Compliance Preorders for Web Services

Michele Bugliesi, Damiano Macedonio, Luca Pino, and Sabina Rossi

Dipartimento di Informatica, Università Ca' Foscari Venezia
{michele,mace,lpino,srossi}@dsi.unive.it

Abstract. Compliance is a basic property of web-service architectures that ensures the absence of deadlocks and livelocks during execution. Following recent attempts in the literature, we interpret compliance as an experiment, much like the experiments made by a test process in testing theories, and use it as the basis for a notion of *compliance preserving substitution* of components within a composition of web services.
We review the different notions of compliance in the literature, analyze their relative strengths and weaknesses, and formalize their interrelationships by providing a uniform formal framework where we reconcile the different perspectives that characterize them.

1 Introduction

Compliance is a basic property that characterizes the correct behavior of concurrent distributed systems. It is used widely in the context of Service Oriented Architectures (SOA) as a formal device to identify well-formed service compositions, those whose interactions are free of synchronization errors.

Formal theories of compliance have been developed within different settings, most notably with *session types* and *behavioral contracts*. Session types [11] have originally been conceived as a generalization of channel types [16] for the static control of interaction patterns in which the same channel is used to send and/or receive payloads of different types at different times. Recently, systems of session types have applied widely in the analysis of various kinds of interaction and conversation structures [6, 12, 5].

Behavioral contracts, our focus in the present paper, arise in process algebraic settings, and provide abstract descriptions of system behavior by means of terms of some process algebra. Formal theories of contracts have first been introduced in [7], and then further developed along independent lines of research in [13, 8, 9], and in [3, 4].

All these papers share the main motivations and the overall technical setup, inspired by the theory of testing in process algebra [15]. In particular, they all interpret compliance as a basic test for investigating services, extract the preorder relationships induced by the test, and justify a compliance-preserving substitution principle for services based on that. On the other hand, the approaches differ significantly in the notion of compliance adopted as well as in the settings where they apply it. In [13, 8, 9], compliance is targeted at preventing deadlocks, and the theory is developed for a client-server setting to provide safety guarantees for

the client. Compliant servers are those which will never get their clients stuck: the compliance preorder is established similarly in terms of the ability of servers to satisfy their clients. In [3, 4], instead, compliance is a stronger condition that ensures the absence of deadlocks *and* livelocks, and the application setting is that of choreographies: compliant choreographies are those whose computations never get stuck or trapped into infinite loops without chances to exit. The compliance pre-order, in turn, is induced on the component contracts, in terms of their ability to preserve the compliance of the choreographies they are part of.

In this paper, we review the existing definitions in the literature to formalize their inter-relationships. As a result of our analysis we provide a uniform framework where we (*i*) reconcile the different perspectives that characterize the existing definitions of compliance, (*ii*) fill some of the existing gaps among them, and thus (*iii*) maximize the potential of cross-fertilization among the different approaches. We start with an analysis of the deadlock-safe notion of compliance developed in [8, 13] for client-server settings, and propose an equivalent formulation that scales naturally to multi-party service compositions. Then, we give a fully-abstract co-inductive characterization of the induced compliance-preorder and discuss its use for the safe replacement of services inside multi-party compositions. We also analyze the stronger definition of compliance proposed by [4] for choreographies, showing that it effectively constitutes a conservative generalization of the deadlock-safe definition of [8, 13] (when the latter is lifted to multiparty compositions). We generalize the coinductive construction of the deadlock-safe preorder to obtain a sound (but not complete) characterization of the stronger preorder. For both pre-orders, we also show how the *filters* from [8] may be employed to achieve a flexible compliance-preserving substitution principle inside choreographies.

Plan. Section 2 introduces the contract language we use for our analysis. Section 3 analyzes the notions of compliance in the literature, and introduces our own variations of such notions. Section 4 develops the coinductive characterizations of the associated compliance preorders. Section 5 discusses generalized versions of the preorders, and their counductive characterizations. Section 6 concludes the presentation.

2 A core contract language

We start introducing a small language for contracts and contract compositions that we use for our analysis. Contracts are represented as (single-threaded) terms of a CCS-like [14] process calculus that includes recursion and operators for external and internal choice. Parallel composition arises in *contract compositions*. We presuppose a denumerable set of action names $\mathcal{A}$, ranged over by a, b, c. Actions represent the basic units of observable behavior of the underlying services, namely input, noted a, and output, note $\overline{a}$. We let α range over actions and co-actions, and note $\overline{\alpha}$ the co-action corresponding to α.

$$\begin{array}{ll} \textit{Contracts} & \sigma ::= \mathbf{1} \mid \mathbf{x} \mid \alpha.\sigma \mid \sigma + \sigma \mid \sigma \oplus \sigma \mid \mathtt{rec}(\mathbf{x})\,\sigma \\ \textit{Compositions} & C ::= \sigma \mid C \parallel C \end{array}$$

1 signals that the service has reached a successful state, $\alpha;\sigma$ describes a service that performs action α and then behaves as σ; $\sigma + \sigma'$ denotes an external choice, guided by the environment, while $\sigma \oplus \sigma'$ indicates a local choice between σ and σ' made irrespective of the structure of the interacting environment; $\mathtt{rec}(\mathbf{x})\,\sigma$ is a recursively defined contract. We assume a standard contractivity condition for the recursion operator, requiring that recursion variables be guarded by a prefix. We let $in(\sigma)$ and $out(\sigma)$ note the set of input and output actions in σ, respectively.

Contract Satisfaction: $\sigma\checkmark$

$$\mathbf{1}\checkmark \qquad \frac{\sigma_i\checkmark}{\sigma_1+\sigma_2\checkmark} \qquad \frac{\sigma\{\mathbf{x} := \mathtt{rec}(\mathbf{x})\,\sigma\}\checkmark}{\mathtt{rec}(\mathbf{x})\,\sigma\checkmark}$$

Contract transitions: $\sigma \xrightarrow{\dot\alpha} \sigma'$

$$\alpha.\sigma \xrightarrow{\alpha} \sigma \qquad \sigma_1 \oplus \sigma_2 \longrightarrow \sigma_i \quad (i=1,2)$$

$$\frac{\sigma_i \xrightarrow{\dot\alpha} \sigma}{\sigma_1+\sigma_2 \xrightarrow{\dot\alpha} \sigma}\,(i=1,2) \qquad \frac{\sigma\{\mathbf{x} := \mathtt{rec}(\mathbf{x})\,\sigma\} \xrightarrow{\dot\alpha} \sigma'}{\mathtt{rec}(\mathbf{x})\,\sigma \xrightarrow{\dot\alpha} \sigma'}$$

Composition satisfaction and transitions:

$$\frac{C_1\checkmark \quad C_2\checkmark}{C_1 \parallel C_2\checkmark} \qquad \frac{C_1 \xrightarrow{\alpha} C_1' \quad C_2 \xrightarrow{\overline{\alpha}} C_2'}{C_1 \parallel C_2 \longrightarrow C_1' \parallel C_2'} \qquad \frac{C_1 \xrightarrow{\dot\alpha} C_1'}{C_1 \parallel C_2 \xrightarrow{\dot\alpha} C_1' \parallel C_2}$$

Table 1. Dynamics of contracts and compositions

Dynamics of contracts and compositions. We define the dynamics of the calculus with a labelled transition system (and a success predicate), with rules reported in Table 1: $\dot\alpha$ indicates the label α or no label. The first block of rules defines the successful states of a contract, i.e., those that expose the success term **1** at top level, or immediately under an external choice (up-to recursive unfoldings). The rules in the second and third blocks define the transitions for contracts and compositions, and are mostly self-explanatory. Notice that a composition reaches a successful state only when all component contracts are themselves successful.

We write $\Longrightarrow$ to note the reflexive and transitive closure of $\longrightarrow$, and $\sigma \stackrel{\alpha}{\Longrightarrow} \sigma'$ for $\sigma \Longrightarrow \stackrel{\alpha}{\longrightarrow} \Longrightarrow \sigma'$. Similarly, for $w = \alpha_1 \dots \alpha_n$ the transition $\sigma \stackrel{w}{\Longrightarrow} \sigma'$ stands for $\sigma \stackrel{\alpha_1}{\Longrightarrow} \cdots \stackrel{\alpha_n}{\Longrightarrow} \sigma'$. We omit the target of a (weak) transition when immaterial, writing $\sigma \stackrel{\alpha}{\longrightarrow}$ and $\sigma \stackrel{\alpha}{\Longrightarrow}$ to signal that σ has a (weak) transition on α to some σ', and $\mathit{init}(\sigma)$ for the set $\{\alpha \mid \sigma \stackrel{\alpha}{\Longrightarrow}\}$. Finally, with $\sigma \downarrow \mathrm{R}$ we note that $\mathrm{R} \subseteq (\mathcal{A} \cup \{\checkmark\})$ is the smallest non-empty set such that $\sigma \stackrel{\alpha}{\longrightarrow}$ implies $\alpha \in \mathrm{R}$, and $\sigma\checkmark$ implies $\checkmark \in \mathrm{R}$. Similarly, $\sigma \Downarrow \mathrm{R}$ whenever $\sigma \Longrightarrow \sigma'$ with $\sigma' \downarrow \mathrm{R}$.

