Formal Choreographic Languages

Franco Barbanera, Ivan Lanese, Emilio Tuosto

▶ **To cite this version:**

Franco Barbanera, Ivan Lanese, Emilio Tuosto. Formal Choreographic Languages. COORDINATION 2022 - 24th International Conference on Coordination Models and Languages, Jun 2022, Lucca, Italy. pp.121-139, 10.1007/978-3-031-08143-9_8 . hal-03917266

HAL Id: hal-03917266
https://inria.hal.science/hal-03917266v1

Submitted on 1 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Formal Choreography Languages

Franco Barbanera[1], Ivan Lanese[2], and Emilio Tuosto[3]
barba@dmi.unict.it, ivan.lanese@gmail.com, emilio.tuosto@gssi.it

[1] Dept. of Mathematics and Computer Science, University of Catania (Italy)
[2] Focus Team, University of Bologna/INRIA (Italy)
[3] Gran Sasso Science Institute (Italy)

Abstract. We introduce *formal choreography languages* as a meta-model to study message-passing systems. This allows us to compare and generalise 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 choreography languages capture existing formalisms for message-passing systems; we detail the cases of multiparty session types and choreography automata.

1 Introduction

Choreographic models of message-passing systems are gaining momentum both in academia [7,11,12] and industry [26,33,9]. 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.

We propose *formal choreography 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 $\mathsf{A}\to\mathsf{B}:\mathsf{m}$, 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 $\mathsf{A}\,\mathsf{B}?\mathsf{m}$ and $\mathsf{A}\,\mathsf{B}!\mathsf{m}$, 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.

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 Section 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 & structure We summarise below our main contributions. (Proofs and further material are in the appendixes.)

Section 2 introduces FCL (g-languages in Def. 2.1, l-languages in Def. 2.2) and adapts standard constructions from the literature. In particular, we render communicating systems as sets of l-languages (Def. 2.3), while we borrow projections from choreographies and multi-party 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, Def. 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. Thm. 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* (Def. 5.3) to ensure the communication properties defined in Section 4 (Thm. 5.5). 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 [24]. 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 [36] 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 Ex. 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 Choreography 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 [37]). 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}, defined as follows[4]

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

where $\mathfrak{P}$ is a fixed set of *participants* (or *roles*, ranged over by A, B, 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 $\Sigma^\infty_{\mathrm{int}}$ (ranged over by $w, w', \ldots$) are called *interaction words* while those on $\Sigma^\infty_{\mathrm{act}}$ (ranged over by $v, v', \ldots$) are called *words of actions*. Hereafter $z, z', \ldots$ range over $\Sigma^\infty_{\mathrm{int}} \cup \Sigma^\infty_{\mathrm{act}}$ and use $\mathcal{L}$ and $\mathbb{L}$ to range over subsets of, respectively, $\Sigma^\infty_{\mathrm{int}}$ and $\Sigma^\infty_{\mathrm{act}}$.

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 $\mathsf{ptp}(\mathcal{L})$ 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 $\mathsf{ptp}(\mathbb{L})$ is finite. An l-language is* A-local *if its words have all actions with subject A.*

As discussed in Section 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 $\mathsf{ptp}(\mathbb{L}_\mathsf{A}) \subseteq \mathcal{P}$ 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 $(\mathcal{L}\!\downarrow_\mathsf{A})_{\mathsf{A} \in \mathsf{ptp}(\mathcal{L})}$.*

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

Def. 2.4 recasts in our setting the notion of projection used, e.g., in [12,23].

Example 2.5. Let $\mathcal{L} = \mathsf{pref}(\varepsilon,\ \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m},\ \mathsf{C}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m},\ \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m}\cdot\mathsf{C}{\to}\mathsf{B}{:}\mathsf{m})$. By Def. 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 [11,12,14,36].

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

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

is the (synchronous) semantics *of S.*

Notice that the above definition coincides with the *join* operation in [17], used in realisability conditions for an asynchronous setting.

Example 2.7. The semantics $[\![\mathcal{L}{\downarrow}]\!]$ of the g-language $\mathcal{L}$ in Ex. 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 [32] 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* [18, Def. 2.1]*, $\sim$ extends to Σ_{int}^{ω} by defining*

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

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 $\hat{w} \in \Sigma_{int}^{\star}$ 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 $\mathcal{L} = \{\, w \in \Sigma_{int}^{\infty} \mid \exists w' \in \mathcal{L} :\ w \sim w' \,\}$.*

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 Def. 2.4 and Def. 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 Prop. 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 ($x \leq y \implies \mathsf{cl}(x) \leq \mathsf{cl}(y)$), extensive ($x \leq \mathsf{cl}(x)$), 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 (Def. 3.1) and, together with monotonicity, implies *harmonicity* (Def. 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 appear sometimes under different names. For instance, in the literature on multiparty session types (see, e.g., the survey [24]) correctness is usually called *session fidelity* and completeness is *subject reduction*. Notice that by Prop. 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 Section 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* $\mathsf{cui}(\mathcal{L})$*) 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 Ex. 2.5 is not CUI because it contains the words

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

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

The language in Ex. 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 [16,37].

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 [35]. 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 $\mathsf{pref}(z) \cap L$ is infinite iff $\mathsf{pref}(z) \subseteq L$ for all $z \in L^\omega$.

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. Also, 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 Def. 3.3 with the additional requirement $w_1 = w_2$ would invalidate Thm. 3.7. Indeed, the language in Ex. 2.5 would become CUI but not correct. The next example shows that the continuity condition in Thm. 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, Thm. 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 Thm. 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 [5,24,13,8,17]) 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 Ex. 3.4 to the language of Ex. 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 Thm. 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. Before communicating its availability 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.

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

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 Ex. 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 Ex. 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 $\mathcal{P}$.*

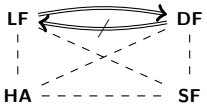
HA *S is* harmonic *if $S(\mathsf{A}) \subseteq [\![S]\!]{\downarrow_{\mathsf{A}}}$ for each $\mathsf{A} \in \mathcal{P}$.*
LF *S is* lock free *if, for each finite word $w \in [\![S]\!]$ and participant $\mathsf{A} \in \mathcal{P}$, if $w{\downarrow_{\mathsf{A}}}$ is not maximal in $S(\mathsf{A})$ then 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 [31,28]

in a channel-based synchronous communication setting. Likewise, the notion of strong lock-freedom in [36] corresponds to ours and, under fair scheduling, to the notion of lock-freedom in [27]. 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 [34,22]. 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 Def. 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 Def. 4.1 guaranteed by projection on any system. This can be obtained as a simple corollary 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 Thm. 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 $w\!\downarrow_{\mathsf{B}} = \mathsf{A}\,\mathsf{B}?\mathsf{m}\cdot\mathsf{B}\,\mathsf{C}!\mathsf{m}$, 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 Prop. 4.2, the system above is also non lock-free. ◇

In many models (cf. [24]) 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*

$$w_1\!\downarrow_{\mathsf{X}} \neq w_2\!\downarrow_{\mathsf{X}} \qquad \textit{and} \qquad w_i\!\downarrow_{\mathsf{X}} \not\approx w_j\!\downarrow_{\mathsf{X}} \ \textit{for all } i \neq j \in \{\,1,2\,\}.$$

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.*

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

Proposition 5.4. *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 [24] 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.

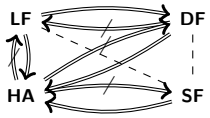
In our case, BA is not needed for correctness, but it is nevertheless useful to prove the communication properties presented in Section 4.

Theorem 5.5 (Consequences of BA). *Let* $\mathcal{L}$ *be a branch-aware and CUI sc-language. Then* $\mathcal{L}\!\downarrow$ *satisfies all the properties in Def. 4.1.*

Example 5.6 (The task dispatching protocol is branch aware). In order to show that the g-language $\mathcal{L}$ in Ex. 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. ◇

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

Proposition 5.7. *When considering only systems which are projections of g-languages the following relations hold among the properties in Def. 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 characterize SLF for systems obtained by projecting CUI languages.

Proposition 5.8 (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 Choreography Languages

The global types of [36] are our first case study. We recall global types adapting some of the notation in [36] 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 [36] 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 [36, 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:*

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

The global type G *is* projectable[5] *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 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:

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

[5] In [36], 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 [36]. First, we relate projectability (cf. Def. 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 [36]. The following result corresponds to [36, Theorem 4.7].

Corollary 6.4. $S(\mathsf{G}\!\downarrow)$ *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 Def. 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 $\mathbb{C}\mathrm{A} = \langle \mathcal{S}, q_0, \Sigma_{\mathrm{int}}, \rightarrow \rangle$ is a finite-state automaton whose transition relation is labelled in Σ_{int}, namely $\rightarrow \subseteq \mathcal{S} \times \Sigma_{\mathrm{int}} \times \mathcal{S}$. (cf. Def. F.2; for the sake of space most of the technical details of this section are in Appendix F.) Observe that the set $\mathcal{P}$ of participants of $\mathbb{C}\mathrm{A}$ is necessarily finite. We have some immediate connection between c-automata and FCL by taking as the language $\mathcal{L}(\mathbb{C}\mathrm{A})$ of $\mathbb{C}\mathrm{A}$ the set of words obtained by concatenating the labels on any of its paths (including infinite paths, cf. Def. F.3). In fact $\mathcal{L}(\mathbb{C}\mathrm{A})$ is a continuous g-language, that is it is prefix-closed (cf. Prop. F.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) [10]. Basically, a CFSM is a finite-state automaton whose transitions are labelled in

Σ_{act} (cf. Def. F.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. Def. F.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. Def. F.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. Prop. F.8).

The synchronous behaviour of 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. Prop. F.7).

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 Def. F.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 Def. 4.1) respectively imply liveness, lock-freedom, and deadlock-freedom of S (cf. Prop. F.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 Appendix F.3 (cf. Ex. F.10). Fortunately, the conditions given in the present paper can be applied also in the setting of c-automata. As shown in Appendix F.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.

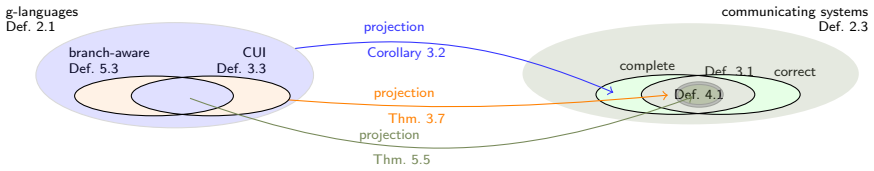


Fig. 1. Contributions of the paper

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 considere two models chosen according to their "proximity" to FCL. The first model, the variant of MPSTs in [36], 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 [32]. 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 Def. 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 [38,15] (and the optimisation in [21]) for finite words and the ones in [18,19,20] 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. [17,6], have some relation with ours. Conversation protocols (CP) [17], 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 Def. 2.6).

Other proposals ascribable to choreographic settings (cf. [24]) 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 [11,30] 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 [11] 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 lan-

guages are generated by finite state automata, hence language inclusion and existential termination are decidable.

In [30] 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 [30] due to explicit parallel composition.

Many multiparty session type systems [24] have two levels of type (global and local) and one implementation level (local processes). This is the case also for synchronous session type systems such as [29,14]. Our approach, like the session type systems in [36,5], 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 Section 6 can be devised for close formalisms, such as [5] whose notion of projection is more general than the one in [36], yet its notion of projectability still implies CUI and BA.

There is a connection between CUI and the closure property CC2 on message-sequence charts (MSCs) [25] introduced in [3]. 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.

Acknowledgements We thank an anonymous reviewer for suggesting the relations with Galois connections.

References

1. Rajeev Alur. The benefits of exposing calls and returns. In Martín Abadi and Luca de Alfaro, editors, *CONCUR 2005 - Concurrency Theory, 16th International*

Conference, CONCUR 2005, San Francisco, CA, USA, August 23-26, 2005, Proceedings, volume 3653 of *Lecture Notes in Computer Science*, pages 2–3. Springer, 2005. doi:10.1007/11539452_2.

