Contractual testing

Maria Grazia Buscemi[1], Rocco De Nicola[1], and Hernán Melgratti[2]

[1] IMT Lucca Institute for Advanced Studies, Italy
{marzia.buscemi,rocco.denicola} @imtlucca.it
[2] FCEyN, University of Buenos Aires, Argentina
hmelgra@dc.uba.ar

Abstract. Variants of must testing approach have been successfully applied in Service Oriented Computing for capturing compliance between (contracts exposed by) a client and a service and for characterising safe replacement, namely the fact that compliance is preserved when a service exposing a 'smaller' contract is replaced by another one with a 'larger' contract. Nevertheless, in multi-party interactions, partners often lack full coordination capabilities. Such a scenario calls for less discriminating notions of testing in which observers are, e.g., the description of uncoordinated multiparty contexts or contexts that are unable to observe the complete behaviour of the process under test. In this paper we propose an extended notion of must preorder, called *contractual preorder*, according to which contracts are compared according to their ability to pass only the tests belonging to a given set. We show the generality of our framework by proving that preorders induced by existing notions of compliance in a distributed setting are instances of the contractual preorder when restricting to suitable sets of observers.

1 Introduction

Communication-centered programming has recently attracted interest as a result of the diffusion of service oriented computing and web technologies. A desired property of communication-centered systems is stuck-freedom, namely that every possible interaction between a pair of communicating partners ends successfully, in the sense that there are no messages waiting forever to be sent or sent messages which are never received. The theories of session types [11,7] and of contracts [3,4,1,8] are the most common frameworks adopted to ensure stuck-freedom. The key idea behind both approaches is to associate to a process a type (or *contract*) that gives an abstract description of the external, visible behavior of the process and to check if the respective types of a pair of processes which are expected to communicate do match. Contracts are basically CCS processes [9] describing communications between clients and services. Contracts come equipped with a notion of service *compliance* that characterises all the valid clients of a service, i.e., the clients that terminate any possible interaction with the service. In this sense, contracts can be used to statically ensure that the composition of two services is safe. Another key notion of contracts is *safe replacement*: a contract σ can be safely replaced by ρ if any valid client of σ is also a client of ρ. Specifically, in the theory of contracts compliance and safe replacement have been sucessfully characterized by using suitable variant of the *must testing* approach [5], which allows comparing processes according to the ability of the contexts to distinguishing them. Processes that are must-equivalent are characterized by the set of tests that they are able to pass: any test is defined as a unique process that runs in parallel with the tested service, namely all interactions with the observed service are handled by a unique, central process, i.e. the test. Technically, given two processes σ and ρ, $\sigma \sqsubseteq_{\mathbf{must}} \rho$, if ρ passes all tests that are passed by σ and, consequently, σ and ρ are regarded as equivalent, $\sigma \approx_{\mathbf{must}} \rho$, if they pass exactly the same tests.

When describing contexts in multiparty interactions such as those occurring in business processes, a key issue to be taken into account is the degree of mutual coordination that partners enjoy. In a typical situation, different parties interact with a service without communicating with each other or where a particular party is interested just on a subset of (inter)actions offered by another party. In this kind of contexts, the work in [10] investigates notions of *controllability* for workflow networks, that are a formal model of services based on Petri nets. A service is said *controllable* if it has at least a partner that can correctly interact with (this notion has been called *viability* in [4]). When moving to a multiparty setting, each service interacts with several partners. Hence, in a common scenario, the interface of a service is partitioned and each partner communicates by using just one part of the interface. The work in [10] highlights the fact that controllability in a multiparty setting highly depends on the coordination capabilities of partners. In particular, two different settings are presented: (i) partners are distributed and have no runtime communication capabilities (coordinated design choices are allowed, though) and (ii) partners are totally independent.

The main goal of this paper is to characterize alternative, less discriminating notions of preorders and process equivalences that take into account contexts with limited coordination capabilities like those studied in [10]. We address this issue in a process algebraic context [5]. As proposed in previous works (e.g., [8]), we rely on service behaviours descripted as CCS processes without τ's [6]. Processes are built up from invoke and accept activities, which are abstractly represented as input and output actions that take place over a set of channels or names. Basic actions can be composed sequentially or as alternatives. The original language of contracts studied in [6] does not use τ actions to represent internal computations of services. Instead, two different choice operators are provided: the internal choice operator $\oplus$ is meant to describe internal non-determinism, i.e., the service choices by itself one of the possible continuations, while the external operator $+$ offers a choice to the context. Moreover, the language of contracts does not provide any operator for parallel composition. Basically, it is assumed that all possible interleavings are made explicit in the description of the service and communication is used only for modelling interaction between different services. In this work we also deal with infinite behaviours described as recursive contracts.