A computation for a composition C is a sequence $C \equiv C_0 \longrightarrow C_1 \longrightarrow \dots$; the computation is *maximal* if either it is infinite or there exists C_n such that $C \Longrightarrow C_n \not\longrightarrow$. A composition C is finite if all its maximal computations are finite. Throughout, we presuppose the following conditions on contracts and compositions.

Definition 1 (Determinacy). *A contract σ is* determinate *if for every action α there exists at most one contract σ' such that $\sigma \Longrightarrow \stackrel{\alpha}{\longrightarrow} \sigma'$, and σ' is itself determinate. A composition $\sigma_1 \parallel \dots \parallel \sigma_n$ is determinate, if so are the σ_i, and in addition, $\mathit{in}(\sigma_i) \cap \mathit{in}(\sigma_j) = \emptyset$ whenever $i \neq j$.*

When $\sigma \stackrel{\alpha}{\Longrightarrow}$, we note $\sigma(\alpha)$ the unique σ' such that $\sigma \Longrightarrow \stackrel{\alpha}{\longrightarrow} \sigma'$. ($\sigma(\alpha)$ is always defined for determinate contracts). Determinacy is technically convenient for our analysis, and represents a fairly mild assumption (cf. [8] for a similar assumption, which in that case is enforced directly by the transition relation). Indeed, contracts can be made determinate, by factoring, without affecting their external behavior. To illustrate, the non-determinate contract $(a.\sigma_1 \oplus b.\sigma_2) + (a.\sigma_3 \oplus b.\sigma_4)$ may equivalently be expressed (up to, internal moves) as the determinate contract $a.(\sigma_1 \oplus \sigma_3) \oplus b.(\sigma_2 \oplus \sigma_4)$. As to compositions, determinacy, simply amounts to interpreting the actions of each contract in a composition as being associated with services located and accessible at univocally identified ports/sites (as done, for instance, in [4]).

3 Compliance tests

We start with a review of the asymmetric, deadlock-safe notion of compliance (ds-compliance, for short), as proposed by [8, 13] for client-server settings, and introduce its asymmetric and symmetric variants for general, multi-party compositions. Then, we discuss the definition of *safe* compliance, that is sensitive to both deadlocks and livelocks.

3.1 Deadlock-safe compliance

For two contracts ρ and σ, let $\rho \bowtie \sigma$ signal that ρ and σ may synchronize, possibly after a sequence of internal moves. Formally $\rho \bowtie \sigma$ iff $\mathit{init}(\sigma) \cap \overline{\mathit{init}(\rho)} \neq \emptyset$, where $\overline{\mathit{init}(\rho)}$ is the set of co-actions corresponding to the actions in $\mathit{init}(\rho)$. The asymmetric presentation distinguishes two roles (client and server, respectively) for the contracts involved in the compliance test.

Definition 2 (Client-server ds-compliance [8, 13]). *A client ρ and a server σ are* ds-compliant*, written $\rho \dashv^{\mathtt{ds}} \sigma$ iff whenever $\rho \parallel \sigma \Longrightarrow \rho' \parallel \sigma'$, either $\rho' \bowtie \sigma'$, or $\checkmark \in \mathrm{R}$ for all R such that $\rho' \Downarrow \mathrm{R}$.*

Accordingly, $\rho \dashv^{\mathtt{ds}} \sigma$ if and only if whenever the interaction between ρ and σ gets stuck (as there is no chance of synchronization) ρ may independently terminate with success. The asymmetric nature of the definition, and its clear bias in favor of ρ, is a consequence of the different intended roles of the two contracts in the composition. In [13], the definition has the following, additional proviso: in case $\sigma'\uparrow$ then for all R such that $\rho' \Downarrow \mathrm{R}$, $\mathrm{R} = \{\checkmark\}$. The intuition here is that if the interaction between client and server gets the server trapped into an internal loop (noted $\sigma'\uparrow$), then the client must terminate successfully without expecting any further synchronization with the server. The proviso holds vacuously in our contract language, given that recursive contracts are formally contractive, hence they may not loop without interaction.

Definition 2 may be restated equivalently by stipulating that ρ and σ are compliant if whenever $\rho \parallel \sigma \Longrightarrow \rho' \parallel \sigma' \not\longrightarrow$, one has $\rho' \checkmark$. That the two definitions are equivalent follows again from our formal-contractiveness assumption, which implies that $\rho \not\bowtie \sigma$ iff $\rho \parallel \sigma \Longrightarrow \rho' \parallel \sigma' \not\longrightarrow$. The new formulation is interesting as it suggests the following, natural lifting of the client-server notion of compliance to multi-party compositions.

Definition 3 (Asymmetric ds-compliance). *A contract ρ is* ds-compliant *with a composition C, written $\rho \dashv^{\mathtt{ds}} C$, iff whenever $\rho \parallel C \Longrightarrow \rho' \parallel C' \not\longrightarrow$, one has $\rho' \checkmark$.*

Asymmetric compliance, in turn, is readily made symmetric by simply removing the bias in favour of any of the component services.

Definition 4 (Symmetric ds-compliance). *A contract composition C is* ds-compliant*, noted $C\!\searrow^{\mathtt{ds}}$, if whenever $C \Longrightarrow C' \not\longrightarrow$, one has $C' \checkmark$.*

3.2 Safe compliance

One weakness of ds-compliance is that it is insensitive to livelocks. The following example helps illustrate the problem in the asymmetric setting. Consider the two contracts $\sigma = \mathtt{rec}(\mathbf{x})\,(a.\mathbf{x} \oplus b.\mathbf{x})$ and $\sigma' = \mathtt{rec}(\mathbf{x})\,a.\mathbf{x}$, and take the contract $\rho = \mathtt{rec}(\mathbf{x})\,(\overline{a}.\mathbf{x} + \overline{b}.\mathbf{1})$. Applying Definition 3, it is a routine check to verify that ρ is ds-compliant with both σ and σ', namely $\rho \dashv^{\mathtt{ds}} \sigma$, and $\rho \dashv^{\mathtt{ds}} \sigma'$. This is not exactly desirable, as the two contracts determine quite different behaviors for ρ: indeed, while σ is acceptable to ρ, as there is always a chance for ρ to reach a successful state, σ' is not as it leaves ρ trapped into a livelock.

If livelocks are to be avoided, we need to strengthen the compliance test. The following definition, that we take verbatim from [4], does the job with a test inspired by the theory of *should*-testing [17].

Definition 5 (Symmetric safe compliance [4]). *A contract composition C is* $\mathtt{s}$-compliant*, noted $C\searrow^{\mathtt{s}}$, if for every C' such that $C \Longrightarrow C'$ there exists C'' such that $C' \Longrightarrow C''\checkmark$.*

In other words, $\mathtt{s}$-compliance ensures that at each intermediate step of the computation in a choreography, each component service has a way to reach a successful state (either autonomously, or via synchronizations within the choreography). This is enough to avoid livelocks. To illustrate, consider the composition $C \stackrel{def}{=} \mathtt{rec}(\mathbf{x})\,(a.\mathbf{x}) \parallel \mathtt{rec}(\mathbf{x})\,(\bar{a}.\mathbf{x})$. C is $\mathtt{ds}$-compliant even though it never reaches any successful state: indeed, the condition imposed by Definition 4 holds vacuously as C has only infinite computation. This is rectified in Definition 5 by demanding that all intermediate computation states offer a path to success for all the services of the composition.

Safe compliance is readily re-cast into an asymmetric presentation, by simply re-introducing the bias in favour of one component and stipulating that $\rho \dashv^{\mathtt{s}} C$ if for every $\rho' \parallel C'$ such that $\rho \parallel C \Longrightarrow \rho' \parallel C'$ there exists $\rho'' \parallel C''$ such that $\rho' \parallel C' \Longrightarrow \rho'' \parallel C''$ and $\rho''\checkmark$. The new predicate conveys the desired guarantees for the client-server setting (when C is a single server): if we go back to our problematic servers σ and σ' above, we now see that $\rho \dashv^{\mathtt{s}} \sigma$ but $\rho \not\dashv^{\mathtt{s}} \sigma'$, as desired.