2. Rajeev Alur, Kousha Etessami, and Mihalis Yannakakis. Realizability and verification of MSC graphs. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings*, volume 2076 of *Lecture Notes in Computer Science*, pages 797–808. Springer, 2001. doi:10.1007/3-540-48224-5_65.
3. Rajeev Alur, Kousha Etessami, and Mihalis Yannakakis. Inference of Message Sequence Charts. *IEEE Trans. Software Eng.*, 29(7):623–633, 2003.
4. Jean-Michel Autebert, Joffroy Beauquier, and Luc Boasson. Langages sur des alphabets infinis. *Discrete Applied Mathematics*, 2(1):1 – 20, 1980. URL: http://www.sciencedirect.com/science/article/pii/0166218X80900505, doi:https://doi.org/10.1016/0166-218X(80)90050-5.
5. Franco Barbanera, Mariangiola Dezani-Ciancaglini, Ivan Lanese, and Emilio Tuosto. Composition and decomposition of multiparty sessions. *Journal of Logical and Algebraic Methods in Programming*, 119:100620, 2021. URL: http://www.sciencedirect.com/science/article/pii/S235222082030105X, doi:https://doi.org/10.1016/j.jlamp.2020.100620.
6. Franco Barbanera, Ivan Lanese, and Emilio Tuosto. Choreography automata. In Simon Bliudze and Laura Bocchi, editors, *Coordination Models and Languages - 22nd IFIP WG 6.1 International Conference, COORDINATION 2020, Held as Part of the 15th International Federated Conference on Distributed Computing Techniques, DisCoTec 2020, Valletta, Malta, June 15-19, 2020, Proceedings*, volume 12134 of *Lecture Notes in Computer Science*, pages 86–106. Springer, 2020. doi:10.1007/978-3-030-50029-0_6.
7. Samik Basu and Tevfik Bultan. Choreography conformance via synchronizability. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar, editors, *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, pages 795–804. ACM, 2011. doi:10.1145/1963405.1963516.
8. Samik Basu, Tevfik Bultan, and Meriem Ouederni. Deciding choreography realizability. In *Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2012, Philadelphia, Pennsylvania, USA, January 22-28, 2012*, pages 191–202, 2012. doi:10.1145/2103656.2103680.
9. Jonas Bonér. *Reactive Microsystems - The Evolution Of Microservices At Scale*. O'Reilly, 2018.
10. Daniel Brand and Pitro Zafiropulo. On communicating finite-state machines. *J. ACM*, 30(2):323–342, 1983.
11. Mario Bravetti and Gianluigi Zavattaro. Towards a unifying theory for choreography conformance and contract compliance. In Markus Lumpe and Wim Vanderperren, editors, *Software Composition, 6th International Symposium, SC 2007, Braga, Portugal, March 24-25, 2007, Revised Selected Papers*, volume 4829 of *Lecture Notes in Computer Science*, pages 34–50. Springer, 2007.
12. Marco Carbone, Kohei Honda, and Nobuko Yoshida. Structured communication-centered programming for web services. *ACM Trans. Program. Lang. Syst.*, 34(2):8:1–8:78, 2012. doi:10.1145/2220365.2220367.
13. Mario Coppo, Mariangiola Dezani-Ciancaglini, Nobuko Yoshida, and Luca Padovani. Global progress for dynamically interleaved multiparty sessions. *Mathematical Structures in Computer Science*, 26(2):238–302, 2016.

14. Mariangiola Dezani-Ciancaglini, Silvia Ghilezan, Svetlana Jaksic, Jovanka Pantovic, and Nobuko Yoshida. Precise subtyping for synchronous multiparty sessions. In Simon Gay and Jade Alglave, editors, *Proceedings Eighth International Workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software, PLACES 2015, London, UK, 18th April 2015*, volume 203 of *EPTCS*, pages 29–43, 2015. doi:10.4204/EPTCS.203.3.
15. Christine Duboc. Mixed product and asynchronous automata. *TCS*, 48(3):183–199, 1986. doi:10.1016/0304-3975(86)90094-0.
16. Samuel Eilenberg. *Automata, languages, and machines., B.* Pure and applied mathematics. Academic Press, 1976. URL: https://www.worldcat.org/oclc/310535259.
17. Xiang Fu, Tevfik Bultan, and Jianwen Su. Conversation protocols: a formalism for specification and verification of reactive electronic services. *TCS*, 328(1-2):19–37, 2004.
18. Paul Gastin. Infinite traces. In Irène Guessarian, editor, *Semantics of Systems of Concurrent Processes, LITP Spring School on Theoretical Computer Science, La Roche Posay, France, April 23-27, 1990, Proceedings*, volume 469 of *Lecture Notes in Computer Science*, pages 277–308. Springer, 1990. doi:10.1007/3-540-53479-2_12.
19. Paul Gastin. Recognizable and rational languages of finite and infinite traces. In Christian Choffrut and Matthias Jantzen, editors, *STACS 91, 8th Annual Symposium on Theoretical Aspects of Computer Science, Hamburg, Germany, February 14-16, 1991, Proceedings*, volume 480 of *Lecture Notes in Computer Science*, pages 89–104. Springer, 1991. doi:10.1007/BFb0020790.
20. Paul Gastin, Antoine Petit, and Wieslaw Zielonka. A kleene theorem for infinite trace languages. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez-Artalejo, editors, *Automata, Languages and Programming, 18th International Colloquium, ICALP91, Madrid, Spain, July 8-12, 1991, Proceedings*, volume 510 of *Lecture Notes in Computer Science*, pages 254–266. Springer, 1991. doi:10.1007/3-540-54233-7_139.
21. Blaise Genest and Anca Muscholl. Constructing exponential-size deterministic zielonka automata. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 565–576. Springer, 2006. doi:10.1007/11787006_48.
22. Silvia Ghilezan, Svetlana Jaksic, Jovanka Pantovic, Alceste Scalas, and Nobuko Yoshida. Precise subtyping for synchronous multiparty sessions. *J. Log. Algebraic Methods Program.*, 104:127–173, 2019. doi:10.1016/j.jlamp.2018.12.002.
23. Kohei Honda, Nobuko Yoshida, and Marco Carbone. Multiparty asynchronous session types. *Journal of the ACM*, 63(1):9:1–9:67, 2016. Extended version of a paper presented at POPL08. doi:10.1145/2827695.
24. Hans Hüttel, Ivan Lanese, Vasco T. Vasconcelos, Luís Caires, Marco Carbone, Pierre-Malo Deniélou, Dimitris Mostrous, Luca Padovani, António Ravara, Emilio Tuosto, Hugo Torres Vieira, and Gianluigi Zavattaro. Foundations of session types and behavioural contracts. *ACM Comput. Surv.*, 49(1):3:1–3:36, 2016.
25. ITU Telecommunication Standardization Sector. *ITU-T recommendation Z.120. Message Sequence Charts (MSC'96)*, 1996.
26. Nickolas Kavantzas, David Burdett, Gregory Ritzinger, Tony Fletcher, Yves Lafon, and Charlton Barreto. Web services choreography description language version 1.0. Technical report, W3C, 2005. http://www.w3.org/TR/ws-cdl-10/.

27. Naoki Kobayashi. A Type System for Lock-Free Processes. *Information and Computation*, 177:122–159, 2002.
28. Naoki Kobayashi and Davide Sangiorgi. A hybrid type system for lock-freedom of mobile processes. *ACM Trans. Program. Lang. Syst.*, 32(5):16:1–16:49, 2010. doi:10.1145/1745312.1745313.
29. Dimitrios Kouzapas and Nobuko Yoshida. Globally governed session semantics. *Log. Methods Comput. Sci.*, 10(4), 2014. doi:10.2168/LMCS-10(4:20)2014.
30. Ivan Lanese, Claudio Guidi, Fabrizio Montesi, and Gianluigi Zavattaro. Bridging the gap between interaction- and process-oriented choreographies. In *Software Engineering and Formal Methods, SEFM 2008*, pages 323–332, 2008.
31. Julien Lange, Nicholas Ng, Bernardo Toninho, and Nobuko Yoshida. Fencing off go: liveness and safety for channel-based programming. In Giuseppe Castagna and Andrew D. Gordon, editors, *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18-20, 2017*, pages 748–761. ACM, 2017. URL: http://dl.acm.org/citation.cfm?id=3009847.
32. Antoni W. Mazurkiewicz. Trace theory. In Wilfried Brauer, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part II, Proceedings of an Advanced Course, Bad Honnef, Germany, 8-19 September 1986*, volume 255 of *Lecture Notes in Computer Science*, pages 279–324. Springer, 1986. doi:10.1007/3-540-17906-2_30.
33. OMG. *Business Process Model and Notation (BPMN), Version 2.0*, January 2011. https://www.omg.org/spec/BPMN.
34. Luca Padovani. From lock freedom to progress using session types. In Nobuko Yoshida and Wim Vanderbauwhede, editors, *Proceedings 6th Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software, PLACES 2013, Rome, Italy, 23rd March 2013*, volume 137 of *EPTCS*, pages 3–19, 2013. doi:10.4204/EPTCS.137.2.
35. Roman R. Redziejowski. Infinite-word languages and continuous mappings. *TCS*, 43:59–79, 1986. doi:10.1016/0304-3975(86)90166-0.
36. Paula Severi and Mariangiola Dezani-Ciancaglini. Observational equivalence for multiparty sessions. *Fundam. Informaticae*, 170(1-3):267–305, 2019. doi:10.3233/FI-2019-1863.
37. Ludwig Staiger. ω-languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages, Volume 3: Beyond Words*, pages 339–387. Springer, 1997. doi:10.1007/978-3-642-59126-6_6.
38. Wieslaw Zielonka. Notes on finite asynchronous automata. *RAIRO Theor. Informatics Appl.*, 21(2):99–135, 1987. doi:10.1051/ita/1987210200991.

A Proofs for Section 2 (Formal Choreography Languages)

The relation $\sim$ can be characterised as follows.

Lemma A.1. *Given a g-language $\mathcal{L}$ and two words $w_1, w_2 \in \mathcal{L}$, $w_1 \sim w_2$ iff $w_1{\downarrow_\mathsf{A}} = w_2{\downarrow_\mathsf{A}}$ for each $\mathsf{A} \in \mathit{ptp}(\mathcal{L})$.*

Proof. This follows directly from [Gas90, Proposition 2.2]. □

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

Proof. Trivial, since closure under swap does not change the projection by Lemma A.1. □

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 $S \in \mathbb{S}$:*

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

Proof. $(\_){\downarrow}$ and $[\![\_]\!]$ are trivially monotone by their definitions.
($\Longrightarrow$) We first observe that

$$\begin{aligned}
[\![\mathcal{L}{\downarrow}]\!] &= \{\, w' \in \Sigma^\infty_{\mathrm{int}} \mid \forall \mathsf{A} \in \mathcal{P} : \mathsf{ptp}(w') \subseteq \mathcal{P} \ \wedge\ w'{\downarrow_\mathsf{A}} \in \mathcal{L}{\downarrow_\mathsf{A}} \,\} && \text{(Def. 2.6)} \\
&\subseteq \{\, w' \in \Sigma^\infty_{\mathrm{int}} \mid \forall \mathsf{A} \in \mathcal{P} : \mathsf{ptp}(w') \subseteq \mathcal{P} \ \wedge\ w'{\downarrow_\mathsf{A}} \in S(\mathsf{A}) \,\} && \text{(hyp. } \mathcal{L}{\downarrow} \subseteq S\text{)} \\
&= [\![S]\!] && \text{(Def. 2.6)}
\end{aligned}$$

Hence, given a word $w \in \mathcal{L}$, we get $w \in [\![\mathcal{L}{\downarrow}]\!]$ by construction and $w \in [\![S]\!]$ by the above.
($\Longleftarrow$) We have to show that $\mathcal{L}{\downarrow_\mathsf{A}} \subseteq S(\mathsf{A})$ for each A. Let hence $w_\mathsf{A} \in \mathcal{L}{\downarrow_\mathsf{A}}$. By definition there is $w \in \mathcal{L}$ with $w{\downarrow_\mathsf{A}} = w_\mathsf{A}$. Then $w \in [\![S]\!]$ by hypothesis and hence, by Def. 2.6, $w{\downarrow_\mathsf{A}}$ $(= w_\mathsf{A}) \in [\![S]\!]$. □

B Proofs for Section 3 (Correctness and Completeness)

