projected from g-languages (Corollary 3.2). Correctness is more tricky; for it, Definition 3.3 introduces *closure under unknown information* (CUI). Intuitively, a g-language is CUI if it contains extensions of words with a single interaction whose participants cannot distinguish the extended word from other words of the language. Theorem 3.7 characterises correctness of projected systems in terms of CUI.

Section 4 shows how FCLs capture many relevant communication properties in a fairly uniform way.

Section 5 proposes *branch-awareness* (Definition 5.3) to ensure the communication properties defined in Sect. 4 (Theorem 5.6). Intuitively, branch-awareness requires each participant to "distinguish" words where its behaviour differs. Notably, we separate the conditions for correctness from the ones for communication properties. Most approaches in the literature instead combine them into a single condition, which takes names such as well-branchedness or projectability [25]. Thus, these single conditions are stronger than each of CUI and branch-awareness.

Sections 6 and 7 illustrate the generality of FCLs on two case studies, respectively taken from multiparty session types [37] and choreography automata [6]. We remark that FCL can capture protocols that cannot be represented by regular g-languages such as the "task dispatching" protocol in Example 3.11. To the best of our knowledge this kind of protocols cannot be formalised in other approaches.

Section 8 draws some conclusions and discusses future work.

2 Formal Choreographic Languages

We briefly recall a few notions used through the paper. The sets of finite and infinite words on a given alphabet Σ are, respectively, denoted by $\Sigma^\star$ and Σ^ω, where an infinite word on Σ is a map from natural numbers to Σ (aka ω-word [38]). Let $\_\cdot\_$ be the concatenation operator on words and ε its neutral element. We write $a_0 \cdot a_1 \cdot a_2 \cdot \ldots$ for the word mapping i to $a_i \in \Sigma$ for all natural numbers i. A language L on Σ is a subset of $\Sigma^\infty = \Sigma^\star \cup \Sigma^\omega$. The prefix-closure of $L \subseteq \Sigma^\infty$ is $\mathsf{pref}(L) = \{\, z \in \Sigma^\infty \mid \exists z' \in L : \; z \preceq z' \,\}$, where $\preceq$ is the prefix relation; L is *prefix-closed* if $L = \mathsf{pref}(L)$. A word z is *maximal* in a language $L \subseteq \Sigma^\infty$ if $z \preceq z'$ for $z' \in L$ implies $z' = z$. As usual we shall write $z \prec z'$ whenever $z \preceq z'$ and $z \neq z'$.

We shall deal with languages on particular alphabets, namely the alphabets of *interactions* Σ_{int} and of *actions* Σ_{act} whose definitions, borrowed from [6], are as follows[1]

$$\Sigma_{\mathrm{int}} = \{\, \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \mid \mathsf{A} \neq \mathsf{B} \in \mathfrak{P}, \mathsf{m} \in \mathfrak{M} \,\} \qquad \text{ranged over by } \alpha, \beta, \ldots$$

$$\Sigma_{\mathrm{act}} = \{\, \mathsf{A}\,\mathsf{B}!\mathsf{m}, \mathsf{A}\,\mathsf{B}?\mathsf{m} \mid \mathsf{A} \neq \mathsf{B} \in \mathfrak{P}, \mathsf{m} \in \mathfrak{M} \,\} \qquad \text{ranged over by } \mathsf{a}, \mathsf{b}, \ldots$$

where $\mathfrak{P}$ is a fixed set of *participants* (or *roles*, ranged over by A, B, X, etc.) and $\mathfrak{M}$ is a fixed set of *messages* (ranged over by m, x, etc.); we take $\mathfrak{P}$ and $\mathfrak{M}$ disjoint. Let $\mathsf{msg}(\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}) = \mathsf{msg}(\mathsf{A}\,\mathsf{B}!\mathsf{m}) = \mathsf{msg}(\mathsf{A}\,\mathsf{B}?\mathsf{m}) = \mathsf{m}$ and $\mathsf{ptp}(\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}) = \mathsf{ptp}(\mathsf{A}\,\mathsf{B}!\mathsf{m}) = \mathsf{ptp}(\mathsf{A}\,\mathsf{B}?\mathsf{m}) = \{\, \mathsf{A}, \mathsf{B} \,\}$. These functions extend homomorphically to (sets of) words. The *subject* of $\mathsf{A}\,\mathsf{B}!\mathsf{m}$ is the sender A and the subject of $\mathsf{A}\,\mathsf{B}?\mathsf{m}$ is the receiver B. Words on $\Sigma_{\mathrm{int}}^\infty$ (ranged over by $w, w', \ldots$) are called *interaction*

[1] These sets may be infinite; formal languages over infinite alphabets have been studied, e.g., in [4].

words while those on $\Sigma_{\text{act}}^{\infty}$ (ranged over by $v, v', \ldots$) are called *words of actions*. Hereafter $z, z', \ldots$ range over $\Sigma_{\text{int}}^{\infty} \cup \Sigma_{\text{act}}^{\infty}$ and we use $\mathcal{L}$ and $\mathbb{L}$ to range over subsets of, respectively, $\Sigma_{\text{int}}^{\infty}$ and $\Sigma_{\text{act}}^{\infty}$.

A *global language* specifies the expected interactions of a system while a *local language* specifies the communication behaviour of participants.

Definition 2.1 (Global language). *A* global language *(*g-language *for short) is a prefix-closed language $\mathcal{L}$ on Σ_{int}^{∞} such that $ptp(\mathcal{L})$ is finite.*

Definition 2.2 (Local language). *A* local language *(*l-language *for short) is a prefix-closed language $\mathbb{L}$ on Σ_{act} such that $ptp(\mathbb{L})$ is finite. An l-language is* A-local *if its words have all actions with subject* A*.*

As discussed in Sect. 1, l-languages give rise to *communicating systems*.

Definition 2.3 (Communicating system). *Let $\mathcal{P} \subseteq \mathfrak{P}$ be a finite set of participants. A* (communicating) system *over $\mathcal{P}$ is a map $S = (\mathbb{L}_{\mathsf{A}})_{\mathsf{A} \in \mathcal{P}}$ assigning an* A*-local language $\mathbb{L}_{\mathsf{A}} \neq \{\varepsilon\}$ such that $ptp(\mathbb{L}_{\mathsf{A}}) \subseteq \mathcal{P}$ to each participant $\mathsf{A} \in \mathcal{P}$.*

By projecting a g-language $\mathcal{L}$ on a participant A we obtain the A-local language describing the sequence of actions performed by A in the interactions involving A in the words of $\mathcal{L}$.

Definition 2.4 (Projection). *The* projection on A *of an interaction* $\mathsf{B}{\to}\mathsf{C}{:}\mathsf{m}$ *is computed by the function $\_\!\downarrow_{\_} : \Sigma_{int} \times \mathfrak{P} \to \Sigma_{act} \cup \{\varepsilon\}$ defined by:*

$$(\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})\downarrow_{\mathsf{A}} = \mathsf{A}\,\mathsf{B}!\mathsf{m} \qquad (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})\downarrow_{\mathsf{B}} = \mathsf{A}\,\mathsf{B}?\mathsf{m} \qquad (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})\downarrow_{\mathsf{C}} = \varepsilon$$

and extended homomorphically to interaction words and g-languages. The projection of a g-language $\mathcal{L}$*, written $\mathcal{L}\!\downarrow$, is the communicating system $(\mathcal{L}\downarrow_{\mathsf{A}})_{\mathsf{A} \in ptp(\mathcal{L})}$.*

Definition 2.4 recasts in our setting the notion of projection used, e.g., in [13,24].