3.3 Symmetric vs Asymmetric Compliance

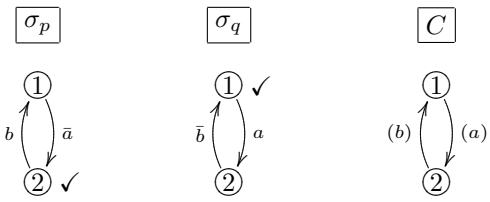
While there is clearly a strong connection between the two presentations of compliance (symmetric vs asymmetric), we are not aware of results establishing formal relationships. We give two such results below, first showing that for deadlock-safe compliance the asymmetric presentation can be defined in terms of the symmetric one, in the following sense. Given $C = \sigma_1 \parallel \ldots \parallel \sigma_n$, we note $C/_i$ the composition "C drop σ_i", defined as follows: $C/_i \stackrel{def}{=} \sigma_1 \parallel \ldots \parallel \sigma_{i-1} \parallel \sigma_{i+1} \parallel \ldots \parallel \sigma_n$.

Theorem 6 (Symmetric vs Asymmetric $\mathtt{ds}$-compliance). *Let C be the composition $\sigma_1 \parallel \ldots \parallel \sigma_n$. Then $C\searrow^{\mathtt{ds}}$ if and only if $\sigma_i \dashv^{\mathtt{ds}} C/_i$ for all $1 \leq i \leq n$.*

Proof. ($\Longrightarrow$) From the hypothesis, for every C' such that $C \Longrightarrow C' \not\longrightarrow$ we have $C'\checkmark$. This means that each component of C' is in a success state. Since this is true of all $C' \not\longrightarrow$ reachable from C, it must be the case that $\sigma_i \dashv^{\mathtt{ds}} C/_i$ for all i. ($\Longleftarrow$) The hypothesis is that for every i, $\sigma_i \dashv^{\mathtt{ds}} C/_i$. Now, for all the C' such that $C \Longrightarrow C' \not\longrightarrow$ we have that $\sigma_i'\checkmark$ (where σ_i' is the state corresponding to σ_i in C'). Since this is true of every i, it follows that $C\searrow^{\mathtt{ds}}$. □

In the forward direction Theorem 6 carries over to the case of safe compliance. Instead, somewhat surprisingly, this is not true of the backward direction. To see that, consider the following counter-example. Take $\sigma_p = \mathtt{rec}(\mathbf{x})\,(\bar{a}.(b.\mathbf{x} + \mathbf{1}))$ and $\sigma_q = \mathtt{rec}(\mathbf{x})\,(a.\bar{b}.\mathbf{x} + \mathbf{1})$, and form the composition $C = \sigma_p \parallel \sigma_q$. The following diagrams show the transitions for each contract and for the composition:

the success states are marked by the satisfaction predicate.



Now, $\sigma_p \dashv^{\mathtt{s}} \sigma_q$ because, for every state reached by the computation of C, we can extend the computation to reach the state 2, where the σ_p reaches its success. $\sigma_q \dashv^{\mathtt{s}} \sigma_p$ holds for the same reason, because it's always possible to reach the state 1 where σ_q reaches its success. However $C \not\searrow^{\mathtt{s}}$ because, as outlined by the graph above, there is not a reachable state where both the contracts are successful.

4 Compliance preorders

Associated with a compliance test, which we mark as $\bullet$ to generalize[1], one defines a corresponding semantics for service contracts. In client-server setting, this is stated in terms of the sets of the compliant clients $[\sigma]^\bullet \stackrel{def}{=} \{\rho \mid \rho \dashv^\bullet \sigma\}$; in multi-party compositions, it is defined similarly in terms of sets of compliant compositions: $[\sigma]^\bullet \stackrel{def}{=} \{C \mid \sigma \parallel C \searrow^\bullet\}$. Then, based on the contract semantics, one defines the contract preorder, uniformly as follows: $\sigma \sqsubseteq^\bullet \sigma' \stackrel{def}{=} [\sigma]^\bullet \subseteq [\sigma']^\bullet$. Defined this way, compliance preorders may be employed to justify a substitution principle for servers, and more generally for services inside choreographies, namely: given two contracts σ and σ', it is safe to substitute (a service described by) σ with (a service described by) σ' as long as $\sigma \sqsubseteq^\bullet \sigma'$.

In this section, we give coinductive versions of the two preorders $\sqsubseteq^{\mathtt{ds}}$ and $\sqsubseteq^{\mathtt{s}}$ induced by the compliance tests introduced in the previous section. As we will show, for the deadlock-safe case, the coinductive version provides a fully abstract characterization of $\sqsubseteq^{\mathtt{ds}}$. For the safe preorder, the characterization is only sound, not complete. In both cases, being our compositions finite, and our contracts finite-state, the characterizations provide an effective construction for deciding the preorders.

4.1 Characterizing the deadlock-safe preorder

We start with the deadlock safe preorder. The definition we give here, and the full-abstraction proof refine those in [13, 9] to account for the symmetric nature of the compliance test.

Definition 7 (Coinductive ds-preorder). *$\mathcal{R}$ is a coinductive* ds*-preorder if $\sigma \mathcal{R} \rho$ implies that: (i) if $\rho \longrightarrow \rho'$, then $\sigma \mathcal{R} \rho'$; (ii) if $\rho \downarrow \mathrm{R}$, then there exists $\mathrm{S} \subseteq \mathrm{R}$ such that $\sigma \Downarrow \mathrm{S}$; (iii) for every action α if $\rho \stackrel{\alpha}{\longrightarrow} \rho'$, then $\sigma \stackrel{\alpha}{\Longrightarrow}$ and $\sigma(\alpha) \mathcal{R} \rho'$. We note $\preceq^{\mathtt{ds}}$ the greatest* ds*-preorder.*

[1] In the paper $\bullet$ will stand either for '`ds`', defined in §3.1, or '`s`', defined in §3.2.

The next lemma shows that the coinductive ds-preorder is preserved not only on single actions, but also on sequences of actions.

Lemma 8. *If $\sigma \preceq^{\mathtt{ds}} \rho$ and $\rho \stackrel{w}{\Longrightarrow} \rho'$, then $\exists\, \sigma'$ such that $\sigma \stackrel{w}{\Longrightarrow} \sigma'$ and $\sigma' \preceq^{\mathtt{ds}} \rho'$.*

Proof. By induction on the length of w. □

Theorem 9 (Soundness). *If $\sigma \preceq^{\mathtt{ds}} \rho$, then $\sigma \sqsubseteq^{\mathtt{ds}} \rho$.*

Proof. Take a composition C such that $(\sigma \parallel C)\!\searrow^{\mathtt{ds}}$ and let

$$\rho \parallel C = \rho_1 \parallel C_1 \longrightarrow \rho_2 \parallel C_2 \longrightarrow \cdots \longrightarrow \rho_n \parallel C_n$$

be a maximal computation from C. We must show that $(\rho_n \parallel C_n)\checkmark$. By Lemma 8, we know that $\sigma \parallel C \Longrightarrow \sigma_n \parallel C_n$ and $\sigma_n \preceq^{\mathtt{ds}} \rho_n$. Also, $\rho_n \downarrow \mathrm{R}$ for some (non-empty) R, because otherwise $\rho_n \longrightarrow$ and the computation from C we are considering would not be maximal. Let then S be such that $\sigma_n \Downarrow \mathrm{S}$ and $\mathrm{S} \subseteq \mathrm{R}$, and take σ'_n such that $\sigma_n \Longrightarrow \sigma'_n \not\longrightarrow$ and $\sigma'_n \downarrow \mathrm{S}$. Obviously $\sigma \parallel C \Longrightarrow \sigma'_n \parallel C_n$; furthermore, $C_n \not\longrightarrow$ since C_n is part of the final state of a maximal computation, and $\sigma'_n \not\longrightarrow$ by construction. Then, for all α such that $C_n \stackrel{\alpha}{\longrightarrow}$ it must be the case that $\sigma'_n \stackrel{\overline{\alpha}}{\not\longrightarrow}$ because otherwise from $\mathrm{S} \subseteq \mathrm{R}$ we would derive $\rho_n \stackrel{\overline{\alpha}}{\longrightarrow}$, which is impossible since $\rho_n \parallel C_n \not\longrightarrow$. It follows, then, that $\sigma \parallel C \Longrightarrow \sigma'_n \parallel C_n \not\longrightarrow$, and given that $(\sigma \parallel C)\!\searrow^{\mathtt{ds}}$ we have $(\sigma'_n \parallel C_n)\checkmark$. This, in turn, implies $C_n \checkmark$ and $\sigma'_n \checkmark$, and hence $\checkmark \in \mathrm{S}$. Now, from $\mathrm{S} \subseteq \mathrm{R}$ we know that $\rho_n \checkmark$, hence $(\rho_n \parallel C_n)\checkmark$ as desired. □

A further lemma shows that all contracts may be composed into at least one ds-compliant choreography. This is a direct consequence of the syntactic structure of our contracts (all choices in a contract must either end up with **1** or with a recursion variable) and the definition of ds-compliance.