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

Proof. In order to show closure of $[\![S]\!]$ under unknown information, let us take words $w_\mathsf{A} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$, $w_\mathsf{B} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$, $w' \in [\![S]\!]$, such that

$$w_\mathsf{A}{\downarrow_\mathsf{A}} = w'{\downarrow_\mathsf{A}} \qquad \text{and} \qquad w_\mathsf{B}{\downarrow_\mathsf{B}} = w'{\downarrow_\mathsf{B}} \tag{4}$$

By Def. 3.3, we have to show that $w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \in [\![S]\!]$. This in turn, by definition of synchronous semantics, amounts to show that $(w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}){\downarrow_\mathsf{X}} \in S(\mathsf{X})$ for each $\mathsf{X} \in \mathcal{P}$. If $\mathsf{X} \neq \mathsf{A}, \mathsf{B}$ then $(w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}){\downarrow_\mathsf{X}} = w'{\downarrow_\mathsf{X}} \in S(\mathsf{X})$ since $w' \in [\![S]\!]$. Otherwise $\mathsf{X} \in \{\, \mathsf{A}, \mathsf{B} \,\}$; hence, by hypothesis and definition of projection we have

$$(w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}){\downarrow_\mathsf{A}} = w_\mathsf{A}{\downarrow_\mathsf{A}} \cdot \mathsf{A}\,\mathsf{B}!\mathsf{m} \in S(\mathsf{A}) \quad \text{and} \quad (w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}){\downarrow_\mathsf{B}} = w_\mathsf{B}{\downarrow_\mathsf{B}} \cdot \mathsf{A}\,\mathsf{B}?\mathsf{m} \in S(\mathsf{B})$$

where the last two equalities hold by hypothesis. □

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

Proof. We prove the first implication. In order to show closure of $\mathcal{L}$ under unknown information, let us take words $w_{\mathsf{A}} = w'_{\mathsf{A}} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$, $w_{\mathsf{B}} = w'_{\mathsf{B}} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$, $w' \in \mathcal{L}$, such that

$$w'_{\mathsf{A}}\!\downarrow_{\mathsf{A}} = w'\!\downarrow_{\mathsf{A}} \qquad \text{and} \qquad w'_{\mathsf{B}}\!\downarrow_{\mathsf{B}} = w'\!\downarrow_{\mathsf{B}} \tag{5}$$

By Def. 3.3, we have to show that $w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \in \mathcal{L}$. Thanks to correctness, it is enough to show that $w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \in [\![\mathcal{L}\!\downarrow]\!]$. This in turn, by definition of synchronous semantics, amounts to show that $v_{\mathsf{X}} = (w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})\!\downarrow_{\mathsf{X}} \in \mathcal{L}\!\downarrow_{\mathsf{X}}$ for each $\mathsf{X} \in \mathcal{P}$. If $\mathsf{X} \neq \mathsf{A}, \mathsf{B}$ then $v_{\mathsf{X}} = (w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})\!\downarrow_{\mathsf{X}} = w'\!\downarrow_{\mathsf{X}} \in \mathcal{L}\!\downarrow_{\mathsf{X}}$. Otherwise $\mathsf{X} \in \{\mathsf{A}, \mathsf{B}\}$; we consider only the case $\mathsf{X} = \mathsf{A}$ since the other case is analogous. We have $v_{\mathsf{X}} = w_{\mathsf{A}}\!\downarrow_{\mathsf{A}} = w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}\!\downarrow_{\mathsf{A}} \in \mathcal{L}\!\downarrow_{\mathsf{X}}$ as required.

We now prove the second implication. By Def. 3.1, we have to show that $[\![\mathcal{L}\!\downarrow]\!] \subseteq \mathcal{L}$; we proceed by contradiction. Fix a word $w \in [\![\mathcal{L}\!\downarrow]\!] \setminus \mathcal{L}$.

I. $\mathcal{L} \subseteq \Sigma^{\star}_{\mathbf{int}}$ By construction $[\![\mathcal{L}\!\downarrow]\!]$ does not contain infinite g-words since there are only finitely many participants in $\mathcal{L}$. Hence, w is finite and we can take its longest prefix w' that belongs to $\mathcal{L}$. Let $\alpha = \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$ be the interaction immediately following w' in w. We can choose $w_{\mathsf{A}}, w_{\mathsf{B}} \in \mathcal{L}$ such that $w_{\mathsf{A}}\!\downarrow_{\mathsf{A}} = w\!\downarrow_{\mathsf{A}}$ and $w_{\mathsf{B}}\!\downarrow_{\mathsf{B}} = w\!\downarrow_{\mathsf{B}}$. (Recall that (by Def. 2.6) for each $\mathsf{X} \in \mathcal{P}$ there is a word $w_{\mathsf{X}} \in \mathcal{L}$ such that $w\!\downarrow_{\mathsf{X}} = w_{\mathsf{X}}\!\downarrow_{\mathsf{X}}$.) Take the shortest prefixes w'_{A} and w'_{B} of w_{A} and w_{B} respectively such that

$$w'_{\mathsf{A}}\!\downarrow_{\mathsf{A}} = (w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})\!\downarrow_{\mathsf{A}} \qquad \text{and} \qquad w'_{\mathsf{B}}\!\downarrow_{\mathsf{B}} = (w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})\!\downarrow_{\mathsf{B}}$$

It is then easy to check that w'_{A} and w'_{B} have necessarily the following shapes

$$w'_{\mathsf{A}} = \tilde{w}_{\mathsf{A}} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \qquad\qquad w'_{\mathsf{B}} = \tilde{w}_{\mathsf{B}} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$$

where

- $\tilde{w}_{\mathsf{A}}, \tilde{w}_{\mathsf{B}} \in \mathcal{L}$, since $\mathcal{L}$ is prefix-closed;
- $\tilde{w}_{\mathsf{A}}\!\downarrow_{\mathsf{A}} = w'\!\downarrow_{\mathsf{A}}$ and $\tilde{w}_{\mathsf{B}}\!\downarrow_{\mathsf{B}} = w'\!\downarrow_{\mathsf{B}}$.

Now, by $w' \in \mathcal{L}$ and the definition of unknown information closure, we infer that $w' \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \in \mathcal{L}$ against the hypothesis that w' was the longest such prefix or w.

II. $\mathcal{L} \not\subseteq \Sigma^{\star}_{\mathbf{int}}$ If the set $\hat{\mathcal{L}} = \{\, w' \in \Sigma^{\star}_{\mathrm{int}} \mid w' \preceq w \wedge w' \in \mathcal{L} \,\}$ is finite then we take the longest g-word in $\hat{\mathcal{L}}$ and the proof is as in case *I*. Otherwise we have a contradiction because by continuity $w \in \mathcal{L}$.

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

Proof. Given an sc-language $\mathcal{L}$, $\mathsf{cui}(\mathsf{cl}(\mathcal{L}))$ holds by Prop. 3.5. Moreover, it is not difficult to check that if $\mathcal{L}$ is an sc-language, so is $\mathsf{cl}(\mathcal{L})$. $\mathcal{L} \subseteq \mathsf{cl}(\mathcal{L})$ holds by

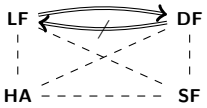
extensiveness of cl. Now, in order to show that $\mathsf{cl}(\mathcal{L})$ is smaller or equal than any CUI sc-language containing $\mathcal{L}$, let us consider any sc-language $\tilde{\mathcal{L}}$ such that $\mathsf{cui}(\tilde{\mathcal{L}})$ and $\mathcal{L} \subseteq \tilde{\mathcal{L}}$. By monotonicity of cl, we have that $\mathsf{cl}(\mathcal{L}) \subseteq \mathsf{cl}(\tilde{\mathcal{L}})$ and, by Thm. 3.7, $\mathsf{cl}(\tilde{\mathcal{L}}) = \hat{\mathcal{L}}$. So $\mathsf{cl}(\mathcal{L}) \subseteq \tilde{\mathcal{L}}$. □

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

Proof. Thanks to Thm. 3.7 $\mathcal{L} = [\![\mathcal{L}\!\downarrow]\!]$, hence it is c-closed thanks to Prop. 2.9. □

C Proofs for Section 4 (Communication Properties)

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



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

Moreover, DF ∧ SF ⇔ SLF.

Proof. **LF $\implies$ DF:** Let us assume S to be lock-free. By contradiction let us assume S not to be deadlock-free. Then there is a finite and maximal word $w \in [\![S]\!]$ and a participant A such that $w\!\downarrow_{\mathsf{A}}$ is not maximal in $S(\mathsf{A})$. Since S is lock-free, by definition, there is a word w' such that $ww' \in [\![S]\!]$ and $w'\!\downarrow_{\mathsf{A}} \neq \varepsilon$. Hence $w' \neq \varepsilon$ and therefore w is not maximal in S, contrary to our assumption.

DF $\not\implies$ LF: Let us take

$$\mathbb{L}_{\mathsf{A}} = \mathsf{pref}(\{\, (\mathsf{A}\,\mathsf{B}!\mathsf{m})^{\infty} \,\}) \quad \mathbb{L}_{\mathsf{B}} = \mathsf{pref}(\{\, (\mathsf{A}\,\mathsf{B}?\mathsf{m})^{\infty} \,\}) \quad \mathbb{L}_{\mathsf{C}} = \mathsf{pref}(\{\, \mathsf{A}\,\mathsf{C}?\mathsf{m} \,\})$$

and consider the communicating system

$$S = (\mathbb{L}_{\mathsf{X}})_{\mathsf{X} \in \{\mathsf{A},\mathsf{B},\mathsf{C}\}} \quad \text{whose synchronous language is} \quad [\![S]\!] = \mathsf{pref}(\{\, (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})^{\infty} \,\})$$

It is immediate to check that S is vacuously deadlock free, since there is no finite maximal word in S. However, S is not lock-free. It is enough to take $w = \varepsilon$, which is finite in $[\![S]\!]$ and such that $w\!\downarrow_{\mathsf{C}} = \varepsilon$ is not maximal in $\mathbb{L}_{\mathsf{C}}$. However there is no w' such that $w \cdot w' \in [\![S]\!]$ and $w'\!\downarrow_{\mathsf{C}} \neq \varepsilon$.

DF $\not\implies$ SF: Let

$$\begin{aligned}
\mathbb{L}_{\mathsf{A}} &= \mathsf{pref}(\{\, (\mathsf{A}\,\mathsf{B}!\mathsf{m})^{\infty},\ \mathsf{A}\,\mathsf{B}!\mathsf{m} \cdot (\mathsf{A}\,\mathsf{C}!\mathsf{m})^{\infty} \,\}) \\
\mathbb{L}_{\mathsf{B}} &= \mathsf{pref}(\{\, (\mathsf{A}\,\mathsf{B}?\mathsf{m})^{\infty} \,\}) \\
\mathbb{L}_{\mathsf{C}} &= \mathsf{pref}(\{\, (\mathsf{A}\,\mathsf{C}?\mathsf{m})^{\infty} \,\})
\end{aligned}$$

and consider the communicating system

$$S = (\mathbb{L}_{\mathsf{X}})_{\mathsf{X} \in \{\mathsf{A},\mathsf{B},\mathsf{C}\}}$$

whose synchronous language is

$$[\![S]\!] = \mathsf{pref}(\{\, (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})^{\infty},\ \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \cdot (\mathsf{A}{\to}\mathsf{C}{:}\mathsf{m})^{\infty} \,\})$$

It is immediate to check that S is vacuously deadlock-free, since there is no finite maximal word in S. However, S is not starvation-free. It is enough to take $w = \mathsf{A{\to}B{:}m}$, which is finite in $[\![S]\!]$ and also $w{\downarrow_\mathsf{C}} = \varepsilon$ is not maximal in $\mathbb{L}_\mathsf{C}$, but for $w' = (\mathsf{A{\to}B{:}m})^\infty$ we have that $w{\cdot}w' \in [\![S]\!]$ and $w'{\downarrow_\mathsf{C}} = \varepsilon$.