We propose two notions of testing preorders that take into account the level of coordination between processes in a multi-party setting: in the first one, called *distributed* (must) preorder, the environment (i.e., the test) cannot state any causal dependency between the actions that take place over different parts of the interface, in the second one, called *local* (must) preorder, processes cannot be distinguished by contexts that have only a partial view of the tested processes and such that observers are unaware of the design choices made by the other parties. Next, we propose an extended notion of the classical must preorder, called *contractual* (must) preorder, according to which contracts are compared with respect to their ability to pass only the tests belonging to a given set. We show the generality of our notion by proving that distributed preorder and local preorder are instances of the contractual preorder when restricting to suitable sets of observers. Specifically, we prove that the distributed preorder can be characterized as a contractual preorder in which valid tests are unable to state any causal dependency between actions that take place in different partitions of the interface at execution time. We call such kind of observers *closed under name swapping*, since they allow to commute the order in which actions over different parts of the interface are executed. On the other side, we show that the local preorder can be defined as a contractual preorder in which the observers that are taken into account belong to the class of processes called *noisy* observers, which allow hidden actions to be executed at any time. Finally, we show that the local preorder is less discriminating than the distributed preorder, which in turns is less discriminating than the ordinary must preorder.

Synopsis The remainder of this paper is organised as follows. In §2 we recall the basics of the classical must testing approach recast for the language of contracts. In §3 and §4 we present the theory of distributed and local observers, respectively. In §5 we introduce the contractual preorder and in §6 we give the main results, namely that distributed preorder and local preorder are particular instances of the contractual preorder and that the distributed preorder relation implies the local one. Lastly, §7 discuss some future developments. Due to lack of space proofs are omitted, they can be found in [2].

2 Contracts (or CCS without τ's)

Let $\mathcal{N}$ be an infinite set of names ranged over by $a, b, \ldots$. As usual, we write co-names in $\overline{\mathcal{N}}$ as $\overline{a}, \overline{b}, \ldots$ and make $\overline{\overline{a}} = a$. We will use α, β to range over $(\mathcal{N} \cup \overline{\mathcal{N}})$. The set of contracts Σ is given by the following grammar.

$$\begin{array}{ll} \alpha ::= a \mid \overline{a} & a \in \mathcal{N} \\ \sigma ::= 0 \mid \alpha.\sigma \mid \sigma \oplus \sigma \mid \sigma + \sigma \mid X \mid \mathbf{rec}_X.\sigma & \end{array}$$

The contract 0 describes a service that does not perform any action. The contract $\alpha.\sigma$ stands for a services that is able to execute α and then continues as σ. The contract $\sigma + \rho$ describes a service that lets the client decide whether to continue as σ or as ρ, while $\sigma \oplus \rho$ stands for a service that internally decides whether to continue as σ or ρ. As usual, trailing 0's are omitted. Contracts will be considered modulo associativity of each sum operators. We usually write summations $\sigma_1 + \sigma_2 + \ldots + \sigma_n$ and $\sigma_1 \oplus \sigma_2 \oplus \ldots \oplus \sigma_n$ respectively as $\Sigma_{i \in \{1,\ldots,n\}} \sigma_i$ and $\bigoplus_{i \in \{1,\ldots,n\}} \sigma_i$. By convention, $\Sigma_{i \in \emptyset} \sigma_i = 0$. The behaviour $\mathbf{rec}_X.\sigma$ defines a possibly recursive contract whose recurrent pattern is σ. A (free) occurrence of the variable X in σ stands for the whole $\mathbf{rec}_X.\sigma$. We write $\mathtt{n}(\sigma)$ for the set of names a such that either a or $\overline{a}$ occur in σ.

The operational semantics of contracts is given in terms of the LTS defined below.

Definition 1 (Transition). *The transition relation of contracts, noted $\xrightarrow{\alpha}$, is the least relation satisfying the rules*

$$\alpha.\sigma \xrightarrow{\alpha} \sigma \quad \sigma \oplus \rho \xrightarrow{\tau} \sigma \quad \frac{\sigma \xrightarrow{\alpha} \sigma'}{\sigma + \rho \xrightarrow{\alpha} \sigma'} \quad \frac{\sigma \xrightarrow{\tau} \sigma'}{\sigma + \rho \xrightarrow{\tau} \sigma' + \rho} \qquad \mathbf{rec}_X.\sigma \xrightarrow{\tau} \sigma\{^{\mathbf{rec}_X.\sigma}/_X\}$$

and closed under mirror cases for the external and internal choices.

The rule for the internal choice $\oplus$ states that a contract $\sigma \oplus \rho$ non-deterministically selects one of its branches by executing an unlabelled transition. Differently, the external choice $\sigma + \rho$ selects one of its branches only after performing a visible action. Note that the internal reductions in one branch of an external choice do not select a continuation (i.e., both branches remain available). Recursive contracts $\mathbf{rec}_X.\sigma$ are unfolded with unlabelled reductions.