Lemma 10. *For all σ there exists C such that $(\sigma \parallel C)\!\searrow^{\mathtt{ds}}$.*

Theorem 11 (Completeness). *If $\sigma \sqsubseteq^{\mathtt{ds}} \rho$, then $\sigma \preceq^{\mathtt{ds}} \rho$.*

Proof. We need to manipulate contract compositions in order to force their behavior. In particular we note $\alpha_1.C_1 + \alpha_2.C_2$ the composition such that if $\alpha_1.C_1 + \alpha_2.C_2 \stackrel{\gamma}{\Longrightarrow} C$, then $\gamma \in \{\alpha_1, \alpha_2\}$ and in addition $\gamma = \alpha_i$ implies $C \equiv C_i$ (for $i = 1, 2$, respectively). We omit the (rather lengthy) details of how these compositions can be formed so that they are determinate (in the sense of Definition 1), and move with the proof of our claim.

We show that $\mathcal{R} \stackrel{def}{=} \{(\sigma, \rho) \mid \sigma \sqsubseteq^{\mathtt{ds}} \rho\}$ is a coinductive ds-preorder. Assume $(\sigma, \rho) \in \mathcal{R}$: we examine the three clauses of the definition in turn.

Assume $\rho \longrightarrow \rho'$. By our hypothesis we know that for all C $(\sigma \parallel C)\!\searrow^{\mathtt{ds}}$ implies $(\rho \parallel C)\!\searrow^{\mathtt{ds}}$, so obviously $(\rho' \parallel C)\!\searrow^{\mathtt{ds}}$ and thus $(\sigma, \rho') \in \mathcal{R}$.

Take R such that $\rho \downarrow \mathrm{R}$: we reason by contradiction. Assume that there is no S such that $\sigma \Downarrow \mathrm{S}$ and $\mathrm{S} \subseteq \mathrm{R}$: we show that $\sigma \not\sqsubseteq^{\mathtt{ds}} \rho$. Let then $A = \{\alpha \in \mathcal{A} \mid \sigma \Downarrow \mathrm{S}$ and $\alpha \in \mathrm{S} \setminus \mathrm{R}\}$. By Lemma 10, for each $\alpha \in A$ there exists C_α such that

$(\sigma(\alpha) \parallel C_\alpha)\searrow^{\mathtt{ds}}$. Now, let $C = \sum_{\alpha\in A} \overline{\alpha}.C_\alpha$. By construction, $(\rho \parallel C)\not\searrow^{\mathtt{ds}}$, and similarly $(\rho \parallel C + \mathbf{1})\not\searrow^{\mathtt{ds}}$ if $\rho\not\checkmark$. On the other hand, again by construction, one easily sees that $(\sigma \parallel C+\mathbf{1})\searrow^{\mathtt{ds}}$. Furthermore, when $\rho\checkmark$, by our initial assumption we know that for all S such that $\sigma \Downarrow$ S, it must be the case that S $\supsetneq \{\checkmark\}$ (for otherwise S $\subseteq$ R): hence, in this case, we also have $(\sigma \parallel C)\searrow^{\mathtt{ds}}$. Summarizing, when $\rho\not\checkmark$, we have $(\sigma \parallel C)\searrow^{\mathtt{ds}}$ and $(\rho \parallel C)\not\searrow^{\mathtt{ds}}$. When $\rho\checkmark$, the same is true of $C + \mathbf{1}$. In both cases we have the desired contradiction.

Let $\rho \xrightarrow{\alpha} \rho'$. Again we reason by contradiction, on the two possible cases.

- $\sigma \overset{\alpha}{\not\Longrightarrow}$. By Lemma 10, there exists a composition C such that $C \overset{\overline{\alpha}}{\not\Longrightarrow}$ and $(\sigma \parallel C)\searrow^{\mathtt{ds}}$. Now, choose a name c fresh for ρ and C, and form the composition $C' = C+\overline{\alpha}.\overline{c}.\mathbf{1}$. We have that $(\sigma \parallel C')\searrow^{\mathtt{ds}}$ and $(\rho \parallel C')\not\searrow^{\mathtt{ds}}$, which contradicts the hypothesis that $\sigma \sqsubseteq^{\mathtt{ds}} \rho$ as desired.
- $\sigma \overset{\alpha}{\Longrightarrow}$ and there exists C such that $(\sigma(\alpha) \parallel C)\searrow^{\mathtt{ds}}$ but $(\rho' \parallel C)\not\searrow^{\mathtt{ds}}$. Then we define

$$C' = \left(\sum_{\sigma \overset{\beta}{\Longrightarrow} \text{ and } \beta\neq\alpha} \overline{\beta}.C_\beta \right) + \overline{\alpha}.C$$

  with C_β such that $(\sigma(\beta) \parallel C_\beta)\searrow^{\mathtt{ds}}$. Again, $(\sigma \parallel C')\searrow^{\mathtt{ds}}$ and $(\rho \parallel C')\not\searrow^{\mathtt{ds}}$ as desired. □

4.2 Characterizing the safe-preorder

The coinductive construction of the s-preorder arises as an extension of the one we just discussed. To motivate the construction, consider the following two contracts:

$$\sigma = \mathtt{rec}(\mathbf{x})\,.(a.(b.\mathbf{x} + \mathbf{1}) \oplus c.\mathbf{1}) \quad \text{and} \quad \sigma' = \mathtt{rec}(\mathbf{x})\,.(a.(b.\mathbf{x} + \mathbf{1})). \tag{1}$$

Applying Definition 7, one verifies that $\sigma \preceq^{\mathtt{ds}} \sigma'$. On the other hand, given the composition $C = \mathtt{rec}(\mathbf{x})\,.(\overline{a}.(\overline{b}.\mathbf{x})+\overline{c}.\mathbf{1})$ one has $(\sigma \parallel C)\searrow^{\mathtt{ds}}$ whereas $(\sigma' \parallel C)\not\searrow^{\mathtt{ds}}$.

In other words, the example in (1) shows that $\preceq^{\mathtt{ds}}$ is not sound for $\sqsubseteq^{\mathtt{s}}$. A closer look at the example shows that the problem is in the second of the clauses that define $\preceq$: in particular, to show that $\sigma \preceq^{\mathtt{ds}} \sigma'$ it is enough for σ' to match (with R) any one of the action sets S such that $\sigma \Downarrow$ S, disregarding the remaining ones. This is fine as long as we only look at *finite* maximal computations, as the choice of any of action sets is arbitrary and effectively excludes the others; it is unsound, instead, when the computation in σ may go back to the same point of choice, as a result of a loop, and select another set. This observation suggests how to rectify the construction, by keeping track of the states reached in the simulation game, and make a sound choice at the looping states.

A *contract-indexed* relation over contracts is a binary relation indexed by sets of contracts. We let H range over sets of contracts, and write $\sigma\ \mathcal{R}_H\ \rho$ to mean that σ and ρ are related by $\mathcal{R}$ at H.

Definition 12 (Coinductive safe-preorder). *A coinductive* s-preorder $\mathcal{R}$ *is contract-indexed relation such that* $\sigma\ \mathcal{R}_H\ \rho$ *implies the following conditions:*

– *if* $\rho \longrightarrow \rho'$, *then* $\sigma\ \mathcal{R}_{H\cup\{\rho\}}\ \rho'$
– *if* $\rho \downarrow \mathrm{R}$, *then*
  • *if* $\rho \notin H$, *then there exists* $\mathrm{S} \subseteq \mathrm{R}$ *such that* $\sigma \Downarrow \mathrm{S}$,
  • *if* $\rho \in H$, *then for every* S *such that* $\sigma \Downarrow \mathrm{S}$ *it holds* $\mathrm{S} \subseteq \mathrm{R}$,
– *if* $\rho \xrightarrow{\alpha} \rho'$, *then* $\sigma \stackrel{\alpha}{\Longrightarrow}$ *and* $\sigma(\alpha)\ \mathcal{R}_{H\cup\{\rho\}}\ \rho'$

We write $\sigma\mathcal{R}\rho$ *when* $\sigma\mathcal{R}_H\rho$ *for some* H, *and note* $\preceq^{\mathtt{s}}$ *the greatest* `s`*-preorder.*

Notice that the coinductive `s`-preorder is a conservative extension of the corresponding `ds`-preorder. The next lemma proves the same result as Lemma 8, now for the $\preceq^{\mathtt{s}}$ preorder.