SF $\not\Longrightarrow$ **DF:** Let

$$\mathbb{L}_\mathsf{A} = \{\, \varepsilon,\ \mathsf{A\,B!m} \,\} \quad \mathbb{L}_\mathsf{B} = \{\, \varepsilon,\ \mathsf{A\,B?m},\ \mathsf{B\,C!m} \,\} \quad \mathbb{L}_\mathsf{C} = \{\, \varepsilon,\ \mathsf{B\,C?m} \,\}$$

and consider the communicating system

$$S = (\mathbb{L}_\mathsf{X})_{\mathsf{X}\in\{\,\mathsf{A,B,C}\,\}} \quad \text{whose synchronous language is} \quad [\![S]\!] = \{\, \varepsilon,\ \mathsf{A{\to}B{:}m},\ \mathsf{B{\to}C{:}m} \,\}$$

It is immediate to check that S is vacuously starvation free, since there is no infinite word in $[\![S]\!]$. However, S is not deadlock free. It is enough to take $w = \mathsf{A{\to}B{:}m}$, which is finite and maximal in $[\![S]\!]$ but $(\mathsf{A{\to}B{:}m}){\downarrow_\mathsf{C}} = \varepsilon$ is not maximal in $\mathbb{L}_\mathsf{C}$.

LF $\not\Longrightarrow$ **HA:** Let

$$\mathbb{L}_\mathsf{A} = \mathsf{pref}(\{\, \mathsf{B\,A?m{\cdot}B\,A?m},\ \mathsf{B\,A?m{\cdot}A\,B!y} \,\}) \qquad \text{and} \qquad \mathbb{L}_\mathsf{B} = \mathsf{pref}(\{\, \mathsf{B\,A!m{\cdot}B\,A!m} \,\})$$

and consider the communicating system

$$S = (\mathbb{L}_\mathsf{X})_{\mathsf{X}\in\{\,\mathsf{A,B}\,\}} \quad \text{whose synchronous language is} \quad [\![S]\!] = \mathsf{pref}(\{\, \mathsf{B{\to}A{:}m{\cdot}B{\to}A{:}m} \,\})$$

It is easy to check that S is lock-free. However, S is not harmonic, since $\mathsf{B\,A?m\cdot A\,B!y} \in S(\mathsf{A})$ and there is no $w \in [\![S]\!]$ such that $w{\downarrow_\mathsf{A}} = \mathsf{B\,A?m{\cdot}A\,B!y}$.

HA $\not\Longrightarrow$ **LF:** Let us consider the system $S = (\mathbb{L}_\mathsf{X})_{\mathsf{X}\in\{\,\mathsf{A,B,C}\,\}}$ where

$$\begin{aligned}
\mathbb{L}_\mathsf{A} &= \mathsf{pref}(\{\, \mathsf{C\,A?m{\cdot}A\,C!m} \,\}) \\
\mathbb{L}_\mathsf{B} &= \mathsf{pref}(\{\, \mathsf{B\,C!m} \,\}) \\
\mathbb{L}_\mathsf{C} &= \mathsf{pref}(\{\, \mathsf{C\,A!m},\ \mathsf{B\,C?m{\cdot}C\,A!m{\cdot}A\,C?m} \,\})
\end{aligned}$$

It is easy to check that

$$[\![S]\!] = \mathsf{pref}(\{\, \mathsf{C{\to}A{:}m},\ \mathsf{B{\to}C{:}m{\cdot}C{\to}A{:}m{\cdot}A{\to}C{:}m} \,\})$$

and that S is harmonic. However, S is not lock-free. In fact, by taking $w = \mathsf{C{\to}A{:}m} \in [\![S]\!]$, we have that $w{\downarrow_\mathsf{A}}$ is not maximal in $\mathcal{L}_\mathsf{A}$, since and $\mathsf{C\,A?m{\cdot}A\,C!m} = w{\downarrow_\mathsf{A}}{\cdot}\mathsf{A\,C!m} \in \mathcal{L}_\mathsf{A}$. However, there is no word w' such that $w{\cdot}w' \in [\![S]\!]$ and $w'{\downarrow_\mathsf{A}} \neq \varepsilon$.

SF $\not\Longrightarrow$ **LF:** Immediate, since otherwise, by LF $\Rightarrow$ DF proved above, we would get SF to imply DF, which we showed not to hold.

LF $\not\Longrightarrow$ **SF:** Let us consider $S = (\mathcal{L}_\mathsf{X})_{\mathsf{X}\in\{\,\mathsf{A,B,C}\,\}}$ where

$$\begin{aligned}
\mathbb{L}_\mathsf{A} &= \mathsf{pref}(\{\, (\mathsf{A\,B!m})^n{\cdot}\mathsf{A\,C!m}{\cdot}(\mathsf{A\,B!m})^\infty \mid n \in \mathbb{N} \,\} \cup \{\, (\mathsf{A\,B!m})^\infty \,\}) \\
\mathbb{L}_\mathsf{B} &= \mathsf{pref}(\{\, (\mathsf{A\,B?m})^\infty \,\}) \\
\mathbb{L}_\mathsf{C} &= \mathsf{pref}(\{\, \mathsf{A\,C?m} \,\})
\end{aligned}$$

It is easy to check that

$$[\![S]\!] = \mathsf{pref}(\{\, (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})^n \cdot \mathsf{A}{\to}\mathsf{C}{:}\mathsf{m} \cdot (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})^\infty \mid n \in \mathbb{N} \,\} \cup \{\, (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})^\infty \,\})$$

It is not difficult also to check that S is lock-free. However, it is not starvation-free. In fact, by taking the non maximal word $w = \varepsilon$, we have that for the infinite word $w' = (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})^\infty$ and for the participant C we have that $w{\downarrow_{\mathsf{C}}}$ is non maximal and $ww' \in [\![S]\!]$. However $w'{\downarrow_{\mathsf{C}}} = \varepsilon$.

SF $\not\Longrightarrow$ **HA:** Let

$$\mathbb{L}_{\mathsf{A}} = \mathsf{pref}(\{\, \mathsf{B}\,\mathsf{A}?\mathsf{m} \cdot \mathsf{B}\,\mathsf{A}?\mathsf{m},\ \ \mathsf{B}\,\mathsf{A}?\mathsf{m} \cdot \mathsf{A}\,\mathsf{B}!\mathsf{y} \,\}) \qquad \text{and} \qquad \mathbb{L}_{\mathsf{B}} = \mathsf{pref}(\{\, \mathsf{B}\,\mathsf{A}!\mathsf{m} \cdot \mathsf{B}\,\mathsf{A}!\mathsf{m} \,\})$$

and consider the communicating system

$$S = (\mathbb{L}_{\mathsf{X}})_{\mathsf{X} \in \{\, \mathsf{A},\mathsf{B} \,\}} \quad \text{whose synchronous language is} \quad [\![S]\!] = \mathsf{pref}(\{\, \mathsf{B}{\to}\mathsf{A}{:}\mathsf{m} \cdot \mathsf{B}{\to}\mathsf{A}{:}\mathsf{m} \,\})$$

S is trivially starvation-free, since it contains no infinite word. However, S is not harmonic, since $\mathsf{B}\,\mathsf{A}?\mathsf{m} \cdot \mathsf{A}\,\mathsf{B}!\mathsf{y} \in S(\mathsf{A})$ and there is no $w \in [\![S]\!]$ such that $w{\downarrow_{\mathsf{A}}} = \mathsf{B}\,\mathsf{A}?\mathsf{m} \cdot \mathsf{A}\,\mathsf{B}!\mathsf{y}$.

HA $\not\Longrightarrow$ **SF:** Let us consider $S = (\mathcal{L}_{\mathsf{X}})_{\mathsf{X} \in \{\, \mathsf{A},\mathsf{B},\mathsf{C} \,\}}$ where

$$\begin{aligned}
\mathbb{L}_{\mathsf{A}} &= \mathsf{pref}(\{\, (\mathsf{A}\,\mathsf{B}!\mathsf{m})^n \cdot \mathsf{A}\,\mathsf{C}!\mathsf{m} \cdot (\mathsf{A}\,\mathsf{B}!\mathsf{m})^\infty \mid n \in \mathbb{N} \,\} \cup \{\, (\mathsf{A}\,\mathsf{B}!\mathsf{m})^\infty \,\}) \\
\mathbb{L}_{\mathsf{B}} &= \mathsf{pref}(\{\, (\mathsf{A}\,\mathsf{B}?\mathsf{m})^\infty \,\}) \\
\mathbb{L}_{\mathsf{C}} &= \mathsf{pref}(\{\, \mathsf{A}\,\mathsf{C}?\mathsf{m} \,\})
\end{aligned}$$

It is easy to check that

$$[\![S]\!] = \mathsf{pref}(\{\, (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})^n \cdot \mathsf{A}{\to}\mathsf{C}{:}\mathsf{m} \cdot (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})^\infty \mid n \in \mathbb{N} \,\} \cup \{\, (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})^\infty \,\})$$

It is not difficult also to check that S is harmonic. However, it is not starvation-free. In fact, by taking the non maximal word $w = \varepsilon$, we have that for the infinite word $w' = (\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m})^\infty$ and for the participant C we have that $w{\downarrow_{\mathsf{C}}}$ is non maximal and $ww' \in [\![S]\!]$. However $w'{\downarrow_{\mathsf{C}}} = \varepsilon$.

DF $\not\Longrightarrow$ **HA:** Let

$$\mathbb{L}_{\mathsf{A}} = \{\, \varepsilon,\ \mathsf{B}\,\mathsf{A}?\mathsf{m},\ \mathsf{A}\,\mathsf{B}!\mathsf{y} \,\} \qquad \text{and} \qquad \mathbb{L}_{\mathsf{B}} = \{\, \varepsilon,\ \mathsf{B}\,\mathsf{A}!\mathsf{m} \,\}$$

and consider the communicating system

$$S = (\mathbb{L}_{\mathsf{X}})_{\mathsf{X} \in \{\, \mathsf{A},\mathsf{B} \,\}} \quad \text{whose synchronous language is} \quad [\![S]\!] = \{\, \varepsilon,\ \mathsf{B}{\to}\mathsf{A}{:}\mathsf{m} \,\}$$

S is deadlock-free, since the only finite maximal word is $\mathsf{B}{\to}\mathsf{A}{:}\mathsf{m}$ whose projection on A and B are both maximal. However, S is not harmonic, since $\mathsf{A}\,\mathsf{B}!\mathsf{y} \in S(\mathsf{A})$ and there is no $w \in [\![S]\!]$ such that $w{\downarrow_{\mathsf{A}}} = \mathsf{A}\,\mathsf{B}!\mathsf{y}$.

HA $\not\Longrightarrow$ **DF:** Let

$$\mathbb{L}_{\mathsf{A}} = \{\, \varepsilon,\ \mathsf{A}\,\mathsf{B}!\mathsf{m},\ \mathsf{A}\,\mathsf{C}!\mathsf{m} \,\} \qquad \text{and} \qquad \mathbb{L}_{\mathsf{B}} = \{\, \varepsilon,\ \mathsf{A}\,\mathsf{B}?\mathsf{m} \,\} \qquad \text{and} \qquad \mathbb{L}_{\mathsf{C}} = \{\, \varepsilon,\ \mathsf{A}\,\mathsf{C}?\mathsf{m} \,\}$$

and consider the communicating system

$$S = (\mathbb{L}_{\mathsf{X}})_{\mathsf{X}\in\{\mathsf{A},\mathsf{B},\mathsf{C}\}} \quad \text{whose synchronous language is} \quad [\![S]\!] = \{\,\varepsilon,\ \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m},\ \mathsf{A}{\to}\mathsf{C}{:}\mathsf{m}\,\}$$

It is easy to check that S is harmonic. However, S is not deadlock-free, since for the maximal and finite word $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \in S$ we have that $w{\downarrow_{\mathsf{C}}}$ $(= \varepsilon)$ is not maximal in $\mathbb{L}_{\mathsf{C}}$.

DF + SF $\implies$ SLF: In order to show SLF, let $w \in [\![S]\!]$ be finite and let $w{\downarrow_{\mathsf{A}}}$ be non maximal in $S(\mathsf{A})$ for $\mathsf{A} \in \mathcal{P}$. Besides, let w' be such that ww' is maximal in $[\![S]\!]$. We have to show that $w'{\downarrow_{\mathsf{A}}} \neq \varepsilon$. We distinguish two cases:
w' **is finite.** Since S is DF we can infer that $w' \neq \varepsilon$ and $w'{\downarrow_{\mathsf{A}}} \neq \varepsilon$.
w' **is infinite.** We get $w'{\downarrow_{\mathsf{A}}} \neq \varepsilon$ immediately by SF.