Example 2.5. Let $\mathcal{L} = \mathsf{pref}(\{\mathsf{C}{\to}\mathsf{A}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m},\ \mathsf{C}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m},\ \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m}\cdot\mathsf{C}{\to}\mathsf{B}{:}\mathsf{m}\})$. By Definition 2.4, we have $\mathcal{L}\!\downarrow = (\mathcal{L}\downarrow_{\mathsf{X}})_{\mathsf{X} \in \{\mathsf{A},\mathsf{B},\mathsf{C}\}}$ where $\mathcal{L}\downarrow_{\mathsf{A}} = \{\varepsilon,\ \mathsf{C}\,\mathsf{A}?\mathsf{m},\ \mathsf{C}\,\mathsf{A}?\mathsf{m}\cdot\mathsf{A}\,\mathsf{B}!\mathsf{m},\ \mathsf{A}\,\mathsf{B}!\mathsf{m}\}$, $\mathcal{L}\downarrow_{\mathsf{B}} = \{\varepsilon,\ \mathsf{A}\,\mathsf{B}?\mathsf{m},\ \mathsf{C}\,\mathsf{B}?\mathsf{m},\ \mathsf{C}\,\mathsf{B}?\mathsf{m}\cdot\mathsf{A}\,\mathsf{B}?\mathsf{m}\}$, and $\mathcal{L}\downarrow_{\mathsf{C}} = \{\varepsilon,\ \mathsf{C}\,\mathsf{A}!\mathsf{m},\ \mathsf{C}\,\mathsf{B}!\mathsf{m},\ \mathsf{C}\,\mathsf{A}!\mathsf{m}\cdot\mathsf{C}\,\mathsf{B}!\mathsf{m}\}$. ◇

We consider a *synchronous* semantics of communicating systems, similarly to other choreographic approaches such as [12,13,15,37]. Intuitively, a choreographic word is in the semantics iff its projection on each participant A yields a word in the local language of A.

Definition 2.6 (Semantics). *Given a system S over $\mathcal{P}$, the set*

$$[\![S]\!] = \{\, w \in \Sigma_{int}^{\infty} \mid ptp(w) \subseteq \mathcal{P} \ \wedge \forall \mathsf{A} \in \mathcal{P} : \ \ w\!\downarrow_{\mathsf{A}} \in S(\mathsf{A}) \,\}$$

is the (synchronous) semantics *of S.*

Notice that the above definition coincides with the *join* operation in [18], used in realisability conditions for an asynchronous setting.

Example 2.7. The semantics $[\![\mathcal{L}\downarrow]\!]$ of the system $\mathcal{L}\downarrow$ in Example 2.5 is the prefix closure of $\{\ \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m},\ \ \mathsf{C}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m},\ \ \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m}\cdot\mathsf{C}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}\ \}$. ◇

Two interactions α and β are *independent* (in symbols $\alpha \parallel \beta$) when $\mathsf{ptp}(\alpha) \cap \mathsf{ptp}(\beta) = \emptyset$. Informally, independent interactions can be swapped. The concurrency closure on infinite words is delicate. One in fact has to allow infinitely many swaps while avoiding that they make an interaction disappear by pushing it infinitely far away. Technically, we consider Mazurkiewicz's traces [33] on Σ_{int} with independence relation $\alpha \parallel \beta$:

Definition 2.8 (Concurrency closure). *Let $\sim$ be the reflexive and transitive closure of the relation $\equiv$ on finite interaction words defined by $w\,\alpha\,\beta\,w' \equiv w\,\beta\,\alpha\,w'$ where $\alpha \parallel \beta$. Following* [Def. 2.1]*[19], $\sim$ extends to Σ^{ω}_{int} by defining*

$$\text{for all } w, w' \in \Sigma^{\omega}_{int}: \qquad w \sim w' \iff w \ll w' \quad \text{and} \quad w' \ll w$$

where $w \ll w'$ iff for each finite prefix w_1 of w there are a finite prefix w'_1 of w' and a g-word $\hat{w} \in \Sigma^{\star}_{int}$ such that $w_1 \cdot \hat{w} \sim w'_1$. A g-language $\mathcal{L}$ is concurrency closed *(c-closed for short) if it coincides with its concurrency closure, namely $\mathcal{L} = \{\, w \in \Sigma^{\infty}_{int} \mid \exists w' \in \mathcal{L} :\ w \sim w' \,\}$.*

Semantics of systems are naturally c-closed since in a distributed setting independent events can occur in any order. Indeed

Proposition 2.9. *Let S be a system. Then $[\![S]\!]$ is c-closed.*

The intuition that g-languages, equipped with the projection and semantic functions of Definition 2.4 and Definition 2.6, do correspond to a natural syntax and semantics for the abstract notion of choreography, can be strengthened by showing that these functions form a Galois connection.

Let us define $\mathbb{G} = \{\, \mathcal{L} \mid \mathcal{L} \text{ is a g-language} \,\}$ and $\mathbb{S} = \{\, S \mid S \text{ is a system} \,\}$. Moreover, given $S, S' \in \mathbb{S}$, we define $S \subseteq S'$ if $S(\mathsf{A}) \subseteq S'(\mathsf{A})$ for each A.

Proposition 2.10. *The functions $\_\downarrow$ and $[\![\_]\!]$ form a (monotone) Galois connection between the posets $(\mathbb{G}, \subseteq)$ and $(\mathbb{S}, \subseteq)$, namely, $\_\downarrow$ and $[\![\_]\!]$ are monotone functions such that, given $\mathcal{L} \in \mathbb{G}$ and $S \in \mathbb{S}$:*

$$\mathcal{L}\downarrow \subseteq S \iff \mathcal{L} \subseteq [\![S]\!]$$

Notice that, by Proposition 2.10, $\mathcal{L}\downarrow \subseteq S$ can be understood as "$\mathcal{L}$ can be realized by S" according to the notion of realisability frequently used in the literature, namely that all behaviours of the choreography are possible for the system.

It is well-known that, given a Galois connection $(f_\star, f^\star)$ the function $\mathsf{cl} = f^\star \circ f_\star$ is a closure operator namely, it is monotone ($x \le y \implies \mathsf{cl}(x) \le \mathsf{cl}(y)$), extensive ($x \le \mathsf{cl}(x)$), and idempotent ($\mathsf{cl}(x) = \mathsf{cl}(\mathsf{cl}(x))$). In our setting $\mathsf{cl}(\_) = [\![\_\downarrow]\!]$, hence the above boils down to the following corollary:

Corollary 2.11. *For all g-languages* $\mathcal{L}, \mathcal{L}' \in \mathbb{G}$,

monotonicity: $\mathcal{L} \subseteq \mathcal{L}' \implies [\![\mathcal{L}\!\downarrow]\!] \subseteq [\![\mathcal{L}'\!\downarrow]\!]$,
extensiveness: $\mathcal{L} \subseteq [\![\mathcal{L}\!\downarrow]\!]$,
idempotency: $[\![\mathcal{L}\!\downarrow]\!] = [\![[\![\mathcal{L}\!\downarrow]\!]\!\downarrow]\!]$.

As we shall see, extensiveness coincides with completeness (Definition 3.1) and, together with monotonicity, implies *harmonicity* (Definition 4.1).

3 Correctness and Completeness

A g-language specifies the expected communication behaviour of a system made of several components. We now define properties relating a communicating system (i.e., a set of l-languages) with a specification (i.e., a g-language).

Definition 3.1 (Correctness and completeness). *A system* S *is* correct *(resp.* complete*)* w.r.t. a g-language $\mathcal{L}$ *if* $[\![S]\!] \subseteq \mathcal{L}$ *(resp.* $[\![S]\!] \supseteq \mathcal{L}$*).*

Correctness and completeness are related to existing notions. For instance, in the literature on multiparty session types (see, e.g., the survey [25]) correctness is analogous to *subject reduction* and completeness to *session fidelity*. Notice that by Proposition 2.10, we can interpret $\mathcal{L}\downarrow \subseteq S$ as a characterisation for completeness of S w.r.t. $\mathcal{L}$.

We discuss now how to ensure correctness and completeness "by construction". Completeness is trivial: it holds for any projected system and coincides with the extensiveness property of the closure operator associated to the Galois connection defined in Sect. 2.

Corollary 3.2. *The projection of a g-language* $\mathcal{L}$ *is complete w.r.t.* $\mathcal{L}$.

We show now how correctness can be characterised as a closure property.

Definition 3.3 (CUI). *A g-language* $\mathcal{L}$ *is* closed under unknown information *(in symbols* $\mathit{cui}(\mathcal{L})$*) if, for all finite words* $w_1 \cdot \alpha, w_2 \cdot \alpha \in \mathcal{L}$ *with the same final interaction* $\alpha = \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \in \Sigma_{int}$, $w \cdot \alpha \in \mathcal{L}$ *for all* $w \in \mathcal{L}$ *such that* $w \downarrow_{\mathsf{A}} = w_1 \downarrow_{\mathsf{A}}$ *and* $w \downarrow_{\mathsf{B}} = w_2 \downarrow_{\mathsf{B}}$.