Lemma 13. *Let* $\mathcal{R}$ *be a coinductive* `s`*-preorder. If* $\sigma\ \mathcal{R}_H\ \rho$ *for some* H *and* $\rho \stackrel{w}{\Longrightarrow} \rho' \not\longrightarrow$, *then there exist* σ', H' *such that* $\sigma \stackrel{w}{\Longrightarrow} \sigma'$ *with* $\sigma'\ \mathcal{R}_{H'}\ \rho'$ *and* $H \subseteq H'$. *Furthermore, if* $\rho' = \rho$ *(with* w *non empty), then* $\rho' \in H'$.

Proof. We show that given $\sigma\ \mathcal{R}_H\ \rho$, if $\rho \stackrel{\alpha_1}{\Longrightarrow} \rho_1 \stackrel{\alpha_2}{\Longrightarrow} \rho_2 \cdots \rho_{n-1} \stackrel{\alpha_n}{\Longrightarrow} \rho_n = \rho'$, then $\sigma \stackrel{\alpha_1}{\Longrightarrow} \sigma_1 \stackrel{\alpha_2}{\Longrightarrow} \sigma_2 \cdots \sigma_{n-1} \stackrel{\alpha_n}{\Longrightarrow} \sigma_n$ and $\sigma_i\ \mathcal{R}_{H_i}\ \rho_i$ for $i = 1 \ldots n$ with $H \subseteq H_1 \subseteq \ldots \subseteq H_{n-1} \subseteq H_n$. We proceed by induction on the length n of the sequence. For the basic step, we have that if $\rho \Longrightarrow \rho'$, then $\sigma\ \mathcal{R}_{H'}\ \rho'$ with $H \subseteq H'$ by the first item of Definition 12. For the induction step, assume that $n > 0$. Then, by the induction hypothesis, $\sigma \stackrel{\alpha_1}{\Longrightarrow} \sigma_1 \ldots \stackrel{\alpha_{n-1}}{\Longrightarrow} \sigma_{n-1}$ with $\sigma_i\ \mathcal{R}_{H_i}\ \rho_i$ for $i = 1 \ldots n-1$ and $H \subseteq H_1 \subseteq \ldots \subseteq H_{n-1}$. Now, since $\rho_{n-1} \stackrel{\alpha_n}{\Longrightarrow} \rho_n = \rho'$, by repeated applications of the first item, and one application of the last item in Definition 12 we obtain $\sigma_{n-1} \stackrel{\alpha_n}{\Longrightarrow}$ and $\sigma_{n-1}(\alpha_n)\ \mathcal{R}_{H_n}\ \rho'$ with $H_{n-1} \subseteq H_n$. We are done, as we can choose $\sigma_n = \sigma_{n-1}(\alpha_n)$.

In case $\rho' = \rho$, by Definition 12 we can immediately prove that when w is non empty $\rho \in H_1$, and thus also $\rho \in H_n$. □

Lemma 14. *Let* $\mathcal{R}$ *be a coinductive* `s`*-preorder. If* $\sigma\ \mathcal{R}\ \rho$ *and* $\rho \stackrel{w_1}{\Longrightarrow} \rho' \stackrel{w_2}{\Longrightarrow} \rho'$ *with* $\rho' \not\longrightarrow$, *then (i)* $\sigma \stackrel{w_1,w_2}{\Longrightarrow} \sigma'$ *with* $\sigma'\ \mathcal{R}\ \rho'$ *and (ii) for every* α *such that* $\sigma' \Longrightarrow \widehat{\sigma} \xrightarrow{\alpha} \sigma''$, *then* $\rho' \xrightarrow{\alpha} \rho''$ *and* $\sigma''\ \mathcal{R}\ \rho''$. *Furthermore,* $\widehat{\sigma}\checkmark$ *implies* $\rho'\checkmark$.

Proof. Let H be an index such that $\sigma\ \mathcal{R}_H\ \rho$ for some H. Then item *(i)* is a direct consequence of Lemma 13. For item *(ii)*, Lemma 13 applied to $\rho' \stackrel{w_2}{\Longrightarrow} \rho'$ says that $\sigma'\ \mathcal{R}_{H'}\ \rho'$ with $\rho' \in H'$. Since $\rho' \not\longrightarrow$ we apply the second item of Definition 12. In particular $\rho' \downarrow \mathrm{R}$ with $\mathrm{S} \subseteq \mathrm{R}$ for all S such that $\sigma' \Downarrow \mathrm{S}$, and this proves the thesis. □

Lemma 15. *Let* C *be a contract composition without finite maximal computations. Then there exists* C' *such that (i)* $C \Longrightarrow C'$ *and (ii) for every* C'' *such that* $C' \Longrightarrow C''$ *then also* $C'' \Longrightarrow C'$.

Proof. Since a composition expressed as a term of our language is a finite-state system, we prove the lemma by induction on the number of the different states reachable from C. Basic step, suppose that C can reach a single state. Then the only maximal computation is $C \longrightarrow C \longrightarrow C \longrightarrow \cdots$ and the thesis is verified

with $C' = C$. Induction step, assume that C can reach n different states, if $C' \Longrightarrow C$ for every C' such that $C \Longrightarrow C'$ then the thesis is verified. Otherwise, there exists C' such that $C \Longrightarrow C'$ and $C' \not\Longrightarrow C$. Then C' can reach at most $n-1$ states. Since C has no finite maximal computation, so does C'. Thus we apply the inductive hypothesis and we have: there exists C'', with $C \Longrightarrow C' \Longrightarrow C''$ such that if $C'' \Longrightarrow C'''$, then $C''' \Longrightarrow C''$. Hence we conclude the thesis. □

Corollary 16. *Let C be a contract composition without finite maximal computations. Then there exists a computation $C \longrightarrow C_1 \longrightarrow \cdots \longrightarrow C_n$ such that for every C' such that $C_n \Longrightarrow C'$ there exists $i \leq n$ such that $C' = C_i$.*

Proof. Lemma 15 says that there exists C' such that $C \Longrightarrow C'$ and, if $C_1, \ldots, C_n$ are all the states reachable from C' then $C_i \Longrightarrow C'$ for $i = 1, \ldots n$. Then consider the computation $C \Longrightarrow C' \Longrightarrow C_1 \Longrightarrow C' \Longrightarrow C_2 \Longrightarrow C' \cdots \Longrightarrow C_n \Longrightarrow C$ and conclude the thesis. □

Theorem 17 (Soundness). *If $\sigma \preceq^{\mathsf{s}} \rho$, then $\sigma \sqsubseteq^{\mathsf{s}} \rho$.*

Proof. We reason by contradiction. We assume that there exists C such that $(\sigma \parallel C)\!\searrow^{\mathsf{s}}$ but $(\rho \parallel C)\!\not\searrow^{\mathsf{s}}$, this means that *(i)* there exists a computation $\rho \parallel C \Longrightarrow \rho' \parallel C'$ such that $(\rho'' \parallel C'') \not\checkmark$ for every $\rho' \parallel C' \Longrightarrow \rho'' \parallel C''$.

If there exists a finite computation $\rho' \parallel C' \Longrightarrow \rho'' \parallel C''$ with *(ii)* $\rho'' \parallel C'' \not\Longrightarrow$ and $(\rho'' \parallel C'') \not\checkmark$, hence $\rho'' \not\longrightarrow$ and $C'' \not\longrightarrow$, moreover either $\rho'' \not\checkmark$ or $C'' \not\checkmark$. Now consider the list w of actions such that $\rho \stackrel{w}{\Longrightarrow} \rho''$ and $C \stackrel{\bar{w}}{\Longrightarrow} C''$, where $\bar{w}$ represents the list of actions performed by C to synchronize with w in order to reduce the whole composition $(\rho \parallel C)$ to $(\rho'' \parallel C'')$. From $\sigma \preceq^{\mathsf{s}} \rho$ and Lemma 13, there exist H, H' such that $\sigma \mathcal{R}_H \rho$, $\sigma' \mathcal{R}_{H'} \rho'$ with $H \subseteq H'$ and $\sigma \stackrel{w}{\Longrightarrow} \sigma'$. Let R such that $\rho' \downarrow \mathrm{R}$. Since $\sigma' \mathcal{R}_{H'} \rho'$, then $\sigma' \Longrightarrow \sigma'' \not\longrightarrow$ with *(iv)* $\sigma'' \downarrow \mathrm{S}$ and $\mathrm{S} \subseteq \mathrm{R}$. Hence we found the computation $\sigma \parallel C \Longrightarrow \sigma' \parallel C' \Longrightarrow \sigma'' \parallel C''$, where the first part is the synchronization on w between σ and C. Due to *(ii)*, *(iii)* and *(iv)*, then $\sigma'' \parallel C'' \not\longrightarrow$ and $\sigma'' \parallel C'' \not\checkmark$. We have $\sigma \parallel C\!\not\searrow^{\mathsf{s}}$, against the hypothesis.