SLF $\implies$ DF + SF : To show DF by contradiction, let us assume to have $w \in [\![S]\!]$ finite and maximal and such that $w{\downarrow_{\mathsf{A}}}$ is non maximal in $S(\mathsf{A})$ for $\mathsf{A} \in \mathcal{P}$. By SLF we have that $w'{\downarrow_{\mathsf{A}}} \neq \varepsilon$ for each w' such that ww' is maximal in $[\![S]\!]$. Since we assumed w to be maximal, the only possible w' is ε, contradicting $w'{\downarrow_{\mathsf{A}}} \neq \varepsilon$.
To show SF by contradiction, let us assume to have $w \in [\![S]\!]$ finite and such that $w{\downarrow_{\mathsf{A}}}$ is non maximal in $S(\mathsf{A})$ for $\mathsf{A} \in \mathcal{P}$. Besides, let us assume that there exists an infinite w' such that $ww' \in [\![S]\!]$ and $w'{\downarrow_{\mathsf{A}}} = \varepsilon$. Since w' is infinite, ww' is trivially maximal in $[\![S]\!]$ and we get a contradiction with SLF. □

D Proofs for Section 5 (Communication Properties by Construction)

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

Proof. By Corollary 3.2, $\mathcal{L} \subseteq [\![\mathcal{L}{\downarrow}]\!]$. Now, by monotonicity of projection, we get $\mathcal{L}{\downarrow} \subseteq [\![\mathcal{L}{\downarrow}]\!]{\downarrow}$, that is harmonicity of $\mathcal{L}{\downarrow}$. □

Proposition 5.4. *Participant X distinguishes two g-words $w_1, w_2 \in \Sigma_{int}^{\infty}$ 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 $\alpha_1{\downarrow_{\mathsf{X}}} \neq \alpha_2{\downarrow_{\mathsf{X}}}$.*

Proof. Trivial □

Theorem 5.5 (Consequences of BA). *Let $\mathcal{L}$ be a branch-aware and CUI sc-language. Then $\mathcal{L}{\downarrow}$ satisfies all the properties in Def. 4.1.*

Proof. Let $\mathcal{L}$ be a branch-aware sc-language such that $\mathsf{cui}(\mathcal{L})$ holds. We prove the properties separately.

Harmonicity Immediate by Corollary 5.1.

Lock-freedom By contradiction, let us assume $\mathcal{L}{\downarrow}$ not to be lock-free. By Def. 4.1 (lock-freedom) and $\mathsf{cui}(\mathcal{L})$, it follows that there exist a participant $\mathsf{A} \in \mathcal{P}$ and a finite g-word $w \in \mathcal{L}$ such that
- $w{\downarrow_{\mathsf{A}}}$ is not maximal in $\mathcal{L}{\downarrow_{\mathsf{A}}}$;

– $\forall w'.\,(ww' \in \mathcal{L} \;\Rightarrow\; w'{\downarrow_{\mathsf{A}}} = \varepsilon)$.

By Corollary 5.1, $\mathcal{L}{\downarrow}$ is harmonic. Hence, by the above and $\mathsf{cui}(\mathcal{L})$, there exists $w'' \in \mathcal{L}$ such that

– $w'' \neq w$;
– $w''{\downarrow_{\mathsf{A}}} = w{\downarrow_{\mathsf{A}}}$;
– $\exists \hat{w}.\,(w''\hat{w} \in \mathcal{L}$ and $\hat{w}{\downarrow_{\mathsf{A}}} \neq \varepsilon)$.

This means that, by taking a maximal extension of w and a maximal extension of $w''\hat{w}$, we would get two non branch-aware words in $\mathcal{L}$, contradicting our hypothesis of $\mathcal{L}$ being branch-aware.

Deadlock-freedom Immediate by Prop. 4.2.

Starvation-freedom By contradiction, let us assume $\mathcal{L}{\downarrow}$ not to be starvation-free. By Def. 4.1 (starvation-freedom) and $\mathsf{cui}(\mathcal{L})$, it follows that there exist a participant $\mathsf{A} \in \mathcal{P}$ and a finite g-word $w \in \mathcal{L}$ such that

– $w{\downarrow_{\mathsf{A}}}$ is not maximal in $\mathcal{L}{\downarrow_{\mathsf{A}}}$;
– $\exists w'.\,(w'$ is infinite,$ww' \in \mathcal{L}$ and $w'{\downarrow_{\mathsf{A}}} = \varepsilon)$.

Now, by harmonicity of $\mathcal{L}{\downarrow}$ (Corollary 5.1), non maximality of $w{\downarrow_{\mathsf{A}}}$ and by $\mathsf{cui}(\mathcal{L})$, it follows that there exist $w'' \in \mathcal{L}$ and a finite word $\hat{w}$ such that

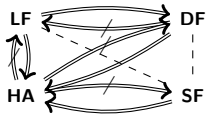
– $w''\hat{w} \in \mathcal{L}$;
– $w''{\downarrow_{\mathsf{A}}} = w{\downarrow_{\mathsf{A}}}$;
– $\hat{w}{\downarrow_{\mathsf{A}}} \neq \varepsilon$.

The above means that by taking ww' and any maximal extension of $w''\hat{w}$ we would get two maximal words in $\mathcal{L}$ which A cannot distinguish, so contradicting our hypothesis of $\mathcal{L}$ being branch-aware.

Strong lock-freedom Immediate since $\mathbf{SLF} = \mathbf{SF} + \mathbf{DF}$.

□

Proposition 5.7. *When considering only systems which are projections of g-languages the following relations hold among the properties in Def. 4.1*



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

Moreover, DF ∧ SF ⇔ SLF.

Proof. **LF** $\Longrightarrow$ **DF:** The proof of Prop. 4.2 applies.
DF $\not\Longrightarrow$ **LF:** Let us take the language

$$\mathcal{L} = \mathsf{pref}(\{\, \mathsf{A}{\to}\mathsf{C}{:}\mathsf{n}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{C}{:}\mathsf{m}\cdot(\mathsf{A}{\to}\mathsf{C}{:}\mathsf{m})^{\infty},\;\; \mathsf{A}{\to}\mathsf{C}{:}\mathsf{m}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}\cdot\mathsf{B}{\to}\mathsf{C}{:}\mathsf{m}\cdot(\mathsf{A}{\to}\mathsf{C}{:}\mathsf{m})^{\infty} \,\})$$

The system $\mathcal{L}{\downarrow}$ is trivially DF since its maximal words are all infinite, while it is not LF since the non maximal word $\mathsf{A}{\to}\mathsf{C}{:}\mathsf{n}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \in [\![\mathcal{L}{\downarrow}]\!]$ is such that its projection on B is not maximal in $\mathcal{L}{\downarrow_{\mathsf{B}}}$ and no continuation in $\mathcal{L}$ of such word contains interactions involving B; in fact for the (only) g-word $\hat{w} = \mathsf{A}{\to}\mathsf{C}{:}\mathsf{m}\cdot(\mathsf{A}{\to}\mathsf{C}{:}\mathsf{m})^{\infty}$ such that $\mathsf{A}{\to}\mathsf{C}{:}\mathsf{n}\cdot\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}\cdot\hat{w} \in [\![\mathcal{L}{\downarrow}]\!]$, we have $\hat{w}{\downarrow_{\mathsf{B}}} = \varepsilon$.
DF $\not\Longrightarrow$ **SF:** The counterexample of Prop. 4.2 applies.
SF $\not\Longrightarrow$ **DF:** The counterexample of Prop. 4.2 applies.
LF $\Longrightarrow$ **HA:** Trivial since HA always holds thanks to Corollary 5.1.

HA $\not\Longrightarrow$ **LF:** Trivial since HA always holds thanks to Corollary 5.1, while this is not the case for LF thanks to Ex. 5.2.
SF $\not\Longrightarrow$ **LF:** The proof of Prop. 4.2 applies.
LF $\not\Longrightarrow$ **SF:** The counterexample of Prop. 4.2 applies.
SF $\Longrightarrow$ **HA:** Trivial since HA always holds thanks to Corollary 5.1.
HA $\not\Longrightarrow$ **SF:** The counterexample of Prop. 4.2 applies.
DF $\Longrightarrow$ **HA:** Trivial since HA always holds thanks to Corollary 5.1.
HA $\not\Longrightarrow$ **DF:** Trivial since HA always holds thanks to Corollary 5.1, while this is not the case for DF thanks to Ex. 5.2.
DF + SF $\Longrightarrow$ **SLF:** The proof of Prop. 4.2 applies.
SLF $\Longrightarrow$ **DF + SF :** The proof of Prop. 4.2 applies.

□

Proposition 5.8 (Branch-awareness characterises SLF). *A CUI g-language $\mathcal{L}$ is branch-aware iff $\mathcal{L}\downarrow$ is strongly lock-free.*

Proof. ($\Rightarrow$) Thm. 5.5.
($\Leftarrow$) By contraposition, let us assume $\mathcal{L}$ not to be branch-aware. Then, by Def. 5.3 there exist a participant X and two maximal words in $\mathcal{L}$ such that their projections on X differ and $w_i\downarrow_{\mathsf{X}} \prec w_j\downarrow_{\mathsf{X}}$ for either $i = 1$ or $i = 2$. Let us assume, without loss of generality that $i = 1$. Let now w_1' and w_2' be the maximal prefixes (hence finite) of, respectively w_1 and w_2 such that $w_1'\downarrow_{\mathsf{X}} = w_2'\downarrow_{\mathsf{X}}$. We distinguishing the following two possible cases.

$w_1' = w_1$.
In such a case we get that $\mathcal{L}\downarrow$ is not deadlock-free, since, for the finite maximal w we have that $w_1\downarrow_{\mathsf{X}}$ is not maximal. So, by Prop. 5.7, $\mathcal{L}\downarrow$ is not strongly lock-free as well.
$w_1' \prec w_1$.
In such a case we get that w_1' is a non maximal finite word such that $w_1 = w_1'\cdot w_1''$ is maximal, $w_1\downarrow_{\mathsf{X}}$ is non maximal and $w_1''\downarrow_{\mathsf{X}} = \varepsilon$, that is $\mathcal{L}\downarrow$ is not strongly lock-free.

□

E Proofs for Section 6 (Global Types as Choreography Languages)

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

Proof. By construction, for any G, either $\mathcal{L}(\mathsf{G}) \subseteq \Sigma_{\text{int}}^{\star}$ or $\mathcal{L}(\mathsf{G})$ is continuous. We begin with branch-awareness. Let us assume $\mathcal{L}(\mathsf{G})$ not to be branch-aware. Then there exist two distinct maximal $w, w' \in \mathcal{L}(\mathsf{G})$ such that $w \not\sim w'$ and their projections on a certain participant A are one a strict prefix of the other. Without loss of generality, let us assume $w'\downarrow_{\mathsf{A}} \prec w\downarrow_{\mathsf{A}}$. By definition of $\mathcal{L}(\_)$ there exist two maximal $\hat{w}, \hat{w}' \in \mathcal{L}'(\mathsf{G})$ such that $\hat{w} \not\sim \hat{w}$ and $\hat{w} \sim w$ and $\hat{w}' \sim w'$.

By definitions of $\sim$ and $(\_)\!\downarrow_{\_}$ we get that $\hat{w}\downarrow_{\mathsf{A}} = w\downarrow_{\mathsf{A}}$ and $\hat{w}'\downarrow_{\mathsf{A}} = w'\downarrow_{\mathsf{A}}$. So, we have that $\hat{w}'\downarrow_{\mathsf{A}} \prec \hat{w}\downarrow_{\mathsf{A}}$. Let w'' be the longest common prefix of $\hat{w}$ and $\hat{w}'$, that is $\hat{w} = w''\tilde{w}$ and $\hat{w}' = w''\tilde{w}'$. Then $\tilde{w} \not\sim \tilde{w}'$ and $\tilde{w}'\downarrow_{\mathsf{A}} \prec \tilde{w}\downarrow_{\mathsf{A}}$. Moreover. there exists a subterm (subtree) G' of G such that $\tilde{w}, \tilde{w}' \in \mathcal{L}'(\mathsf{G}')$. By definition of projectability this is however impossible since, by definition of projectability again, if G is projectable so is G'.