Intuitively, participants cannot distinguish words with the same projection on their role. Hence, if two participants A and B find words w_1 and w_2 compatible with another word w, and interaction $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$ can occur after both w_1 and w_2, then it should be enabled also after w. Indeed, A (resp. B) cannot know whether the current word is w or w_1 (resp. w_2), hence A and B are willing to take $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$, which can thus happen at the system level. Closure under unknown information (CUI for short) lifts this requirement at the level of g-language.

Example 3.4. The language $\mathcal{L}$ in Example 2.5 is not CUI because it contains the words

$$w_1 \cdot \alpha = \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \qquad w_2 \cdot \alpha = \mathsf{C}{\to}\mathsf{B}{:}\mathsf{m} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \qquad \text{and} \qquad w = \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m} \cdot \mathsf{C}{\to}\mathsf{B}{:}\mathsf{m}$$

and A cannot distinguish between w_1 and w while B cannot distinguish between w_2 and w; nonetheless $w{\cdot}\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} = \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m}{\cdot}\mathsf{C}{\to}\mathsf{B}{:}\mathsf{m}{\cdot}\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \notin \mathcal{L}$. Notice that $w{\cdot}\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \in [\![\mathcal{L}\!\downarrow]\!]$, hence $\mathcal{L} \not\supseteq [\![\mathcal{L}\!\downarrow]\!]$. ◇

The language in Example 3.4 is not the semantics of any system, in fact languages obtained as semantics of a communicating system are always CUI.

Proposition 3.5 (Semantics is CUI). *For all systems S, $[\![S]\!]$ is CUI.*

The next property connects finite and infinite words in a language; it corresponds to the closure under the limit operation used in ω-languages [17,38].

Definition 3.6 (Continuity). *A language L on an alphabet Σ is* continuous *if $z \in L$ for all $z \in \Sigma^\omega$ such that $\mathsf{pref}(z) \cap L$ is infinite.*

This notion of continuity, besides being quite natural, is the most suitable for our purposes among the possible ones [36]. Intuitively, a language L is continuous if an ω-word is in L when infinitely many of its approximants (i.e., finite prefixes) are in L. A g-language $\mathcal{L}$ is *standard or continuous* (sc-language, for short) if either $\mathcal{L} \subseteq \Sigma^\star_{\mathrm{int}}$ or $\mathcal{L}$ is continuous. Notice that for prefix-closed languages for all $z \in L^\omega$ we have that $\mathsf{pref}(z) \cap L$ is infinite iff $\mathsf{pref}(z) \subseteq L$.

Closure under unknown information characterises correct projected systems.

Theorem 3.7 (Characterisation of correctness). *If $\mathcal{L}\!\downarrow$ is correct w.r.t. $\mathcal{L}$ then $\mathsf{cui}(\mathcal{L})$ holds. If $\mathcal{L}$ is an sc-language and $\mathsf{cui}(\mathcal{L})$ then $\mathcal{L}\!\downarrow$ is correct w.r.t. $\mathcal{L}$.*

Notice that CUI is defined in terms of g-languages only, hence checking CUI does not require to build the corresponding system. Also, strengthening the precondition of Definition 3.3 with the additional requirement $w_1 = w_2$ would invalidate Theorem 3.7. Indeed, the language in Example 2.5 would become CUI but not correct. The next example shows that the continuity condition in Theorem 3.7 is necessary for languages containing infinite g-words.

Example 3.8 (Continuity matters). The CUI language

$$\mathcal{L} = \mathsf{pref}\Big(\bigcup_{i\geq 0}\{\,\mathsf{A}{\to}\mathsf{B}{:}\mathsf{l}\cdot\mathsf{B}{\to}\mathsf{C}{:}\mathsf{n}\cdot(\mathsf{C}{\to}\mathsf{D}{:}\mathsf{n})^i\,\} \cup \{\,\mathsf{A}{\to}\mathsf{B}{:}\mathsf{r}\cdot\mathsf{B}{\to}\mathsf{C}{:}\mathsf{n}\cdot(\mathsf{C}{\to}\mathsf{D}{:}\mathsf{n})^\omega\,\}\Big)$$

does contain an infinite word but it is not continuous. The projection of $\mathcal{L}$ is not correct because its semantics contains the g-word $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{l}\cdot\mathsf{B}{\to}\mathsf{C}{:}\mathsf{n}\cdot(\mathsf{C}{\to}\mathsf{D}{:}\mathsf{n})^\omega \notin \mathcal{L}$ since the projections of C and D can exchange infinitely many messages n due to the infinite g-word of $\mathcal{L}$ regardless whether A and B exchange l or r. ◇

Notice that, since $\mathcal{L} \subseteq [\![\mathcal{L}\!\downarrow]\!]$ always holds, Theorem 3.7 implies that $\mathsf{cui}(\mathcal{L})$ characterises the languages $\mathcal{L}$ such that $\mathcal{L} = [\![\mathcal{L}\!\downarrow]\!]$. Besides, the following corollary descends from Theorem 3.7.

Corollary 3.9. *For each sc-language $\mathcal{L}$, $\mathsf{cl}(\mathcal{L})$ is the smallest CUI sc-language containing $\mathcal{L}$.*

CUI ensures that continuous g-languages are c-closed.

Proposition 3.10. *If $\mathcal{L}$ is an sc-language and $\mathsf{cui}(\mathcal{L})$, then $\mathcal{L}$ is c-closed.*

Hence, an sc-language cannot be CUI unless it is c-closed.

As recalled before, in many choreographic formalisms (such as [5,9,14,18,25]) the correctness and completeness of a projected system, namely $\mathcal{L} = [\![\mathcal{L}\!\downarrow]\!]$ (together with some forms of liveness and deadlock-freedom properties), is guaranteed by *well-branchedness* conditions. Most of such conditions guarantee, informally, that participants reach consensus on which branch to take when choices arise. For instance, a well-branchedness condition could be that, at each choice, there is a unique participant deciding the branch to follow during a computation and that such participant informs each other participant. Such a condition is actually not needed to prove $\mathcal{L} = [\![\mathcal{L}\!\downarrow]\!]$. In fact the g-language obtained by adding the word w of Example 3.4 to the language of Example 2.5 is CUI, without being well-branched in the above sense. Indeed, after the interaction $\mathsf{C}{\to}\mathsf{A}{:}\mathsf{m}$, there is a branching in the projected system, since both the interactions $\mathsf{C}{\to}\mathsf{B}{:}\mathsf{m}$ and $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$ can be performed. However, these interactions do not have the same sender.

The next example exhibits a non-regular CUI g-language of finite words. By Theorem 3.7 and Corollary 3.2, the projected system is correct and complete.

Example 3.11 (Task dispatching). As soon as a server (S) communicates its availability (a), a dispatcher (D) sends a task (t) to S. The server either processes the task directly and sends back the resulting data (d) to D or sends the task to participant H for some pre-processing, aiming at resuming it later on. Indeed, after communicating a result to D, the server can resume (r) a previous task (if any) from H, process it, and send the result to D. The server eventually stops by sending s to both D and H; this can happen only when all dispatched tasks have been processed.