Thus we conclude that $\rho' \parallel C'$ has no finite maximal computations. Then Corollary 16 says that $\rho' \parallel C' \longrightarrow \rho_1 \parallel C_1 \longrightarrow \cdots \longrightarrow \rho_n \parallel C_n$ and for every C^* such that $\rho_n \parallel C_n \Longrightarrow C^*$ there exists $i \leq n$ with $C^* = \rho_i \parallel C_i$. As done above, consider the list $w = w_1, w_2$ of actions such that $\rho \stackrel{w_1}{\Longrightarrow} \rho' \stackrel{w_2}{\Longrightarrow} \rho_n$ and $C \stackrel{\bar{w}}{\Longrightarrow} C_n$, were again $\bar{w}$ represents the list of actions performed by C to synchronize with w in order to reduce the whole composition $(\rho \parallel C)$ to $(\rho_n \parallel C_n)$. Again, by Lemma 13 *(v)* there exist H, H' such that $\sigma \mathcal{R}_H \rho$, $\sigma_n \mathcal{R}_{H'} \rho_n$ with $H \subseteq H'$ and $\sigma \stackrel{w}{\Longrightarrow} \sigma_n$.

We distinguish two cases. (1) If for every C^* such that $\rho_n \parallel C_n \Longrightarrow C^*$ we have $C^* = \rho_n \parallel \widehat{C}$, thus ρ_n does not perform any more action along the computation. Note that $\rho_n \not\longrightarrow$, then let R such that $\rho_n \downarrow \mathrm{R}$. Moreover, due to *(i)*, it holds $\rho_i \parallel C_i \not\checkmark$ for $i = 1 \ldots n$, hence *(vi)* either $\rho_i \not\checkmark$ or $C_i \not\checkmark$ for $i = 1 \ldots n$. Moreover, consider *(v)* and let σ'_n such that $\sigma_n \Longrightarrow \sigma'_n$ and $\sigma'_n \downarrow \mathrm{S}$ with $\mathrm{S} \subseteq \mathrm{R}$. Due to the assumption on $\rho_n \parallel C_n$ and *(vi)*: for every C^* such that $\sigma_n \parallel C_n \Longrightarrow C^*$ we have $C^* = \sigma_n \parallel \widehat{C}$, and also either $\sigma_n \not\checkmark$ or $C_i \not\checkmark$ for

$i = 1 \ldots n$. Thus $C^* \not\checkmark$ for every C^* such that $\sigma_n \parallel C_n \Longrightarrow C^*$. We conclude that $\sigma \parallel C \not\searrow^{\mathtt{s}}$, against the hypothesis. Thus the only possible case is (2): there exists at least one $\widehat{\rho} \neq \rho_n$ such that $\rho_n \parallel C_n \Longrightarrow \widehat{\rho} \parallel \widehat{C}$ and moreover for every $\widehat{\rho}$ such that $\rho_n \parallel C_n \Longrightarrow \widehat{\rho} \parallel \widehat{C}$ it holds $\widehat{\rho} = \rho_i$ for some $i < n$. Thus for every $\widehat{\rho}$ such that $\rho_n \parallel C_n \Longrightarrow \widehat{\rho} \parallel \widehat{C}$ it holds: $\rho \stackrel{v_1}{\Longrightarrow} \widehat{\rho} \stackrel{v_2}{\Longrightarrow} \rho_n \stackrel{v_3}{\Longrightarrow} \widehat{\rho}$. Thanks to Lemma 14 and *(v)* we conclude the following

Fact 18 *For every $\widehat{\rho}$ such that $\rho_n \parallel C_n \Longrightarrow \widehat{\rho} \parallel \widehat{C}$ with $\rho_n \stackrel{\widehat{w}}{\Longrightarrow} \widehat{\rho}$ and $\widehat{\rho} \not\longrightarrow$ there exist $\widehat{\sigma}$ and H' such that $\sigma_n \stackrel{\widehat{w}}{\Longrightarrow} \widehat{\sigma}$ and $\sigma_n \parallel C_n \Longrightarrow \widehat{\sigma} \parallel \widehat{C}$ with $\widehat{\sigma}\ \mathcal{R}_{H'}\ \widehat{\rho}$ and for every $\widehat{\sigma} \Longrightarrow \widehat{\sigma}' \stackrel{\alpha}{\longrightarrow} \sigma''$ then also $\widehat{\rho} \stackrel{\alpha}{\longrightarrow} \widehat{\rho}'$ and $\widehat{\sigma}''\ \mathcal{R}_{H''}\ \widehat{\rho}'$ for some H''. Furthermore $\widehat{\sigma}'' \checkmark$ implies $\widehat{\rho}' \checkmark$.*

Since $(\sigma \parallel C \searrow^{\mathtt{s}})$ there exists a computation $\sigma_n \parallel C_n \Longrightarrow \sigma^* \parallel C^*$ such that $(\sigma^* \parallel C^*)\checkmark$. In particular $\sigma_n \stackrel{w^*}{\Longrightarrow} \sigma^*$. If the sequence w^* is empty, then Fact 18 says that also $\widehat{\rho}\checkmark$. Hence $\rho_n \parallel C_n \Longrightarrow \rho_n \parallel C^*$ and $(\rho_n \parallel C^*)\checkmark$. In case w^* is not empty, we prove that also

$$\rho_n \stackrel{w^*}{\Longrightarrow} \rho^* \text{ with } \sigma^*\ \mathcal{R}_{H^*}\ \rho^*. \tag{2}$$

We proceed by induction on the length of w. Fact 18 gives the basis of the induction. For the induction step, let $w = \alpha, w'$. Then $\sigma_n \Longrightarrow \sigma'_n \stackrel{\alpha}{\longrightarrow} \sigma''$ and Fact 18 says that $\rho_n \stackrel{\alpha}{\longrightarrow} \rho'_n$ with $\sigma''_n\ \mathcal{R}_{H''}\ \rho'_n$. Now the induction hypothesis holds for σ''_n and ρ'_n, and we are done. Now, from (2), it follows that $\rho_n \parallel C_n \Longrightarrow \rho^* \parallel C^*$ and the fact that $\sigma^*\ \mathcal{R}_{H^*}\ \rho^*$ says that $(\rho^* \parallel C^*)\checkmark$. Hence we found a contradiction and we conclude $(\rho \parallel C)\searrow^{\mathtt{s}}$. □

The converse of Theorem 17 does not hold. Here is a counter-example. Let:

$$\sigma = \mathtt{rec}(\mathbf{x})\,(a.\mathbf{x} \oplus b.\mathbf{1}) \qquad \rho = \mathtt{rec}(\mathbf{x})\,(\,a.(a.\mathbf{x} \oplus b.\mathbf{1})\,) \tag{3}$$

We can easily see that $\sigma \sqsubseteq^{\mathtt{s}} \rho$, because $\sigma \stackrel{w}{\Longrightarrow}$ for all w such that $\rho \stackrel{w}{\Longrightarrow}$. In fact, given an arbitrary C, C must be able to respond to all of these traces, with a corresponding dual trace in $\sigma \parallel C$. Given our observation, the same is true in the composition $\rho \parallel C$. On the other hand, $\sigma \not\preceq^{\mathtt{s}} \rho$ because, after one round of the loop, $\rho \downarrow \{a\}$, whereas $\sigma \Downarrow \{b\}$ and $\{b\} \not\subseteq \{a\}$, which breaks the condition required by the coinductive game. The problem would seemingly be solved by replacing the third condition with the following, slightly weaker requirement: if $\rho \stackrel{\alpha}{\longrightarrow} \rho'$, then there exists σ' such that $\sigma \stackrel{\alpha}{\Longrightarrow} \sigma'$ and $\sigma'\ \mathcal{R}_{H\cup\{\rho\}}\ \rho'$. On the other hand, this coarser definition is unsound: taking the two contracts we discussed earlier in (1), one verifies that $\sigma \preceq^{\mathtt{s}} \sigma'$ even though $\sigma \not\sqsubseteq^{\mathtt{ds}} \sigma'$, hence $\sigma \not\sqsubseteq^{\mathtt{s}} \sigma'$.

As it turns out, characterizing the safe preorder exactly, is hard. Indeed, to our knowledge, no such characterization exists in the literature. The only related result we are aware of is the trace-based full-abstract characterization of *should* testing in [17]. However, the construction is non-effective, as it requires infinite sums to characterize the infinite traces that capture the possible test processes.

5 Filtered preorders

As the last step of our analysis, we show how the *filters* introduced in [8, 9] can be recast into the setting of general service compositions to achieve an expressive compliance-preserving substitution principle inside choreographies. Following [8, 9], we define filters as behavioral coercions that specify the legal flow of actions for individual contracts. Their syntax is defined by the following productions:

$$f \;:=\; \mathbf{0} \mid \alpha.f \mid f \times f \mid f \otimes f \mid \mathbf{x} \mid \mathtt{rec}(\mathbf{x})\, f.$$

Their semantics, in Table 2, is best understood by viewing a filter as a finite-state automaton accepting possibly infinite strings of actions, with $\times$ and $\otimes$ noting the intersection and union automata. Then, applying a filter f to a contract σ, as in $f(\sigma)$, corresponds to verify that the sequence of visible transitions made by the contract σ forms a string of the filter's language.