We proceed now with CUI. As observed above, from the definitions of $\sim$ and $(\_)\!\downarrow_{\_}$ it follows that, for each w and w' and A, if $w \sim w'$ then $w\downarrow_{\mathsf{A}} = w'\downarrow_{\mathsf{A}}$. From this we can infer that $\mathsf{cui}(\mathcal{L}'(\mathsf{G}))$ implies $\mathsf{cui}(\mathcal{L}(\mathsf{G}))$. We then show $\mathsf{cui}(\mathcal{L}'(\mathsf{G}))$. Let $w_1 \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}, w_2 \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}, w \in \mathcal{L}(\mathsf{G})$ such that $w_1\downarrow_{\mathsf{A}} = w\downarrow_{\mathsf{A}}$, $w_2\downarrow_{\mathsf{B}} = w\downarrow_{\mathsf{B}}$. Let now w_0 be the longest common prefix of w_1, w_2 and w, and let w_0' be the longest common prefix of w_1 and w (the other possibilities for the common prefixes of the words w_1, w_2 and w can be treated in a similar way). Hence, by definition of global type, we have that

- $w_1 = w_0\mathsf{C}{\to}\mathsf{D}{:}\mathsf{m}\cdot w_1'$ and $w = w_0\mathsf{C}{\to}\mathsf{D}{:}\mathsf{n}\cdot w'$ for some $\mathsf{C}, \mathsf{D}, \mathsf{m}, \mathsf{n}$ with $\mathsf{m} \neq \mathsf{n}$; and
- $w_2 = \mathsf{E}{\to}\mathsf{F}{:}\mathsf{x}\cdot w_2'$ and $w = \mathsf{E}{\to}\mathsf{F}{:}\mathsf{y}\cdot w''$ for some $\mathsf{E}, \mathsf{F}, \mathsf{x}, \mathsf{y}$ with $\mathsf{x} \neq \mathsf{y}$.

We now observe that $\mathsf{A} \neq \mathsf{C}$ and $\mathsf{A} \neq \mathsf{D}$, since we assumed $w_1\downarrow_{\mathsf{A}} = w\downarrow_{\mathsf{A}}$. Moreover, $\mathsf{B} \neq \mathsf{E}$ and $\mathsf{B} \neq \mathsf{F}$, since we assumed $w_2\downarrow_{\mathsf{B}} = w\downarrow_{\mathsf{B}}$.
Now, by definition of projection, from $\mathsf{A} \neq \mathsf{C}$ and $\mathsf{A} \neq \mathsf{D}$ we can infer that $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$ belongs to any maximal words having w as prefix. Let us consider the first occurrence of $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$ in one maximal words having w as prefix. In order to complete the proof, we need to show that no interaction of the form $\mathsf{L}{\to}\mathsf{B}{:}\mathsf{z}$ is present after w and before $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$. This is not possible since, by definition of projection, from $\mathsf{B} \neq \mathsf{E}$ and $\mathsf{B} \neq \mathsf{F}$, we should have $\mathsf{L} = \mathsf{A}$. Than we should also have, by definition of projection again and from $\mathsf{A} \neq \mathsf{C}$ and $\mathsf{A} \neq \mathsf{D}$, that $\mathsf{z} \neq \mathsf{m}$, so contradicting that we had considered the first occurrence of $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$. □

The following auxiliary lemma is useful in the proof of Prop. 6.3.

Lemma E.1. *If* $\mathsf{G}\upharpoonright_{\mathsf{A}}$ *is defined then* $w \in \mathcal{L}'(\mathsf{G})$ *implies* $w\downarrow_{\mathsf{A}} \in \mathbb{L}'(\mathsf{G}\upharpoonright_{\mathsf{A}})$.

Proof. By coinduction on G. If $\mathsf{G} = \mathtt{end}$ the thesis trivially hold. If $\mathsf{G} = \mathsf{A} \to \mathsf{B} : \{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n}$ or $\mathsf{G} = \mathsf{B} \to \mathsf{A} : \{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n}$ then the thesis immediately follows by coinduction observing that $w = \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}_i \cdot w'$ or $w = \mathsf{B}{\to}\mathsf{A}{:}\mathsf{m}_i \cdot w'$ for some $1 \le i \le n$ and $w' \in \mathcal{L}'(\mathsf{G}_i)$. If $\mathsf{G} = \mathsf{R} \to \mathsf{S} : \{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n}$ with $\mathsf{A} \neq \mathsf{R}$ and $\mathsf{A} \neq \mathsf{S}$ then $w = \mathsf{R}{\to}\mathsf{S}{:}\mathsf{m}_i \cdot w'$ for some $1 \le i \le n$ and $w' \in \mathcal{L}'(\mathsf{G}_i)$; hence $w\upharpoonright_{\mathsf{A}} = w'\upharpoonright_{\mathsf{A}}$. By Def. 6.1, either (a) $n = 1$ and $\mathsf{G}\upharpoonright_{\mathsf{A}} = \mathsf{G}_1\upharpoonright_{\mathsf{A}}$ or (b) $n > 1$ and $\mathsf{G}\upharpoonright_{\mathsf{A}} = \mathsf{S}?(\Lambda_1 \uplus \ldots \uplus \Lambda_n)$ where, for all $1 \le i \le n$, $\mathsf{G}_i\upharpoonright_{\mathsf{A}} = \mathsf{S}?\Lambda_i$ and Λ_i of the form $\{\mathsf{m}_1^i : \mathsf{G}_1^i, \ldots, \mathsf{m}_{n_i}^i : \mathsf{G}_{n_i}^i\}$. In case (a) the thesis immediately follows by coinduction observing that $i = n = 1$. In case (b), $w'\upharpoonright_{\mathsf{A}} = \mathsf{S}?\mathsf{m}_{ij}{\cdot}v$ with $1 \le j \le n_i$ and, coinductively, $v \in \mathbb{L}'(\mathsf{A} \triangleright \mathsf{G}_i\upharpoonright_{\mathsf{A}})$ the projection on A of some word in $\mathcal{L}'(\mathsf{G}_i)$ otherwise we would violate the projectability of G on A.

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

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

Proof. (2) We first show $\mathcal{L}'(\mathsf{G}) \subseteq \{\, w \mid \mathsf{G} \xrightarrow{w} \,\}$. If $\mathsf{G} = \mathtt{end}$ then the thesis trivially follows since $\mathsf{G} \xrightarrow{\varepsilon}$. If $\mathsf{G} = \mathsf{A} \to \mathsf{B} : \{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n}$ then each word in $\mathcal{L}'(\mathsf{G})$ begins with an interaction $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}_i$ with $1 \le i \le n$. By the reduction semantics of global types, $\mathsf{G} \xrightarrow{\mathsf{A}\to\mathsf{B}:\mathsf{m}_i} \mathsf{G}_i$. By coinduction [KS17] we obtain the thesis. Each word in $\mathcal{L}(\mathsf{G})$ is a prefix of a concurrency equivalent word $w \in \mathcal{L}'(\mathsf{G})$. Hence we immediately have that $\mathsf{G} \xrightarrow{w}$ from the previous argument by a possibly repeatedly use the second rule of the reduction semantics of global types to obtain the thesis.

Case $\supseteq$. If $\mathsf{G} \xrightarrow{w}$ then, for some G' we have $\mathsf{G} \xrightarrow{\alpha} \mathsf{G}' \xrightarrow{w'}$ where $w = \alpha \cdot w'$. If $\mathsf{G} \xrightarrow{\alpha} \mathsf{G}'$ is an instance of the first rule of the operational semantics of global types then the thesis follows by coinduction. Otherwise, assuming $\mathsf{G} = \mathsf{R} \to \mathsf{S} : \{\mathsf{m}_i.\mathsf{G}_i\}_{1\le i\le n}$, for each $1 \le i \le n$ we have (i) $\mathsf{G}_i \xrightarrow{\alpha} \mathsf{G}_i'$, (ii) $\mathsf{ptp}(\alpha) \cap \{\mathsf{R}, \mathsf{S}\} = \emptyset$, and (iii) $\mathsf{G}' = \mathsf{R} \to \mathsf{S} : \{\mathsf{m}_i.\mathsf{G}_i'\}_{1\le i\le n} \xrightarrow{w'}$. By coinduction, $w' \in \mathcal{L}(\mathsf{G}')$ hence $w = \alpha \cdot w' \in \mathcal{L}(\mathsf{G})$.

(3) We get $[\![S(\mathsf{G}\upharpoonright)]\!] = \{\, w \mid \mathsf{G}\upharpoonright \xrightarrow{w} \,\}$ by showing that

$$S(\mathsf{G}\upharpoonright) = \mathcal{L}(\mathsf{G})\downarrow \quad (6)$$

Indeed, from (2) we have $\mathcal{L}(\mathsf{G}) = \{\, w \mid \mathsf{G} \xrightarrow{w} \,\}$, which implies $[\![S(\mathsf{G}\upharpoonright)]\!] = [\![\mathcal{L}(\mathsf{G})\downarrow]\!]$. Prop. 6.2 ensures $S(\mathsf{G}\upharpoonright)$ is complete and correct, hence $[\![S(\mathsf{G}\upharpoonright)]\!] = \mathcal{L}(\mathsf{G})$ by Corollary 3.2 and Thm. 3.7. So, assume $\mathsf{G}\upharpoonright = \mathsf{A}_1 \triangleright P_1 \mid \ldots \mid \mathsf{A}_n \triangleright P_n$, then

$$\begin{aligned}
\mathcal{L}(\mathsf{G})\upharpoonright &= \{\, w \mid \mathsf{G} \xrightarrow{w} \,\}\downarrow && \text{by (2)}\\
&= \{\, \mathsf{A}_i \mapsto \{\, w\downarrow_{\mathsf{A}_i} \mid \mathsf{G} \xrightarrow{w} \,\} \mid 1 \le i \le n \,\} && \text{by Def. 2.4}\\
&= \{\, \mathsf{A}_i \mapsto \{\, w\downarrow_{\mathsf{A}_i} \mid w \in \mathsf{pref}(\{\, w' \in \mathcal{L}'(\mathsf{G}) \mid w \sim w' \,\}) \,\} \mid 1 \le i \le n \,\} && \text{by def. of } \mathcal{L}(\mathsf{G})\\
&= \{\, \mathsf{A}_i \mapsto \mathbb{L}(\mathsf{A}_i \triangleright P_i) \mid 1 \le i \le n \,\} && \text{by Lemma E.1}
\end{aligned}$$

And $\{\, \mathsf{A}_i \mapsto \mathbb{L}(\mathsf{A}_i \triangleright P_i) \mid 1 \le i \le n \,\} = S(\mathsf{G}\upharpoonright)$ by definition of $S(\_)$. □

F Details on Choreography Automata

We show now how the choreographic formalism of *choreography automata* [BLT20] (c-automata for short) naturally fits FCL. This essentially consists in restricting our languages to continuous closures of regular languages, so making conditions like CUI and branch-awareness decidable.

F.1 Systems of communicating machines and c- automata

Communicating finite-state machines (CFSMs) [BZ83] are adopted to model the local behaviour of systems of distributed components, whereas finite state machines with labels Σ_{int} model the global behaviours. The following definitions are borrowed from [BZ83] and adapted to our context.

Definition F.1 (System of CFSMs).
A communicating finite-state machine *(CFSM) is an FSA with set of labels* $\Sigma = \Sigma_{act}$*, where* Σ_{act} *is as defined in Def. 2.2. A CFSM is* A-local *if all its transitions have subject* A*.*

A system of CFSMs *is a map* $S = (M_{\mathsf{A}})_{\mathsf{A}\in\mathcal{P}}$ *assigning an* A*-local CFSM* M_{A} *to each participant* $\mathsf{A} \in \mathcal{P}$ *such that* $\mathcal{P} \subseteq \mathfrak{P}$ *is finite and any participant occurring in a transition of* M_{A} *is in* $\mathcal{P}$*.*

C-automata (ranged over by $\mathbb{C}\mathrm{A}$, $\mathbb{C}\mathrm{B}$, etc.) are FSAs with labels in Σ_{int}.