This protocol corresponds to the g-language $\mathcal{L} = \mathsf{pref}(\mathfrak{L})$, where $\mathfrak{L}$ is the (non-regular) language generated by the following context-free grammar.

$$S ::= S' \cdot \mathsf{S}{\to}\mathsf{D}{:}\mathsf{s} \cdot \mathsf{S}{\to}\mathsf{H}{:}\mathsf{s} \qquad S' ::= \mathsf{S}{\to}\mathsf{D}{:}\mathsf{a} \cdot \mathsf{D}{\to}\mathsf{S}{:}\mathsf{t} \cdot \mathsf{S}{\to}\mathsf{H}{:}\mathsf{t} \cdot S' \cdot \mathsf{S}{\to}\mathsf{H}{:}\mathsf{r} \cdot \mathsf{H}{\to}\mathsf{S}{:}\mathsf{r} \cdot \mathsf{S}{\to}\mathsf{D}{:}\mathsf{d} \cdot S' \mid \mathsf{S}{\to}\mathsf{D}{:}\mathsf{a} \cdot \mathsf{D}{\to}\mathsf{S}{:}\mathsf{t} \cdot \mathsf{S}{\to}\mathsf{D}{:}\mathsf{d} \cdot S' \mid \varepsilon$$

Since S is involved in all the interactions of $\mathcal{L}$, for each pair of words $w, w' \in \mathcal{L}$: $w\!\downarrow_{\mathsf{S}} = w'\!\downarrow_{\mathsf{S}}$ *iff* $w = w'$. Now, if $w_1\,\alpha, w_2\,\alpha, w \in \mathcal{L}$ satisfy the required conditions for CUI then either $w_1\!\downarrow_{\mathsf{S}} = w\!\downarrow_{\mathsf{S}}$ or $w_2\!\downarrow_{\mathsf{S}} = w\!\downarrow_{\mathsf{S}}$, since $\mathsf{S} \in \mathsf{ptp}(\alpha)$. Hence $\mathsf{cui}(\mathcal{L})$ trivially holds. ◇

The language in Example 3.11 is non-regular since it has the same structure of a language of well-balanced parenthesis. Remarkably, this implies that the g-language cannot be expressed in any other choreographic model we are aware of. The argument used to show $\mathsf{cui}(\mathcal{L})$ in Example 3.11 proves the following.

Proposition 3.12. *If there exists a participant involved in all the interactions of a g-language $\mathcal{L}$ then $\mathsf{cui}(\mathcal{L})$ holds.*

4 Communication Properties

Besides correctness and completeness, other properties could be of interest. For instance, one would like to ensure that participants eventually interact, if they are willing to. We consider a few properties, informally described as follows.

Harmonicity (HA): each sequence of communications that a participant is able to perform can be executed in some computation of the system.
Lock-freedom (LF): if a participant has pending communications to make on an ongoing computation, then there is a continuation of the computation involving that participant.
Strong lock-freedom (SLF): if a participant has pending communications to make on an ongoing computation, then each maximal continuation of the computation involves that participant.
Starvation-freedom (SF): if a participant has pending communications to make on an ongoing computation, then each infinite continuation of the computation involves that participant.
Deadlock-freedom (DF): in all completed computations each participant has no pending actions.

We now formalise the properties above.

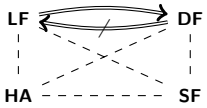
Definition 4.1 (Communication properties). *Let S be a system on $\mathcal{P}$.*

HA *S is* harmonic *if $S(\mathsf{A}) \subseteq [\![S]\!]\!\downarrow_{\mathsf{A}}$ for each $\mathsf{A} \in \mathcal{P}$.*
LF *S is* lock free *if, for each finite word $w \in [\![S]\!]$ and participant $\mathsf{A} \in \mathcal{P}$, if $w\!\downarrow_{\mathsf{A}}$ is not maximal in $S(\mathsf{A})$ then there is a word w' such that $ww' \in [\![S]\!]$ and $w'\!\downarrow_{\mathsf{A}} \neq \varepsilon$.*
SLF *S is* strongly lock free *if, for each finite $w \in [\![S]\!]$ and participant $\mathsf{A} \in \mathcal{P}$, if $w\!\downarrow_{\mathsf{A}}$ is not maximal in $S(\mathsf{A})$ then for each word w' such that ww' is maximal in $[\![S]\!]$ we have $w'\!\downarrow_{\mathsf{A}} \neq \varepsilon$.*
SF *S is* starvation free *if, for each finite $w \in [\![S]\!]$ and participant $\mathsf{A} \in \mathcal{P}$, if $w\!\downarrow_{\mathsf{A}}$ is not maximal in $S(\mathsf{A})$ then $w'\!\downarrow_{\mathsf{A}} \neq \varepsilon$ for each infinite word w' such that $ww' \in [\![S]\!]$.*
DF *S is* deadlock free *if, for each finite and maximal word $w \in [\![S]\!]$ and participant $\mathsf{A} \in \mathcal{P}$, $w\!\downarrow_{\mathsf{A}}$ is maximal in $S(\mathsf{A})$.*

Barred for harmonicity, these properties appear in the literature under different names in various contexts. For instance, the notion of lock-freedom in [5] corresponds to ours, which in turn corresponds to the notion of liveness in [29,32] in a channel-based synchronous communication setting. Likewise, the notion of strong lock-freedom in [37] corresponds to ours and, under fair scheduling, to the notion of lock-freedom in [28]. As a final example, the definition of deadlock-freedom in its (equivalent) contrapositive form, coincides with the notion of progress as defined for synchronous processes in [23,35]. Harmonicity, introduced in the present paper, assures that no behaviour of a participant can be taken out from a system without affecting the overall behaviour of the system itself. Notice that the inverse of harmonicity, $[\![S]\!]\!\downarrow_{\mathsf{A}} \subseteq S(\mathsf{A})$, holds by construction.

The next proposition highlights the relations among our properties.

Proposition 4.2. *The following relations hold among the properties in Definition 4.1*



where implication does not hold in any direction between properties connected by dashed lines

Moreover, DF ∧ SF ⇔ SLF.

5 Communication Properties by Construction

Harmonicity is the only property in Definition 4.1 guaranteed by projection on any system. This can be obtained as a simple consequence of Corollary 3.2.

Corollary 5.1. *If $\mathcal{L}$ is a g-language then $\mathcal{L}\downarrow$ is harmonic.*

The other properties require some conditions on systems to be enjoyed by $\mathcal{L}\downarrow$. Basically, we will strengthen CUI which is too weak. For instance, $\mathsf{cui}(\mathcal{L})$ does imply neither deadlock-freedom nor lock-freedom for $\mathcal{L}\downarrow$.

Example 5.2 (CUI ⇏ DF). Consider the following words

$$w = \mathsf{A}{\to}\mathsf{C}{:}\mathsf{l}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{C}{:}\mathsf{m} \qquad \text{and} \qquad w' = \mathsf{A}{\to}\mathsf{C}{:}\mathsf{r}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{B}{\to}\mathsf{C}{:}\mathsf{m}$$

It is easy to check that the g-language $\mathcal{L} = \mathsf{pref}(\{\, w, w' \,\})$ is CUI. Informally, $\mathsf{cui}(\mathcal{L})$ holds because C can ascertain which of its last actions to execute from the first input. So, Corollary 3.2 and Theorem 3.7 ensure that $\mathcal{L} = [\![\mathcal{L}\downarrow]\!]$. However, $\mathcal{L}\downarrow$ is not deadlock-free. In particular, $w \in \mathcal{L} = [\![\mathcal{L}\downarrow]\!]$ is a deadlock since it is a finite maximal word whose projection on B, namely $w\downarrow_{\mathsf{B}} = \mathsf{A}\,\mathsf{B}?\mathsf{m}$, is not maximal in $\mathcal{L}\downarrow_{\mathsf{B}}$ because $w'\downarrow_{\mathsf{B}} = \mathsf{A}\,\mathsf{B}?\mathsf{m}\cdot\mathsf{B}\,\mathsf{C}!\mathsf{m} \in \mathcal{L}\downarrow_{\mathsf{B}}$.

By Proposition 4.2, the system above is also non lock-free. ◇

In many models (cf. [25]) in order to ensure, besides other properties, also the correctness of $\mathcal{L}\downarrow$, a condition called *well-branchedness* is required. We identify a notion weaker than well-branchedness, which by analogy we dub *branch-awareness* (BA for short).