Transitions for filters

$$\alpha.f \overset{\alpha}{\longmapsto} f \qquad \frac{f\{\mathbf{x} := \mathtt{rec}(\mathbf{x})\, f\} \overset{\alpha}{\longmapsto} f'}{\mathtt{rec}(\mathbf{x})\, f \overset{\alpha}{\longmapsto} f'} \qquad \frac{f \overset{\alpha}{\longmapsto} f_\alpha \qquad g \overset{\alpha}{\longmapsto} g_\alpha}{f \otimes g \overset{\alpha}{\longmapsto} f_\alpha \otimes g_\alpha}$$

$$\frac{f \overset{\alpha}{\longmapsto} f_\alpha \qquad g \overset{\alpha}{\longmapsto} g_\alpha}{f \times g \overset{\alpha}{\longmapsto} f_\alpha \times g_\alpha} \qquad \frac{f \overset{\alpha}{\longmapsto} f_\alpha \qquad g \overset{\alpha}{\not\longmapsto}}{f \times g \overset{\alpha}{\longmapsto} f_\alpha} \qquad \frac{f \overset{\alpha}{\not\longmapsto} \qquad g \overset{\alpha}{\longmapsto} g_\alpha}{f \times g \overset{\alpha}{\longmapsto} g_\alpha}$$

Transitions for filtered contracts

$$\frac{\sigma \overset{\alpha}{\longrightarrow} \sigma' \qquad f \overset{\alpha}{\longmapsto} f'}{f(\sigma) \overset{\alpha}{\longrightarrow} f'(\sigma')} \qquad \frac{\sigma \longrightarrow \sigma'}{f(\sigma) \longrightarrow f(\sigma')} \qquad \frac{\sigma\,\checkmark}{f(\sigma)\,\checkmark}$$

Table 2. Dynamics of Filtered Contracts

In their original, client-server formulation by [8, 9], filters generalize the notion of *contract interface* introduced in [13]. Like filters, interfaces are intended to constrain the behavior of contracts, by defining the set of actions that a contract may legally engage in. However, while with filters this set may vary dynamically as the contract unfolds, with interfaces the set is determined statically and does not change over time. In addition, filters also play a role in strengthening the substitution principle based on compliance preorders. Specifically, given any compliance preorder $\leq^{\bullet}$ (whether coinductive or not), one defines a corresponding filtered preorder as the following relation: $\sigma \leq^{\mathtt{F}\bullet} \rho$ if there exists a filter f such that $\sigma \leq^{\bullet} f(\rho)$. Thus, even when $\sigma \not\leq^{\bullet} \rho$, one may still rely on a filter f

to justify the replacement of σ with $f(\rho)$, provided that $\sigma \leq^{\bullet} f(\rho)$. In a client-server setting,. the filtered preorder of [8, 9] generalizes the interface-indexed preorder of [13], which relates contracts based on a their ability to comply with clients that follow the discipline imposed by the indexing interface, disregarding all clients that do not follow that discipline.

Our present use of filters provides corresponding generalizations of the concepts of *input-output sets* and *input-output indexed preorder* by [3, 4] that parallel the notions of interface and interface indexed preorder in the analysis of multi-party compositions. In [4], the authors provide an effective decision procedure for their preorder based on the theory of should-testing [17]. In the rest of this section, we provide an effective construction for the filtered version of the coinductive safe-preorder. The same construction can be given, *mutatis mutandis*, for the deadlock-safe preorder.

Definition 19 (filtered s-preorder). *A filtered* $\mathtt{s}$*-preorder is a contract indexed relation* $\mathcal{F}$ *such that if* $\sigma \mathcal{F}_H \rho$*, then*

1. *if* $\rho \longrightarrow \rho'$*, then* $\sigma \mathcal{F}_{H'} \rho'$ *with* $H' = H \cup \{\rho\}$*,*
2. *else if* $\rho \downarrow \mathrm{R}$*, then*
   (a) *if* $\rho \notin H$*, then there exists* $\mathrm{S_R} \subseteq \mathrm{R}$ *such that* $\sigma \Downarrow \mathrm{S_R}$*, and for every* $\alpha \in \mathrm{S_R}$ *it is the case that* $\rho \xrightarrow{\alpha} \rho'$ *and* $\sigma \stackrel{\alpha}{\Longrightarrow}$ *with* $\sigma(\alpha) \mathcal{F}_{H'} \rho'$ *and* $H' = H \cup \{\rho\}$*.*
   (b) *if* $\rho \in H$*, then for every* S *such that* $\sigma \Downarrow \mathrm{S}$ *it holds* $\mathrm{S} \subseteq \mathrm{R}$*, and for every action* $\alpha \in \bigcup_{\sigma \Downarrow \mathrm{S}} \mathrm{S}$ *if* $\rho \xrightarrow{\alpha} \rho'$*, then* $\sigma \stackrel{\alpha}{\Longrightarrow}$ *with* $\sigma(\alpha) \mathcal{F}_{H'} \rho'$ *and* $H' = H \cup \{\rho\}$*.*

We write $\sigma \mathcal{F} \rho$ *whenever* $\sigma \mathcal{F}_H \rho$ *for some* H*, and note* $\sigma \preceq^{\mathtt{Fs}} \rho$ *the greatest filtered* $\mathtt{s}$*-preorder.*

Theorem 20. $\sigma \preceq^{\mathtt{Fs}} \rho$ *iff there exists a filter* f *such that* $\sigma \preceq^{\mathtt{s}} f(\rho)$*.*

Proof. Define *contract bisimilarity*, noted $\sim$, is the greatest symmetric relation such that $\sigma \sim \rho$ implies (i) $\sigma\checkmark$ iff $\rho\checkmark$, and (ii) if $\sigma \xrightarrow{\dot{\alpha}} \sigma'$, then also $\rho \xrightarrow{\dot{\alpha}} \rho'$ and $\sigma' \sim \rho'$. Clearly, $\sim \subseteq \preceq^{\mathtt{F}\bullet}$.

We proceed with the proof of the theorem, in the two directions in turn.
($\Longrightarrow$) Take a filtered $\mathtt{s}$-preorder $\mathcal{F}$. For every H and $(\sigma, \rho) \in \mathcal{F}_H$ we define

$$Set_H(\sigma, \rho) \stackrel{def}{=} \begin{cases} \bigcup_{\rho \Downarrow \mathrm{R}} \mathrm{S}_R & \text{if } \rho \notin H \text{ and } \mathrm{S}_R \text{ is given by item 2.}a \text{ of Definition 19} \\ \bigcup_{\sigma \Downarrow \mathrm{S}} \mathrm{S} & \text{if } \rho \in H \end{cases}$$

Then, given a set D of pairs of contracts, we define

$$f^D_{\sigma,\rho,H} \stackrel{def}{=} \begin{cases} \mathtt{rec}(\mathbf{x}_{(\sigma,\rho)}) \times_{\alpha \in Set_H(\sigma,\rho)} \alpha . f^{D \cup \{(\sigma,\rho)\}}_{\sigma(\alpha),\rho(\alpha),H \cup \{\rho\}} & \text{if } (\sigma,\rho) \notin D \\ \mathbf{x}_{(\sigma,\rho)} & \text{otherwise} \end{cases}$$

and let $f_{\sigma,\rho,H} = f^{\emptyset}_{\sigma,\rho,H}$. Since the reachable states are finite, f is well defined. Furthermore note that, if $f_{\sigma,\rho,H}(\rho) \stackrel{\alpha}{\Longrightarrow} f'(\rho')$ and $\sigma \stackrel{\alpha}{\Longrightarrow}$, then $f'(\rho') \sim f_{\sigma(\alpha),\rho',H \cup \{\rho\}}(\rho')$. Finally we define the following contract indexed relation:

$$\mathcal{R}_H \stackrel{def}{=} \{(\sigma, f(\rho)) : (\sigma, \rho) \in \mathcal{F}_H \text{ and } f(\rho) \sim f_{\sigma,\rho,H}(\rho)\}$$

We prove that $\mathcal{R}$ is a coinductive $\mathtt{s}$-preorder, by a case analysis of the items in the Definition 12. Given a filter f, and a set $\mathrm{R} \subseteq \mathcal{A} \cup \{\checkmark\}$, let $\mathrm{R}_{|f}$ note the set $\{\alpha \in \mathrm{R} \mid f \stackrel{\alpha}{\longmapsto}\}$. Let then $(\sigma, f(\rho)) \in \mathcal{R}_H$.