Definition F.2 (Choreography automata [BLT20]). *A* choreography automaton *(c-automaton for short) is an* ε*-free FSAs on the alphabet* Σ_{int}*.*

Definition F.3 ([BLT20]). *The language of interactions (resp. actions)* $\mathcal{L}(\mathbb{C}\mathrm{A})$ *(resp.* $\mathbb{L}(M)$*) associated to the c-automaton* $\mathbb{C}\mathrm{A}$ *(resp. CFSM* M*) is the language recognised by* $\mathbb{C}\mathrm{A}$ *(resp.* M*) by considering all of its states as accepting states, union the infinite words recognised by* $\mathbb{C}\mathrm{A}$ *(resp.* M*) looked at as a Büchi automaton where all states are accepting.*

By construction it follows that

Proposition F.4. *Given a c-automaton* $\mathbb{C}\mathrm{A}$*,* $\mathcal{L}(\mathbb{C}\mathrm{A})$ *is a continuous g-language.*

Proof. To show that $\mathcal{L}(\mathbb{C}\mathrm{A})$ is a g-language we need to show prefix closure. It follows from the fact that all the states are considered accepting states: if a word w is in $\mathcal{L}(\mathbb{C}\mathrm{A})$ then any prefix of w can be generated by taking the corresponding prefix of the computation generating w.

We now need to show that the language is continuous. Given that determinisation preserves the language, we can assume $\mathbb{C}\mathrm{A}$ deterministic. We need to show that $\mathcal{L}(\mathbb{C}\mathrm{A})$ contains an infinite word if it contains all its finite prefixes. Note that, thanks to determinism, words which are prefixes one of the other are generated by computations which are prefixes one of the other as well. Let w be an infinite word whose prefixes are in $\mathcal{L}(\mathbb{C}\mathrm{A})$. The infinite run obtained as the limit of the runs generating the prefixes of w generates w. This concludes the proof. □

For sake of readability, we overload the projection operator of Def. 2.4 by letting it work on c-automata.

Definition F.5 (Projection of c-automata [BLT20]). *Let* $\mathbb{C}\mathrm{A} = \langle \mathcal{S}, q_0, \Sigma_{int}, \rightarrow \rangle$ *be a c-automaton and* A *be a participant. The* projection of $\mathbb{C}\mathrm{A}$ on A*, is the CFSM* $\mathbb{C}\mathrm{A}{\downarrow_{\mathsf{A}}}$ *obtained by determinising up-to-language equivalence the* intermediate *automaton*

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

The projection of $\mathbb{C}\mathrm{A}$*, written* $\mathbb{C}\mathrm{A}{\downarrow}$*, is the communicating system* $(\mathbb{C}\mathrm{A}{\downarrow_{\mathsf{A}}})_{\mathsf{A}\in\mathcal{P}}$*.*

F.2 Semantics of CFSMs

The synchronous semantics of a system of CFSMs is the language of traces of an LTS where labels are *interactions*, that is elements of Σ_{int} (see Def. 2.1). The next definition is a slightly different (yet equivalent) version of the one in [BLT20], where the semantic of a system of CFSMs is the automaton M_S below rather than its language.

Definition F.6 (Synchronous semantics of systems of CFSMs). *Let $S = (M_{\mathsf{A}})_{\mathsf{A}\in\mathcal{P}}$ be a system of CFSMs where $M_{\mathsf{A}} = \langle \mathcal{S}_{\mathsf{A}}, q_{0\mathsf{A}}, \Sigma_{act}, \to_{\mathsf{A}}\rangle$ for each participant $\mathsf{A} \in \mathcal{P}$. A* synchronous configuration *for S is a map $s = (q_{\mathsf{A}})_{\mathsf{A}\in\mathcal{P}}$ assigning a* local state *$q_{\mathsf{A}} \in \mathcal{S}_{\mathsf{A}}$ to each $\mathsf{A} \in \mathcal{P}$.*

The synchronous semantics *of S, denoted as $(\!|S|\!)$, is the g-language $\mathcal{L}(\mathsf{M}_S)$ where $\mathsf{M}_S = \langle \mathcal{S}, s_0, \Sigma_{int}, \to\rangle$ is defined as follows:*

- *$\mathcal{S}$ is the set of configurations of S, as defined above, and $s_0 : \mathsf{A} \mapsto q_{0\mathsf{A}}$ for each $\mathsf{A} \in \mathcal{P}$ is the* initial *configuration of $\mathcal{S}$*
- *$\to$ is the set of transitions*
  - *$s_1 \xrightarrow{\mathsf{A}\to\mathsf{B}:\mathsf{m}} s_2$ such that*
    - *$s_1(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}} s_2(\mathsf{A})$ in M_{A} and $s_1(\mathsf{B}) \xrightarrow{\mathsf{A}\,\mathsf{B}?\mathsf{m}} s_2(\mathsf{B})$ in M_{B}, and*
    - *for all $\mathsf{X} \in \mathcal{P} \setminus \{\mathsf{A}, \mathsf{B}\}$, $s_1(\mathsf{X}) = s_2(\mathsf{X})$*
  - *$s_1 \xrightarrow{\varepsilon} s_2$ such that $s_1(\mathsf{A}) \xrightarrow{\varepsilon} s_2(\mathsf{A})$ in M_{A}, and for all $\mathsf{X} \in \mathcal{P} \setminus \{\mathsf{A}\}$, $s_1(\mathsf{X}) = s_2(\mathsf{X})$.*

Recall that, given a system of CFSMs $S = (M_{\mathsf{X}})_{\mathsf{X}\in\mathcal{P}}$, $\hat{S} = (\mathbb{L}(M_{\mathsf{X}}))_{\mathsf{X}\in\mathcal{P}}$ is the abstract system corresponding to S. Unsurprisingly, the semantics of S and $\hat{S}$ do coincide.

Proposition F.7. *For all systems $S = (M_{\mathsf{X}})_{\mathsf{X}\in\mathcal{P}}$ of CFSMs, $(\!|S|\!) = [\![\hat{S}]\!]$*

Proof. Let $M_{\mathsf{X}} = \langle \mathcal{S}_{\mathsf{X}}, q_{0\mathsf{X}}, \Sigma_{\text{act}}, \to_{\mathsf{X}}\rangle$ and let $(\!|S|\!) = \mathcal{L}(\langle \mathcal{S}, s_0, \Sigma_{\text{int}}, \to\rangle)$ where $\langle \mathcal{S}, s_0, \Sigma_{\text{int}}, \to\rangle$ is as in Def. F.6.

Case $(\!|S|\!) \subseteq [\![\hat{S}]\!]$ By definition of $\hat{S}$, it is enough to show that, for any $w \in (\!|S|\!)$ and for any $\mathsf{X} \in \mathcal{P}$, we have that $w{\downarrow_{\mathsf{X}}} \in \mathbb{L}(M_{\mathsf{X}})$. Let $w \in (\!|S|\!)$ and proceed by coinduction.

If $w = \varepsilon$ the thesis follows immediately. Otherwise, let $w = (\mathsf{A}\to\mathsf{B}:\mathsf{m})\cdot w'$. By definition of recognised word, there is $s_1 \in \mathcal{S}$ such that $s_0 \xrightarrow{\mathsf{A}\to\mathsf{B}:\mathsf{m}} s_1$ and w' is recognised by the automaton $\langle \mathcal{S}, s_1, \Sigma_{\text{int}}, \to\rangle$. Necessarily, by Def. F.6,

1. $s_0(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}}_{\mathsf{A}} s_1(\mathsf{A})$ and $s_1(\mathsf{B}) \xrightarrow{\mathsf{A}\,\mathsf{B}?\mathsf{m}}_{\mathsf{B}} s_2(\mathsf{B})$, and
2. for all $\mathsf{X} \neq \mathsf{A}, \mathsf{B}$, $s_1(\mathsf{X}) = s_0(\mathsf{X})$.

It follows that $w' \in (\!|S'|\!)$ where $S' = (\langle \mathcal{S}_{\mathsf{X}}, s_1(\mathsf{X}), \Sigma_{\text{act}}, \to_{\mathsf{X}}\rangle)_{\mathsf{X}\in\{\mathsf{A},\mathsf{B}\}} \cup (M_{\mathsf{X}})_{\mathsf{X}\in\mathcal{P}\setminus\{\mathsf{A},\mathsf{B}\}}$. The thesis hence follows by coinduction, since $w{\downarrow_{\mathsf{A}}} = (\mathsf{A}\,\mathsf{B}!\mathsf{m})\cdot w'{\downarrow_{\mathsf{A}}}$, $w{\downarrow_{\mathsf{B}}} = (\mathsf{A}\,\mathsf{B}?\mathsf{m})\cdot w'{\downarrow_{\mathsf{B}}}$ and, for each $\mathsf{X} \in \mathcal{P} \setminus \{\mathsf{A}, \mathsf{B}\}$, $w{\downarrow_{\mathsf{X}}} = w'{\downarrow_{\mathsf{X}}}$.

Case $[\![\hat{S}]\!] \subseteq (\!|S|\!)$ Let $w \in [\![\hat{S}]\!]$ and proceed by coinduction. If $w = \varepsilon$ the thesis follows immediately. Otherwise, let $w = (\mathsf{A}\to\mathsf{B}:\mathsf{m})w'$. Since $w{\downarrow_{\mathsf{A}}} = (\mathsf{A}\,\mathsf{B}!\mathsf{m})w'{\downarrow_{\mathsf{A}}}$, $w{\downarrow_{\mathsf{B}}} = (\mathsf{A}\,\mathsf{B}?\mathsf{m})\cdot w'{\downarrow_{\mathsf{B}}}$ and, for each $\mathsf{X} \in \mathcal{P} \setminus \{\mathsf{A}, \mathsf{B}\}$, $w{\downarrow_{\mathsf{X}}} = w'{\downarrow_{\mathsf{X}}}$, and since, for each $\mathsf{X} \in \mathcal{P}$, we have $w{\downarrow_{\mathsf{X}}} \in \mathbb{L}(M_{\mathsf{X}})$, it follows, by definition of recognised word and by Def. 2.6, that $w' \in [\![\hat{S}']\!] = (\mathbb{L}(\langle \mathcal{S}_{\mathsf{X}}, s_1(\mathsf{X}), \mathcal{L}_{\text{act}}, \to_{\mathsf{X}}\rangle))_{\mathsf{X}\in\{\mathsf{A},\mathsf{B}\}} \cup (\mathbb{L}(M_{\mathsf{X}}))_{\mathsf{X}\in\mathcal{P}\setminus\{\mathsf{A},\mathsf{B}\}}$, where

1. $s_0(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}}_{\mathsf{A}} s_1(\mathsf{A})$ and $s_1(\mathsf{B}) \xrightarrow{\mathsf{A}\,\mathsf{B}?\mathsf{m}}_{\mathsf{B}} s_2(\mathsf{B})$, and
2. for all $\mathsf{X} \neq \mathsf{A}, \mathsf{B}$, $s_1(\mathsf{X}) = s_0(\mathsf{X})$.

The thesis hence follows by coinduction. □

By construction and [BLT20, Prop. 3.5] the following can be inferred.

Proposition F.8. *If* $\mathbb{C}\mathrm{A}$ *is a c-automaton on* $\mathcal{P}$ *then for all* $\mathsf{A} \in \mathcal{P}$, $\mathcal{L}(\mathbb{C}\mathrm{A})\!\downarrow_{\mathsf{A}} = \mathbb{L}(\mathbb{C}\mathrm{A}\!\downarrow_{\mathsf{A}})$.

Proof. By definition of projection and since determinisation preserves the language $\mathbb{L}(\mathbb{C}\mathrm{A}\!\downarrow_{\mathsf{A}}) = \mathbb{L}(A_{\mathsf{A}})$ where A_{A} is the intermediate automaton (cfr. Def. F.5). Since A_{A} is obtained by taking the projection of every transition in $\mathbb{C}\mathrm{A}$, $\mathbb{L}(A_{\mathsf{A}}) = \mathcal{L}(\mathbb{C}\mathrm{A})\!\downarrow_{\mathsf{A}}$. The thesis follows by transitivity. □

F.3 Properties of systems of CFSMs

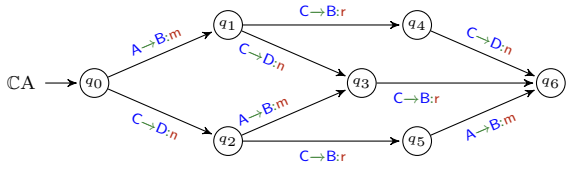
The next definition recalls the standard properties of systems of CFSMs from [BLT20].