Definition 5.3 (Branch-awareness). *A participant* X distinguishes *two g-words* $w_1, w_2 \in \Sigma_{int}^{\infty}$ *if*

$$w_1\downarrow_{\mathsf{X}} \neq w_2\downarrow_{\mathsf{X}} \quad \textit{and} \quad w_1\downarrow_{\mathsf{X}} \not\prec w_2\downarrow_{\mathsf{X}} \quad \textit{and} \quad w_2\downarrow_{\mathsf{X}} \not\prec w_1\downarrow_{\mathsf{X}} .$$

A g-language $\mathcal{L}$ on $\mathcal{P}$ is branch-aware *if each $\mathsf{X} \in \mathcal{P}$ distinguishes all maximal words in $\mathcal{L}$ whose projections on X differ.*

Example 5.4. The language $\mathcal{L} = \mathsf{pref}(\{\, w, w' \,\})$ with $w = \mathsf{A}{\to}\mathsf{C}{:}\mathsf{l}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{C}{:}\mathsf{m}$ and $w' = \mathsf{A}{\to}\mathsf{C}{:}\mathsf{r}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{B}{\to}\mathsf{C}{:}\mathsf{m}$ from Example 5.2 is not branch-aware, since $w\downarrow_{\mathsf{B}} = \mathsf{A}\,\mathsf{B}?\mathsf{m}$ and $w'\downarrow_{\mathsf{B}} = \mathsf{A}\,\mathsf{B}?\mathsf{m}\cdot\mathsf{B}\,\mathsf{C}!\mathsf{m}$, hence $w\downarrow_{\mathsf{B}} \neq w'\downarrow_{\mathsf{B}}$ but $w\downarrow_{\mathsf{B}} \prec w'\downarrow_{\mathsf{B}}$. ◇

Condition $w_1 \downarrow_{\mathsf{X}} \neq w_2 \downarrow_{\mathsf{X}}$ in Definition 5.3 is not strictly needed to define BA, but it makes the notion of 'distinguishes' more intuitive. Equivalently, as shown in Proposition 5.5 below, a participant X distinguishes two branches if, after a common prefix, X is actively involved in both branches, performing different interactions.

Proposition 5.5. *Participant* X *distinguishes two g-words* $w_1, w_2 \in \Sigma^{\infty}_{int}$ *iff there are* $w_1' \cdot \alpha_1 \preceq w_1$ *and* $w_2' \cdot \alpha_2 \preceq w_2$ *such that* $w_1' \downarrow_{\mathsf{X}} = w_2' \downarrow_{\mathsf{X}}$ *and* $\alpha_1 \downarrow_{\mathsf{X}} \neq \alpha_2 \downarrow_{\mathsf{X}}$.

The notions of well-branchedness in the literature [25] additionally impose that $\alpha_1 \downarrow_{\mathsf{X}}$ and $\alpha_2 \downarrow_{\mathsf{X}}$ in the above proposition are input actions, but for a (unique) participant (a.k.a., the *selector*) which is required to have different outputs.

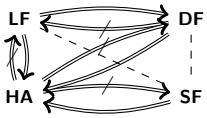
In our case, BA is not needed for correctness, but it is nevertheless useful to prove the communication properties presented in Sect. 4.

Theorem 5.6 (Consequences of BA). *Let* $\mathcal{L}$ *be a branch-aware and CUI sc-language. Then* $\mathcal{L}\downarrow$ *satisfies all the properties in Definition 4.1.*

Example 5.7 (The task dispatching protocol is branch aware). In order to show that the g-language $\mathcal{L}$ in Example 3.11 is branch-aware, we first notice that each maximal word in $\mathcal{L}$ ends with the interactions $\mathsf{S}{\to}\mathsf{D}{:}\mathsf{s}{\cdot}\mathsf{S}{\to}\mathsf{H}{:}\mathsf{s}$. If $\mathcal{L}$ were not branch-aware, there should be two maximal words $w{\cdot}\mathsf{S}{\to}\mathsf{D}{:}\mathsf{s}{\cdot}\mathsf{S}{\to}\mathsf{H}{:}\mathsf{s}$ and $w'{\cdot}\mathsf{S}{\to}\mathsf{D}{:}\mathsf{s}{\cdot}\mathsf{S}{\to}\mathsf{H}{:}\mathsf{s}$ and a participant $\mathsf{X} \in \mathsf{ptp}(\mathcal{L})$ such that $(w{\cdot}\mathsf{S}{\to}\mathsf{D}{:}\mathsf{s}{\cdot}\mathsf{S}{\to}\mathsf{H}{:}\mathsf{s})\downarrow_{\mathsf{X}} \prec (w'{\cdot}\mathsf{S}{\to}\mathsf{D}{:}\mathsf{s}{\cdot}\mathsf{S}{\to}\mathsf{H}{:}\mathsf{s})\downarrow_{\mathsf{X}}$. This is impossible, since w and w' are both generated by the non terminal symbol S' and hence cannot contain the message s. ◇

Proposition 4.2 refines as follows when restricting to projections of g-languages.

Proposition 5.8 *When considering only systems which are projections of g-languages the following relations hold among the properties in Definition 4.1*



where implication does not hold in any direction between properties connected by dashed lines

Moreover, $DF \wedge SF \Leftrightarrow SLF$.

It is not difficult to show that branch-awareness actually characterises SLF for systems obtained by projecting CUI languages.

Proposition 5.9 (Branch-awareness characterises SLF). *A CUI g-language* $\mathcal{L}$ *is branch-aware iff* $\mathcal{L}\downarrow$ *is strongly lock-free.*

6 Global Types as Choreographic Languages

The global types of [37] are our first case study. We recall global types adapting some of the notation in [37] to our setting. Informally, a global type $\mathsf{A} \to \mathsf{B}$:

$\{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n}$ specifies a protocol where participant A must send to B a message m_i for some $1 \le i \le n$ and then, depending on which m_i was chosen by A, the protocol continues as G_i. Global types and multiparty sessions are defined in [37] in terms of the following grammars:

$$\begin{array}{lll} \mathsf{G} ::=^{\text{co}} \mathtt{end} & P ::=^{\text{co}} \mathbf{0} & \mathcal{M} ::= \mathsf{A} \triangleright P \\ \quad \mid \mathsf{A} \to \mathsf{B} : \{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n} & \quad \mid \mathsf{A}?\{\mathsf{m}_i.P_i\}_{1\le i\le n} & \quad \mid \mathcal{M} \mid \mathcal{M} \\ & \quad \mid \mathsf{A}!\{\mathsf{m}_i.P_i\}_{1\le i\le n} & \end{array}$$

respectively for *pre-global types*, *pre-processes*, and *pre-multiparty sessions*. The first two grammars are interpreted coinductively, that is their solutions are both minimal and maximal fixpoints (the latter corresponding to infinite trees) and all messages m_i are pairwise different. A pre-global type G (resp. pre-process P) is a *global type* (resp. *process*) if its tree representation is *regular*, namely it has finitely many distinct sub-trees. A *multiparty session* (MPS for short) is a pre-multiparty session such that (a) in $\mathsf{A} \triangleright P$, participant A does not occur in process P and (b) in $\mathsf{A}_1 \triangleright P_1 \mid \ldots \mid \mathsf{A}_n \triangleright P_n$, participants A_i are pairwise different.

The semantics of global types is the LTS induced by

$$\mathsf{A} \to \mathsf{B} : \{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n} \xrightarrow{\mathsf{A}\to\mathsf{B}:\mathsf{m}_i} \mathsf{G}_i \qquad \mathsf{R} \to \mathsf{S} : \{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n} \xrightarrow{\mathsf{A}\to\mathsf{B}:\mathsf{m}} \mathsf{R} \to \mathsf{S} : \{\mathsf{m}_i.\mathsf{G}'_i\}_{1\le i\le n}$$

where in the latter rule $\{\mathsf{A},\mathsf{B}\} \cap \{\mathsf{R},\mathsf{S}\} = \emptyset$ and for each $1 \le i \le n$, $\mathsf{G}_i \xrightarrow{\mathsf{A}\to\mathsf{B}:\mathsf{m}} \mathsf{G}'_i$. A *branch* is a set $\{\mathsf{m}_i.P_i\}_{1\le i\le n}$ where messages m_i are pairwise distinct.

The semantics for MPSs is the LTS defined by the following rule

$$\mathsf{A} \triangleright \mathsf{B}!(\{\mathsf{m}.P\} \uplus \Lambda) \mid \mathsf{B} \triangleright \mathsf{A}?(\{\mathsf{m}.P'\} \uplus \Lambda') \mid \mathcal{M} \xrightarrow{\mathsf{A}\to\mathsf{B}:\mathsf{m}} \mathsf{A} \triangleright P \mid \mathsf{B} \triangleright P' \mid \mathcal{M} \qquad (1)$$

where $\_ \uplus \_$ is the union of branches defined only on branches with disjoint sets of messages. Rule (1) applies only if the messages in Λ' include those in Λ, which is the case for MPSs obtained by projection, defined below.