If $\rho \longrightarrow \rho'$, then also $f(\rho) \longrightarrow f(\rho')$ and we have $(\sigma, \rho') \in \mathcal{F}_{H'}$ with $H' = H \cup \{\rho\}$, so $(\sigma, f(\rho')) \in \mathcal{R}_{H'}$. If instead $\rho \downarrow \mathrm{R}$, we distinguishes two cases. *(i)* If $\rho \notin H$, then, since $(\sigma, \rho) \in \mathcal{F}_H$, there exists $\mathrm{S}_R \subseteq \mathrm{R}$ such that $\sigma \Downarrow \mathrm{S}_\mathrm{R}$ and by definition we have $\mathrm{S}_\mathrm{R} \subseteq \mathit{Set}_H(\sigma, \rho)$; so $\mathrm{S}_\mathrm{R} \subseteq \mathrm{R}_{|(f_{\sigma,\rho,H})}$. *(ii)* If $\rho \in H$, then, since $(\sigma, \rho) \in \mathcal{F}_H$, for every S such that $\sigma \Downarrow \mathrm{S}$ it holds $\mathrm{S} \subseteq \mathrm{R}$ and also $\mathrm{S} \subseteq \mathit{Set}_H(\sigma, \rho)$, hence $\mathrm{S} \subseteq \mathrm{R}_{|(f_{\sigma,\rho,H})}$.

Assume now $f_{\sigma,\rho,H}(\rho) \stackrel{\alpha}{\longrightarrow}$. Then there exists ρ' such that $\rho \stackrel{\alpha}{\longrightarrow} \rho'$, so $f_{\sigma,\rho}(\rho) \stackrel{\alpha}{\longrightarrow} f'(\rho')$, where $f'(\rho') \sim f_{\sigma',\rho',H'}(\rho')$ with $\sigma' = \sigma(\alpha)$ and $H' = H \cup \rho$. Since $f_{\sigma,\rho,H} \stackrel{\alpha}{\longmapsto}$ then $\sigma \stackrel{\alpha}{\Longrightarrow}$ with $(\sigma(\alpha), \rho') \in \mathcal{F}_{H'}$ and $H' = H \cup \{\rho\}$. We conclude that $(\sigma(\alpha), f'(\rho')) \in \mathcal{R}_{H'}$.
($\Longleftarrow$) Let $\mathcal{R}$ be a coinductive $\mathtt{s}$-preorder. Given H, we define

$$\mathcal{F}_H \stackrel{def}{=} \{(\sigma, \rho) \,|\, \text{there exists } f \text{ such that } (\sigma, f(\rho)) \in \mathcal{R}_{f(H)}\}$$

where $f(H) = \{f(\rho) : \rho \in H\}$. We show that $\mathcal{F}$ is a filtered $\mathtt{s}$-preorder. Take $(\sigma, \rho) \in \mathcal{F}_H$, i.e. $(\sigma, f(\rho)) \in \mathcal{R}_{f(H)}$.

If $\rho \longrightarrow \rho'$, then $f(\rho) \longrightarrow f'(\rho')$ and $(\sigma, f'(\rho')) \in \mathcal{R}_{f(H')}$ with $H' = H \cup \{\rho\}$, hence $(\sigma, \rho') \in \mathcal{F}_{H'}$. If instead $\rho \downarrow \mathrm{R}$, we distinguishes two cases. *(i)* If $\rho \notin H$, then also $f(\rho) \notin f(H)$ hence there exists S such that $\mathrm{S} \subseteq \mathrm{R}_{|f}$. Since $\mathrm{R}_{|f} \subseteq \mathrm{R}$ we have $\mathrm{S} \subseteq \mathrm{R}$. Take now $\alpha \in \mathrm{S}$, then $\alpha \in \mathrm{R}_{|f}$ so $f(\rho) \stackrel{\alpha}{\longrightarrow} f'(\rho')$. Now, from $(\sigma, f(\rho)) \in \mathcal{R}_{f(H)}$ we know that $(\sigma(\alpha), f'(\rho')) \in \mathcal{R}_{f(H')}$ with $H' = H \cup \{f(\rho)\}$, so $(\sigma(\alpha), \rho') \in \mathcal{F}_{H'}$. (*(ii)* If $\rho \in H$, then also $f(\rho) \in f(H)$. Now, for every S such that $\sigma \Downarrow \mathrm{S}$ it holds $\mathrm{S} \subseteq \mathrm{R}_{|f}$, hence $\mathrm{S} \subseteq \mathrm{R}_{|f}$ and for every action $\alpha \in \bigcup_{\sigma \Downarrow \mathrm{S}} \mathrm{S}$ if $f(\rho) \stackrel{\alpha}{\longrightarrow} f(\rho')$, then $\sigma \stackrel{\alpha}{\Longrightarrow}$ with $(\sigma(\alpha), f(\rho')) \in \mathcal{R}_{f(H')}$ and $H' = H \cup \{\rho\}$, hence also $(\sigma(\alpha), \rho') \in \mathcal{F}_{H'}$. □

6 Conclusion

We have developed a formal framework for the analysis of different theories of compliance in the literature. Besides investigating the relationships between the existing definitions of compliance, we have also shown how to obtain compliance preorders for multiparty service compositions, by recasting and generalizing the theory of behavioral coercions from [8, 9] to this setting. Our present endeavor continues on the line of work we initiated in [2]. There, we used filters to provide a new solution to the problem of web service adaptation within service compositions [18, 1, 10]. Specifically, we showed how filters may be employed as adapters to enforce the compliance of a choreography, by blocking the transition paths in all the components that may get the choreography stuck or trapped into a livelock. Here, our focus has been on providing effective techniques for the construction of expressive compliance preorders supporting contract replacement. Collectively, the resulting theory constitutes an elegant support for a formal analysis of component/service compliance, adaptation and replacement inside choreographies.

Acknowledgements We gratefully acknowledge comments from the anonymous referees.

References

1. Daniela Berardi, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Massimo Mecella. Automatic service composition based on behavioral descriptions. *Int. J. Cooperative Inf. Syst.*, 14(4):333–376, 2005.
2. G. Bernardi, M. Bugliesi, D. Macedonio, and S. Rossi. A theory of adaptable contract-based service composition. In *GlobalComp*. IEEE Computer Society, 2008.
3. M. Bravetti and G. Zavattaro. Contract based multi-party service composition. In *FSEN'07*, volume 4767 of *LNCS*, pages 207–222. Springer, 2007.
4. M. Bravetti and G. Zavattaro. Towards a unifying theory for choreography conformance and contract compliance. In *SC'07*, volume 4829 of *LNCS*, pages 34–50. Springer, 2007.
5. L. Caires and H. T. Vieira. Conversation types. In *ESOP'09*, volume 5502 of *LNCS*, pages 285–300. Springer, 2009.
6. M. Carbone, K. Honda, and N. Yoshida. Structured communication-centred programming for web services. In *ESOP'07*, LNCS, pages 2–17. Springer, 2007.
7. S. Carpineti, G. Castagna, C. Laneve, and L. Padovani. A formal account of contracts for web services. In *WS-FM'06*, volume 4184 of *LNCS*, pages 148–162. Springer, 2006.
8. G. Castagna, N. Gesbert, and L. Padovani. A theory of contracts for web services. In *POPL'08*, pages 261–272. ACM press, 2008.
9. G. Castagna, N. Gesbert, and L. Padovani. A theory of contracts for web services. *ACM Transactions on Programming Languages and Systems*, 2009. To appear.
10. Giuseppe De Giacomo and Sebastian Sardiña. Automatic synthesis of new behaviors from a library of available behaviors. In Manuela M. Veloso, editor, *IJCAI*, pages 1866–1871, 2007.
11. K. Honda, V. Vasconcelos, and M. Kubo. Language primitives and type discipline for structured communication-based programming. In *ESOP '98*, volume 1381 of *LNCS*, pages 122–138. Springer, 1998.
12. K. Honda, N. Yoshida, and M. Carbone. Multiparty asynchronous session types. In *POPL'08*, pages 273–284. ACM Press, 2008.
13. C. Laneve and L. Padovani. The *must* preorder revisited. In *CONCUR'07*, volume 4703 of *LNCS*, pages 212–225. Springer, 2007.
14. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
15. R. De Nicola and M. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984.
16. B. C. Pierce and D. Sangiorgi. Typing and subtyping for mobile processes. *Mathematical Structures in Computer Science*, 6(5):409–453, 1996.
17. A. Rensink and W. Vogler. Fair testing. *Information and Computation*, 205(2):125–198, 2007.
18. Paolo Traverso and Marco Pistore. Automated composition of semantic web services into executable processes. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 380–394. Springer, 2004.