Definition F.9 (Communication properties of systems of CFSMs [BLT20]). *Let* $S = (M_{\mathsf{A}})_{\mathsf{A}\in\mathcal{P}}$ *be a system of CFSMs and let* M_S *be as defined in Def. F.6*

- **Liveness***: S is* live *if for each configuration $s \in \mathsf{RC}(\!|S|\!)$ and each $\mathsf{A} \in \mathcal{P}$ such that $s(\mathsf{A})$ has some outgoing transition in M_{A}, there exists a run of $\!|S|\!$ from s including a transition involving A.*
- **Lock freedom***: A configuration $s \in \mathsf{RC}(\mathsf{M}_S)$ is a* lock *if there is $\mathsf{A} \in \mathcal{P}$ with an outgoing transition t from $s(\mathsf{A})$ in M_{A} and there exists a run of M_S starting from s maximal w.r.t. prefix order containing no transition t' involving A. System S is* lock-free *if for each $s \in \mathsf{RC}(\mathsf{M}_S)$, s is not a lock.*
- **Deadlock freedom***: A configuration $s \in \mathsf{RC}(\mathsf{M}_S)$ is a* deadlock *if s has no outgoing transitions in M_S, yet there exists $\mathsf{A} \in \mathcal{P}$ such that $s(\mathsf{A})$ has an outgoing transition in M_{A}. System S is* deadlock-free *if for each $s \in \mathsf{RC}(\mathsf{M}_S)$, s is not a deadlock.*

In [BLT20] a notion of well-formedness was provided in order to guarantee the above properties (besides what we call *correctness* in the present paper). We found out later on that such a notion is flawed. In fact it does not correctly handle the interplay between concurrent transitions and choices. This is shown by the following example.

Example F.10. The c-automaton $\mathbb{C}\mathrm{A}$ below is well-formed according to [BLT20, Def. 4.12].



However, the system $\mathbb{C}\mathrm{A}\!\downarrow$ admits the run $\mathsf{C}{\to}\mathsf{B}{:}\mathsf{r}\ \ \mathsf{C}{\to}\mathsf{D}{:}\mathsf{n}$ which is not a word accepted by $\mathbb{C}\mathrm{A}$. ◇

In our setting, the problem of the above $\mathbb{CA}$ is that $\mathcal{L}(\mathbb{CA})$ is not CUI. In fact, by setting $w_1 = \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$ and $w_2 = \mathsf{C}{\to}\mathsf{D}{:}\mathsf{n}$, we have that $w_1{\cdot}\mathsf{C}{\to}\mathsf{B}{:}\mathsf{r}, w_2{\cdot}\mathsf{C}{\to}\mathsf{B}{:}\mathsf{r} \in \mathcal{L}(\mathbb{CA})$. Now, by taking $w = \varepsilon$, we have that $w_1{\downarrow_\mathsf{C}} = w{\downarrow_\mathsf{C}}$ and $w_2{\downarrow_\mathsf{B}} = w{\downarrow_\mathsf{B}}$, but $w{\cdot}\mathsf{C}{\to}\mathsf{B}{:}\mathsf{r} = \mathsf{C}{\to}\mathsf{B}{:}\mathsf{r} \notin \mathcal{L}(\mathbb{CA})$.

By restricting to c-languages associated to c-automata we get that CUI and BA are decidable properties (see Subsect. F.4 below). The approach of the present paper can hence be used to prove the communication properties considered in [BLT20] (namely those in Def. F.9) by showing their corresponding ones in the FCL setting.

We proceed now to show, as expected, that lock-, strong lock- and deadlock freedom for the communicating system corresponding to a system of CFSMs do imply to the analogous properties also for the latter.

Lemma F.11. *Let $S = (M_\mathsf{X})_{\mathsf{X}\in\mathcal{P}}$ be a system of CFSMs and let M_S be as in Def. F.6. If $w \in \mathcal{L}(\mathsf{M}_S)$ is recognised by a run from s_0 to s then, for each $\mathsf{A} \in \mathcal{P}$, $w{\downarrow_\mathsf{A}} \in \mathbb{L}(M_\mathsf{A})$ and $w{\downarrow_\mathsf{A}}$ is recognised by a run from $s_0(\mathsf{A})$ to $s(\mathsf{A})$.*

Proof. By induction on the length of w using Def. F.6. □

Proposition F.12. *For all systems of CFSMs $S = (M_\mathsf{X})_{\mathsf{X}\in\mathcal{P}}$*

(i) $[\![\hat{S}]\!]$ lock-free implies S live;
(ii) $[\![\hat{S}]\!]$ strong lock-free implies S lock-free;
(iii) $[\![\hat{S}]\!]$ deadlock-free implies S deadlock-free;

Proof. (i) Let $[\![\hat{S}]\!]$ be lock-free. Following Def. F.9, in order to show the liveness of S, let us consider $s \in \mathsf{RC}(\mathsf{M}_S)$, and let $s(\mathsf{A}) \xrightarrow{l} s(\mathsf{A})$ be a transition in M_A (for a certain $\mathsf{A} \in \mathcal{P}$). Let now w be the trace of a run of M_S from s_0 to s. By Prop. F.7 $w \in [\![\hat{S}]\!]$ and, by Lemma F.11, $w{\downarrow_\mathsf{A}} \in \mathbb{L}(M_\mathsf{A})$ and $w{\downarrow_\mathsf{A}}$ is the trace of a run from $s_0(\mathsf{A})$ to $s(\mathsf{A})$ in M_A. Now, to prove that S is live, we have to show that there exists a run of M_S from s including a transition which has a transition from $s(\mathsf{A})$ as component transition. We have that $w{\downarrow_\mathsf{A}}\, l \in \mathbb{L}(M_\mathsf{A})$. Hence, by lock-freedom of $[\![\hat{S}]\!]$, there is w' such that $w{\cdot}w' \in [\![\hat{S}]\!]$ and $w'{\downarrow_\mathsf{A}} \neq \varepsilon$. By Prop. F.7 w' is also a run of M_S from s and, by Lemma F.11 (considering the CFSMs $S' = (M'_\mathsf{X})_{\mathsf{X}\in\mathcal{P}}$ where each M'_X is like M'_X but with $s(\mathsf{X})$ as initial state), such a run has a transition from $s(\mathsf{A})$ as component transition.

The proof of (ii) and (iii) are similar to the one (i). □

F.4 Deciding CUI and branch-awareness for c-automata

Theorem F.13. *CUI is decidable.*

Proof. By definition of CUI we have $\neg\mathsf{cui}(L(\mathbb{CA})) \iff \exists\alpha = \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}.\, \exists w_1, w_2, w \in \mathcal{L}.\ w_1, w_2, w$ are finite, $w_1{\cdot}\alpha, w_2{\cdot}\alpha \in \mathcal{L}$, $w{\downarrow_\mathsf{A}} = w_1{\downarrow_\mathsf{A}}$, $w{\downarrow_\mathsf{B}} = w_1{\downarrow_\mathsf{B}}$ and $w{\cdot}\alpha \notin \mathcal{L}$. By definition of projection the condition $w{\downarrow_\mathsf{A}} = w_1{\downarrow_\mathsf{A}}$ means that the states reached with computations with labels w and w_1 are in the same equivalence class in the minimisation of $\mathbb{CA}{\downarrow_\mathsf{A}}$, and similarly for $w{\downarrow_\mathsf{B}} = w_1{\downarrow_\mathsf{B}}$. Thus we need to look for states $s_\mathsf{A}, s_\mathsf{B}, s$ in $\mathbb{CA}$ such that:

– s_{A} and s are in the same equivalence class in the minimisation of $\mathbb{C}\mathrm{A}{\downarrow_{\mathsf{A}}}$;
– s_{B} and s are in the same equivalence class in the minimisation of $\mathbb{C}\mathrm{A}{\downarrow_{\mathsf{B}}}$;
– $s_{\mathsf{A}}, s_{\mathsf{B}}$ have an outgoing transition $\mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$, while s has not.

Since all the involved automata are finite the check can be done within a finite time. □

Theorem F.14. *Branch-awareness is decidable.*

Proof. Fix a c-automaton $\mathbb{C}\mathrm{A}$ and let A_{X} be the intermediate automaton obtained by projecting $\mathbb{C}\mathrm{A}$ on one of its participants X. Also, given a state p of $\mathbb{C}\mathrm{A}$, we define A_p as the automaton that coincides with A but for the initial state, which is now p; likewise, A^p is the automaton that coincides with A where only p is accepting.

In order to get a decision procedure for $\mathsf{ba}(\mathcal{L}(\mathbb{C}\mathrm{A}))$, we observe that the following equivalences do hold.

$\neg\mathsf{ba}(\mathcal{L}(\mathbb{C}\mathrm{A}))$
$\iff \exists \mathsf{X} \in \mathcal{P} : \exists w_1, w_2 \in \mathcal{L}(\mathbb{C}\mathrm{A}) :$ w_1, w_2 are maximal and $w_1 \neq w_2$ and $w_1{\downarrow_{\mathsf{X}}} \prec w_2{\downarrow_{\mathsf{X}}}$
$\iff \exists \mathsf{X} \in \mathcal{P} : \exists p, q$ states of $\mathbb{C}\mathrm{A} :$ $p \neq q$ and
a) $\mathbb{L}(A^p_{\mathsf{X}}) \cap \mathbb{L}(A^q_{\mathsf{X}}) \neq \emptyset$;
b) p and q are such that there is a maximal $w'_1 \in \mathcal{L}(\mathbb{C}\mathrm{A}_p)$ such that $w'_1{\downarrow_{\mathsf{X}}} = \varepsilon$ and a finite $w'_2 \in \mathcal{L}(\mathbb{C}\mathrm{A}_q)$ such that $w'_2{\downarrow_{\mathsf{X}}} \neq \varepsilon$

Intuitively, the last equivalence above states that for a participant X it is not possible to distinguish when a computation halts with word w_1 or continues as w_2. Transferring this condition on the automata yields the last equivalence, which basically identifies that $\mathbb{L}(A^p_{\mathsf{X}}) \cap \mathbb{L}(A^q_{\mathsf{X}}) \neq \emptyset$, since it contains at least $w_1{\downarrow_{\mathsf{X}}}$. Therefore, there are two states p and q in the intermediate automaton A_{X} that accept a common prefix of the projections $w_1{\downarrow_{\mathsf{X}}}$ and $w_2{\downarrow_{\mathsf{X}}}$, and such that there is a maximal path from p made only of ε-transitions ($w'_1{\downarrow_{\mathsf{X}}}$) while some path from q involves X ($w'_2{\downarrow_{\mathsf{X}}}$).

We now notice that (a) is decidable because the intersection of ω-regular languages is computatable. In order to decide (b), instead, we can proceed as follows. In order to check the existence of a maximal $w'_1 \in \mathcal{L}(\mathbb{C}\mathrm{A}_p)$ such that $w'_1{\downarrow_{\mathsf{X}}} = \varepsilon$, it is enough to follow all possible paths from p in A^p_{X} containing only ε-transitions using breadth-first search. Finiteness of states guarantees that we either end up to a state without outgoing transitions, or to a state already visited, or we find out that no maximal path out of p does contain only ε-transitions. The same procedure enables to check if there exists a $w'_2 \in \mathcal{L}(\mathbb{C}\mathrm{A}_q)$ such that $w'_2{\downarrow_{\mathsf{X}}} \neq \varepsilon$.

□

References for the Appendix

BLT20. Franco Barbanera, Ivan Lanese, and Emilio Tuosto. Choreography automata. In Simon Bliudze and Laura Bocchi, editors, *Coordination Models and Languages - 22nd IFIP WG 6.1 International Conference, COORDINATION 2020, Held as Part of the 15th International Federated Conference on Distributed Computing Techniques, DisCoTec 2020, Valletta, Malta, June 15-19, 2020, Proceedings*, volume 12134 of *Lecture Notes in Computer Science*, pages 86–106. Springer, 2020.

BZ83. Daniel Brand and Pitro Zafiropulo. On communicating finite-state machines. *J. ACM*, 30(2):323–342, 1983.

Gas90. Paul Gastin. Infinite traces. In Irène Guessarian, editor, *Semantics of Systems of Concurrent Processes, LITP Spring School on Theoretical Computer Science, La Roche Posay, France, April 23-27, 1990, Proceedings*, volume 469 of *Lecture Notes in Computer Science*, pages 277–308. Springer, 1990.

KS17. Dexter Kozen and Alexandra Silva. Practical Coinduction. *Mathematical Structures in Computer Science*, 27(7):1132–1152, 2017.