Definition 6.1 (Projection *[37,* Definition 3.4]). *The* projection *of* G *on a participant* X *such that the depths of its occurrences in* G *are bounded is the partial function* $\mathsf{G}\upharpoonright_{\mathsf{X}}$ *coinductively defined by* $\mathtt{end}\upharpoonright_{\mathsf{X}} = \mathbf{0}$ *and, for a global type* $\mathsf{G} = \mathsf{A} \to \mathsf{B} : \{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n}$*, by:*

$$\mathsf{G}\upharpoonright_{\mathsf{X}} = \begin{cases} \mathbf{0} & \text{if } \mathsf{X} \text{ is not a participant of } \mathsf{G} \\ \mathsf{B}!\{\mathsf{m}_i.\mathsf{G}_i\upharpoonright_{\mathsf{X}}\}_{1\le i\le n} & \text{if } \mathsf{X} = \mathsf{A} \\ \mathsf{A}?\{\mathsf{m}_i.\mathsf{G}_i\upharpoonright_{\mathsf{X}}\}_{1\le i\le n} & \text{if } \mathsf{X} = \mathsf{B} \\ \mathsf{G}_1\upharpoonright_{\mathsf{X}} & \text{if } \mathsf{X} \notin \{\mathsf{A},\mathsf{B}\} \text{ and } n = 1 \\ \mathsf{S}?(\Lambda_1 \uplus \ldots \uplus \Lambda_n) & \text{if } \mathsf{X} \notin \{\mathsf{A},\mathsf{B}\},\ n > 1,\ \text{and } \forall 1 \le i \le n :\ \mathsf{G}_i\upharpoonright_{\mathsf{X}} = \mathsf{S}?\Lambda_i \end{cases}$$

The global type G *is* projectable[2] *if* $\mathsf{G}\upharpoonright_{\mathsf{X}}$ *is defined for all participants* X *of* G*, in which case* $\mathsf{G}\upharpoonright$ *denotes the corresponding MPS.*

The g-language $\mathcal{L}(\mathsf{G})$ associated to a global type G is the concurrency and prefix closure of $\mathcal{L}'(\mathsf{G})$, that is $\mathcal{L}(\mathsf{G}) = \mathsf{pref}(\{\, w \in \Sigma_{\text{int}}^{\infty} \mid \exists w' \in \mathcal{L}'(\mathsf{G}) :\ w \sim w' \,\})$ where $\mathcal{L}'(\mathsf{G})$ is coinductively defined as follows:

$$\mathcal{L}'(\mathtt{end}) = \{\varepsilon\} \qquad \text{and} \qquad \mathcal{L}'(\mathsf{A} \to \mathsf{B} : \{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n}) = \bigcup_{1\le i\le n} \{\, \mathsf{A}\to\mathsf{B}:\mathsf{m}_i \cdot w \mid w \in \mathcal{L}'(\mathsf{G}_i) \,\}$$

[2] In [37], projectability embeds well-branchedness.

We define the l-language $\mathbb{L}(\mathsf{B} \triangleright P)$ associated to a named process $\mathsf{B} \triangleright P$ as the prefix closure of $\mathbb{L}'(\mathsf{B} \triangleright P)$ which, letting $\star \in \{?, !\}$, is defined by

$$\mathbb{L}'(\mathsf{B} \triangleright \mathbf{0}) = \{\varepsilon\} \quad \text{and} \quad \mathbb{L}'(\mathsf{B} \triangleright \mathsf{A}\star\{\mathsf{m}_i.P_i\}_{1\leq i\leq n}) = \bigcup_{1\leq i\leq n}\{\mathsf{AB}\star\,\mathsf{m}_i \cdot w \mid w \in \mathbb{L}'(P_i)\}$$

The system associated to an MPS is defined as the following map:

$$S(\mathsf{A}_1 \triangleright P_1 \mid \ldots \mid \mathsf{A}_n \triangleright P_n) = \{\, \mathsf{A}_i \mapsto \mathbb{L}(\mathsf{A}_i \triangleright P_i) \mid 1 \leq i \leq n \,\}$$

Our constructions capture relevant properties of the global types in [37]. First, we relate projectability (cf. Definition 6.1) and our properties.

Proposition 6.2. *If* G *is a projectable global type then* $\mathcal{L}(\mathsf{G})$ *is a CUI and branch-aware sc-language.*

This yields the following correspondences between the two frameworks.

Proposition 6.3. *Given a projectable global type* G*,*

$$\mathcal{L}(\mathsf{G}) = \{\, w \mid \mathsf{G} \xrightarrow{w} \} \quad (2) \qquad [\![S(\mathsf{G}\upharpoonright)]\!] = \{\, w \mid \mathsf{G}\upharpoonright \xrightarrow{w} \} \quad (3)$$

Projectable global types are proved strongly lock-free in [37]. The following result corresponds to [37, Theorem 4.7].

Corollary 6.4. $S(\mathsf{G}\upharpoonright)$ *is strongly lock-free for any projectable* G*.*

The symmetry between senders and receivers in CUI and branch-awareness allows for an immediate generalisation of the projection in Definition 6.1 by extending the last case with the clause:

$$\mathsf{S}!(\Lambda_1 \uplus \ldots \uplus \Lambda_n) \qquad \text{if } \mathsf{X} \notin \{\mathsf{A}, \mathsf{B}\},\ n > 1,\ \text{and } \forall 1 \leq i \leq n :\ \mathsf{G}_i \upharpoonright_{\mathsf{X}} = \mathsf{S}!\Lambda_i$$

Corollary 6.4 still holds for this generalised definition of projection.

7 Choreography Automata

Recently we introduced *choreography automata* (c-automata) [6] as an expressive and flexible model of global specifications. A c-automaton $\mathbb{CA} = \langle \mathcal{S}, q_0, \Sigma_{\text{int}}, \rightarrow \rangle$ is a finite-state automaton whose transition relation is labelled in Σ_{int}, namely $\rightarrow \subseteq \mathcal{S} \times \Sigma_{\text{int}} \times \mathcal{S}$ (cf. [7, Def. 8.2]: for the sake of space most of the technical details of this section are in [7]). Observe that the set $\mathcal{P}$ of participants of $\mathbb{CA}$ is necessarily finite. We have some immediate connection between c-automata and FCL by taking as the language $\mathcal{L}(\mathbb{CA})$ of $\mathbb{CA}$ the set of words obtained by concatenating the labels on any of its paths (including infinite paths, cf. [7, Def. 8.3]). In fact $\mathcal{L}(\mathbb{CA})$ is a continuous g-language, that is it is prefix-closed (cf. [7, Prop. 8.4]).

The local behaviour of a participant $\mathsf{A} \in \mathcal{P}$ can be straightforwardly obtained by projecting c-automata on *communicating finite-state machines* (CFSMs) [11].

Basically, a CFSM is a finite-state automaton whose transitions are labelled in Σ_{act} (cf. [7, Def. 8.1]). Formally, the *projection* of a c-automaton $\mathbb{C}\mathsf{A}$ on A, written $\mathbb{C}\mathsf{A}\downarrow_{\mathsf{A}}$, is obtained by determinising up-to-language equivalence the *intermediate* automaton

$$A_{\mathsf{A}} = \langle \mathcal{S}, q_0, \Sigma_{\mathrm{act}} \cup \{\varepsilon\}, \{\, q \xrightarrow{\lambda\downarrow_{\mathsf{A}}} q' \mid q \xrightarrow{\lambda} q' \,\} \rangle$$

Finally, $\mathbb{C}\mathsf{A}\downarrow = (\mathbb{C}\mathsf{A}\downarrow_{\mathsf{A}})_{\mathsf{A}\in\mathcal{P}}$ is the *projection of* $\mathbb{C}\mathsf{A}$(cf. [7, Def. 8.5]).

By applying the definition of language of c-automaton to CFSMs we can associate an l-language $\mathbb{L}(M)$ to each CFSM M (cf. [7, Def. 8.3]). Projections of c-automata and of the corresponding g-languages are related: $\mathbb{L}(\mathbb{C}\mathsf{A}\downarrow_{\mathsf{A}}) = \mathcal{L}(\mathbb{C}\mathsf{A})\downarrow_{\mathsf{A}}$ (cf. [7, Prop. 8.8]).

The synchronous behaviour of a system of CFSMs $(M_{\mathsf{A}})_{\mathsf{A}\in\mathcal{P}}$ can be given as an LTS where states are maps assigning a state in M_{A} to each $\mathsf{A} \in \mathcal{P}$ and transitions are labelled by interactions (or by ε). Intuitively, given a configuration s, if M_{A} and M_{B} have respectively transitions $s(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}} q'_{\mathsf{A}}$ and $s(\mathsf{B}) \xrightarrow{\mathsf{A}\,\mathsf{B}?\mathsf{m}} q'_{\mathsf{B}}$ then $s \xrightarrow{\mathsf{A}\to\mathsf{B}:\mathsf{m}} s[\mathsf{A} \mapsto q'_{\mathsf{A}}, \mathsf{B} \mapsto q'_{\mathsf{B}}]$, where $f[x \mapsto y]$ denotes the update of f on x with y. Likewise, $s(\mathsf{A}) \xrightarrow{\varepsilon} q'_{\mathsf{A}}$ in M_{A} implies $s \xrightarrow{\varepsilon} s[\mathsf{A} \mapsto q'_{\mathsf{A}}]$. Observing that $\mathbb{C}\mathsf{A}\downarrow$ is ε-free, the LTS of $\mathbb{C}\mathsf{A}\downarrow$ is a c-automaton and its language coincides with the g-language of the system $\{\, \mathbb{L}(\mathbb{C}\mathsf{A}\downarrow_{\mathsf{X}}) \,\}_{\mathsf{X}\in\mathcal{P}}$ (cf. [7, Prop. 8.7]).

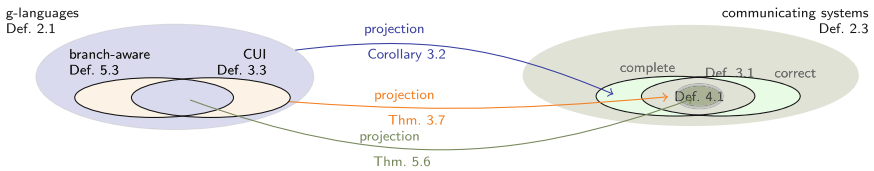


Fig. 1. Contributions of the paper

The communication properties of a system of CFSMs S on $\mathcal{P}$ considered in [6] are liveness, lock-freedom, and deadlock-freedom. We give an intuition of such properties (see [7, Def. 8.9] for a precise account).

- S is *live* when each reachable configuration where a participant $\mathsf{A} \in \mathcal{P}$ can execute a communication has a continuation where A is involved;
- S is *lock-free* when in all computations starting from a reachable configuration where a participant $\mathsf{A} \in \mathcal{P}$ can execute, A is involved;
- S is *deadlock-free* if in none of its reachable configurations s without outgoing transitions there exists $\mathsf{A} \in \mathcal{P}$ willing to communicate.

A system of CFSMs $S = (M_{\mathsf{X}})_{\mathsf{X}\in\mathcal{P}}$ is abstractly represented by the system $\hat{S} = (\mathbb{L}(M_{\mathsf{X}}))_{\mathsf{X}\in\mathcal{P}}$. It is the case that lock-freedom, strong lock-freedom, and deadlock-freedom of $\hat{S}$ (in the sense of Definition 4.1) respectively imply liveness, lock-freedom, and deadlock-freedom of S (cf. [7, Prop. 8.12]).

The conditions on c-automata devised in [6] in order to guarantee the above communication properties in the synchronous case turned out to be flawed. This is shown in [7, Sec. 8.3] (cf. [7, Ex. 8.10]).

Fortunately, the conditions given in the present paper can be applied also in the setting of c-automata. As shown in [7, Sec. 8.4], CUI and branch-awareness are decidable.

8 Concluding Remarks

We developed a general and abstract theory of choreographies based on formal languages, in which we recasted known properties and constructions such as projections from global to local specifications. We briefly recap our main contributions, synoptically depicted in Fig. 1.

One of our contributions is the characterisation of systems' correctness in terms of closure under unknown information (CUI). Other communication properties can be ensured by additionally requiring branch awareness (BA).

Finally, the versatility of FCL allows us to capture existing models. We considered two models chosen according to their "proximity" to FCL. The first model, the variant of MPSTs presented in [37], being based on behavioural types, radically differs from FCL. The second framework, the c-automata in [6], is closer to FCL given that it retraces the connection between automata and formal language theories.

Related Work. The use of formal language theories for the modelling of concurrent systems dates back to the theory of traces [33]. A trace is an equivalence class of words that differ only for swaps of independent symbols. Closure under concurrency corresponds on finite words to form traces, as we noted after Definition 2.8. An extensive literature has explored a notion of realisability whereby a language of traces is realisable if it is accepted by some class of finite-state automata. Relevant results in this respect are the characterisations in [16,39] (and the optimisation in [22]) for finite words and the ones in [19–21] for infinite ones. A key difference of our framework w.r.t. this line of work is that we aim to stricter notions of realisability: in our context it is not enough that the runs of the language may be faithfully executed by a certain class of finite-state automata. Rather we are interested in identifying conditions on the g-languages that guarantee well-behaved executions in "natural" realisations.

Other abstract models of choreographies, e.g. [6,18], have some relation with ours. Conversation protocols (CP) [18], probably the first automata-based model of choreographies, are non-deterministic Büchi automata whose alphabet resembles a constrained variant of our Σ_{int}. A comparison with the g-languages accepted by CPs is not immediate as CPs are based on asynchronous communications (although some connections are evident as noted below Definition 2.6).

Other proposals ascribable to choreographic settings (cf. [25]) define global views that can be seen as g-languages. We focus on synchronous approaches because our current theory needs to be extended to cope with asynchrony.

In [12,31] the correctness of implementations of choreographies (called *choreography conformance*) is studied in a process algebraic setting. The other communication properties we consider here are not discussed there.

The notion of choreography implementation in [12] corresponds to our correctness plus a form of existential termination. It is shown that one can decide whether a system is an implementation of a given choreography, since both languages are generated by finite-state automata, hence language inclusion and existential termination are decidable.

In [31] three syntactic conditions (connectedness, unique points of choice and causality safety) ensure bisimilarity (hence trace equivalence) between a choreography and its projection. Connectedness rules out systems which are not c-closed, while we conjecture that unique points of choice and connectedness together imply our CUI and BA. Causality safety, immaterial in our case, is needed in [31] due to explicit parallel composition.

Many multiparty session type systems [25] have two levels of types (global and local) and one implementation level (local processes). This is the case also for synchronous session type systems such as [15,30]. Our approach, like the session type systems in [5,37], considers only (two) abstract descriptions, g-languages and l-languages. The literature offers several behavioural types featuring correctness-by-construction principles through conditions (known as projectability or well-branchedness) more demanding than ours. For instance, relations similar to those in Sect. 6 can be devised for close formalisms, such as [5] whose notion of projection is more general than the one in [37], yet its notion of projectability still implies CUI and BA.

There is a connection between CUI and the closure property CC2 [3] on message-sequence charts (MSCs) [26]. On finite words CC2 and CUI coincide. Actually, CUI can be regarded as a step-by-step way to ensure CC2 on finite words. The relations between our properties and CC3, also used in MSCs, are still under scrutiny.

Future Work. Our investigation proposes a new point of view for choreography formalisms and the related constructions. As such, a number of extensions and improvements need to be analysed, to check how they may fit in our setting. We list below the most relevant.

First, we need to extend our theory to cope with *asynchronous* communications. While the general approach should apply, it is not immediate how to extend CUI in order to characterize correctness for an asynchronous semantics. This is somehow confirmed by the results in [1,2] on the realisability of MSCs showing that in the asynchronous setting this is a challenging problem.

A second direction is analysing how to drop prefix-closure, so allowing for specifications where the system (and single participants) may stop their execution at some points but not at others; a word would hence represent a complete computation, not only a partial one.

A further direction would unveil the correspondence between closure properties and subtyping relations used in many multiparty session types.

References

1. Alur, R.: The benefits of exposing calls and returns. In: Abadi, M., de Alfaro, L. (eds.) CONCUR 2005. LNCS, vol. 3653, pp. 2–3. Springer, Heidelberg (2005). https://doi.org/10.1007/11539452_2
2. Alur, R., Etessami, K., Yannakakis, M.: Realizability and verification of MSC graphs. In: Orejas, F., Spirakis, P.G., van Leeuwen, J. (eds.) ICALP 2001. LNCS, vol. 2076, pp. 797–808. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-48224-5_65
3. Alur, R., Etessami, K., Yannakakis, M.: Inference of message sequence charts. IEEE Trans. Softw. Eng. **29**(7), 623–633 (2003)
4. Autebert, J.-M., Beauquier, J., Boasson, L.: Langages sur des alphabets infinis. Discrete Appl. Math. **2**(1), 1–20 (1980). http://www.sciencedirect.com/science/article/pii/0166218X80900505. https://doi.org/10.1016/0166-218X(80)90050-5
5. Barbanera, F., Dezani-Ciancaglini, M., Lanese, I., Tuosto, E.: Composition and decomposition of multiparty sessions. J. Log. Algebraic Methods Program. **119**, 100620 (2021). http://www.sciencedirect.com/science/article/pii/S235222082030105X. https://doi.org/10.1016/j.jlamp.2020.100620
6. Barbanera, F., Lanese, I., Tuosto, E.: Choreography automata. In: Bliudze, S., Bocchi, L. (eds.) COORDINATION 2020. LNCS, vol. 12134, pp. 86–106. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50029-0_6
7. Barbanera, F., Lanese, I., Tuosto, E.: Formal choreographic languages (extended version). Technical report, GSSI (2022). https://emwww.github.io/home/tr/fcl.pdf
8. Basu, S., Bultan, T.: Choreography conformance via synchronizability. In: Srinivasan, S., Ramamritham, K., Kumar, A., Ravindra, M.P., Bertino, E., Kumar, R. (eds.) Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, 28 March–1 April 2011, pp. 795–804. ACM (2011). https://doi.org/10.1145/1963405.1963516
9. Basu, S., Bultan, T., Ouederni, M.: Deciding choreography realizability. In: Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2012, Philadelphia, Pennsylvania, USA, 22–28 January 2012, pp. 191–202 (2012). https://doi.org/10.1145/2103656.2103680
10. Bonér, J.: Reactive Microsystems - The Evolution Of Microservices At Scale. O'Reilly (2018)
11. Brand, D., Zafiropulo, P.: On communicating finite-state machines. J. ACM **30**(2), 323–342 (1983)
12. Bravetti, M., Zavattaro, G.: Towards a unifying theory for choreography conformance and contract compliance. In: Lumpe, M., Vanderperren, W. (eds.) SC 2007. LNCS, vol. 4829, pp. 34–50. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77351-1_4
13. Carbone, M., Honda, K., Yoshida, N.: Structured communication-centered programming for web services. ACM Trans. Program. Lang. Syst. **34**(2), 8:1–8:78 (2012). https://doi.org/10.1145/2220365.2220367
14. Coppo, M., Dezani-Ciancaglini, M., Yoshida, N., Padovani, L.: Global progress for dynamically interleaved multiparty sessions. Math. Struct. Comput. Sci. **26**(2), 238–302 (2016)
15. Dezani-Ciancaglini, M., Ghilezan, S., Jaksic, S., Pantovic, J., Yoshida, N.: Precise subtyping for synchronous multiparty sessions. In: Gay, S., Alglave, J. (eds.) Proceedings Eighth International Workshop on Programming Language Approaches

to Concurrency- and Communication-cEntric Software, PLACES 2015, London, UK, 18th April 2015, vol. 203. EPTCS, pp. 29–43 (2015). https://doi.org/10.4204/EPTCS.203.3
16. Duboc, C.: Mixed product and asynchronous automata. TCS **48**(3), 183–199 (1986). https://doi.org/10.1016/0304-3975(86)90094-0
17. Eilenberg, S.: Automata, Languages, and Machines., B. Pure and Applied Mathematics. Academic Press (1976). https://www.worldcat.org/oclc/310535259
18. Xiang, F., Bultan, T., Jianwen, S.: Conversation protocols: a formalism for specification and verification of reactive electronic services. TCS **328**(1–2), 19–37 (2004)
19. Gastin, P.: Infinite traces. In: Guessarian, I. (ed.) LITP 1990. LNCS, vol. 469, pp. 277–308. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-53479-2_12
20. Gastin, P.: Recognizable and rational languages of finite and infinite traces. In: Choffrut, C., Jantzen, M. (eds.) STACS 1991. LNCS, vol. 480, pp. 89–104. Springer, Heidelberg (1991). https://doi.org/10.1007/BFb0020790
21. Gastin, P., Petit, A., Zielonka, W.: A Kleene theorem for infinite trace languages. In: Albert, J.L., Monien, B., Artalejo, M.R. (eds.) ICALP 1991. LNCS, vol. 510, pp. 254–266. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-54233-7_139
22. Genest, B., Muscholl, A.: Constructing exponential-size deterministic Zielonka automata. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 565–576. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_48
23. Ghilezan, S., Jaksic, S., Pantovic, J., Scalas, A., Yoshida, N.: Precise subtyping for synchronous multiparty sessions. J. Log. Algebraic Methods Program. **104**, 127–173 (2019). https://doi.org/10.1016/j.jlamp.2018.12.002
24. Honda, K., Yoshida, N., Carbone, M.: Multiparty asynchronous session types. J. ACM **63**(1), 9:1–9:67 (2016). Extended version of a paper presented at POPL08. https://doi.org/10.1145/2827695
25. Hüttel, H., et al.: Foundations of session types and behavioural contracts. ACM Comput. Surv. **49**(1), 3:1–3:36 (2016)
26. ITU Telecommunication Standardization Sector. ITU-T recommendation Z.120. Message Sequence Charts (MSC'96) (1996)
27. Kavantzas, N., Burdett, D., Ritzinger, G., Fletcher, T., Lafon, Y., Barreto, C.: Web services choreography description language version 1.0. Technical report, W3C (2005). http://www.w3.org/TR/ws-cdl-10/
28. Kobayashi, N.: A type system for lock-free processes. Inf. Comput. **177**, 122–159 (2002)
29. Kobayashi, N., Sangiorgi, D.: A hybrid type system for lock-freedom of mobile processes. ACM Trans. Program. Lang. Syst. **32**(5), 16:1–16:49 (2010). https://doi.org/10.1145/1745312.1745313
30. Kouzapas, D., Yoshida, N.: Globally governed session semantics. Log. Methods Comput. Sci. **10**(4) (2014). https://doi.org/10.2168/LMCS-10(4:20)2014
31. Lanese, I., Guidi, C., Montesi, F., Zavattaro, G.: Bridging the gap between interaction- and process-oriented choreographies. In: Software Engineering and Formal Methods, SEFM 2008, pp. 323–332 (2008)
32. Lange, J., Ng, N., Toninho, B., Yoshida, N.: Fencing off go: liveness and safety for channel-based programming. In: Castagna, G., Gordon, A.D. (eds.) Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, 18–20 January 2017, pp. 748–761. ACM (2017). http://dl.acm.org/citation.cfm?id=3009847

33. Mazurkiewicz, A.: Trace theory. In: Brauer, W., Reisig, W., Rozenberg, G. (eds.) ACPN 1986. LNCS, vol. 255, pp. 278–324. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-17906-2_30
34. OMG. Business Process Model and Notation (BPMN), Version 2.0, January 2011. https://www.omg.org/spec/BPMN
35. Padovani, L.: From lock freedom to progress using session types. In: Yoshida, N., Vanderbauwhede, W. (eds.) Proceedings 6th Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software, PLACES 2013, Rome, Italy, 23rd March 2013, vol. 137. EPTCS, pp. 3–19 (2013). https://doi.org/10.4204/EPTCS.137.2
36. Redziejowski, R.R.: Infinite-word languages and continuous mappings. TCS **43**, 59–79 (1986). https://doi.org/10.1016/0304-3975(86)90166-0
37. Severi, P., Dezani-Ciancaglini, M.: Observational equivalence for multiparty sessions. Fundam. Informaticae **170**(1–3), 267–305 (2019). https://doi.org/10.3233/FI-2019-1863
38. Staiger, L.: ω-languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, pp. 339–387. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-642-59126-6_6
39. Zielonka, W.: Notes on finite asynchronous automata. RAIRO Theor. Informatics Appl. **21**(2), 99–135 (1987). https://doi.org/10.1051/ita/1987210200991