We write $\Rightarrow$ for the reflexive and transitive closure of $\xrightarrow{\tau}$; $\sigma \overset{\alpha}{\Longrightarrow} \rho$ for $\sigma \Longrightarrow\xrightarrow{\alpha}\Longrightarrow$; $\sigma \overset{\alpha_0 \ldots \alpha_n}{\Longrightarrow} \sigma'$ for $\sigma \overset{\alpha_0}{\Longrightarrow} \ldots \overset{\alpha_n}{\Longrightarrow} \sigma'$, and $\sigma \overset{\phi}{\Longrightarrow}$ with $\phi \in (\mathcal{N} \cup \overline{\mathcal{N}})^*$ if there exists σ' s.t. $\sigma \overset{\phi}{\Longrightarrow} \sigma'$. We write $\sigma \uparrow$ when σ diverges, i.e., there exists an infinite internal computation $\sigma = \sigma_0 \xrightarrow{\tau} \sigma_1 \xrightarrow{\tau} \ldots$, and $\sigma \downarrow$ if not $\sigma \uparrow$. We use $\mathtt{init}(\sigma)$ to denote the set of visible actions that could be emitted by σ, i.e., $\mathtt{init}(\sigma) = \{\alpha \mid \exists \sigma' s.t. \sigma \overset{\alpha}{\Longrightarrow} \sigma'\}$.

The following result will be useful in the following sections.

Proposition 1. *Let* σ *be a contract s.t.* $\sigma \overset{\phi}{\Longrightarrow}$. *Then,* $\exists n > 0$ *s.t.* $\#\{\rho \mid \sigma \overset{\phi}{\Longrightarrow} \rho\} = n$

Proof. (Hint) By straightforward induction on the length of the derivation $\overset{\phi}{\Longrightarrow}$.

Definition 2 (Ready sets). *Let* $\mathcal{P}_f(\mathcal{N} \cup \overline{\mathcal{N}})$ *be the set of finite parts of* $\mathcal{N} \cup \overline{\mathcal{N}}$, *called* ready sets. *Let also* $\sigma \Downarrow_R$ *be the least relation between contracts* $\sigma \in \Sigma$ *and ready sets* R *in* $\mathcal{P}_f(\mathcal{N} \cup \overline{\mathcal{N}})$ *such that*

$$0 \Downarrow_\emptyset \quad \alpha.\sigma \Downarrow_{\{\alpha\}} \quad \frac{\sigma \Downarrow_R \quad \rho \Downarrow_S}{\sigma + \rho \Downarrow_{R \cup S}} \quad \frac{\sigma \Downarrow_R}{\sigma \oplus \rho \Downarrow_R} \quad \frac{\rho \Downarrow_R}{\sigma \oplus \rho \Downarrow_R} \quad \frac{\sigma\{^{\mathbf{rec}_X.\sigma}/_X\} \Downarrow_R}{\mathbf{rec}_X.\sigma \Downarrow_R}$$

For a given ready set R, $\mathtt{co}(R)$ stands for its complementary ready set, i.e., $\mathtt{co}(R) = \{\overline{\alpha} \mid \alpha \in R\}$.

A contract (e.g., the service) interacts with another contract (e.g., the client) that executes in parallel. The notion of communicating client and service extends the transition relation to pairs of contracts or *configurations* as follows.

$$\frac{\sigma \xrightarrow{\tau} \sigma'}{\sigma|\rho \xrightarrow{\tau} \sigma'|\rho} \quad \frac{\rho \xrightarrow{\tau} \rho'}{\sigma|\rho \xrightarrow{\tau} \sigma|\rho'} \quad \frac{\sigma \xrightarrow{\alpha} \sigma' \quad \rho \xrightarrow{\overline{\alpha}} \rho'}{\sigma|\rho \xrightarrow{\tau} \sigma'|\rho'}$$

As behavioural semantics, we will consider the must-testing preorder [5]. As usual, the set O of *observers*, ranged over by o, is defined as the set of processes but with the additional distinguished action $\checkmark \notin Act$ used to report success. Then, the notion of passing a test and the corresponding behavioural equivalence are as follows.

Definition 3 (must). *A sequence of transitions* $\sigma_0|o_0 \xrightarrow{\tau} \ldots \xrightarrow{\tau} \sigma_k|o_k \xrightarrow{\tau} \ldots$ *is a maximal computation if either it is infinite or the last term* $\sigma_n|o_n$ *is such that* $\sigma_n|o_n \not\xrightarrow{\tau}$. *Let* σ **must** o *iff for each maximal interaction* $\sigma|o = \sigma_0|o_0 \xrightarrow{\tau} \ldots \xrightarrow{\tau} \sigma_k|o_k \xrightarrow{\tau} \ldots$ *there exists* $n \geq 0$ *such that* $o_n \xrightarrow{\checkmark}$.

Definition 4 (must preorder). $\sigma \sqsubseteq_{\mathbf{must}} \rho$ *iff* $\forall o \in O : \sigma$ **must** o *implies* ρ **must** o. *We write* $\sigma \approx_{\mathbf{must}} \rho$ *when both* $\sigma \sqsubseteq_{\mathbf{must}} \rho$ *and* $\rho \sqsubseteq_{\mathbf{must}} \sigma$.

3 Distributed observers

Processes that are must-equivalent are characterised by the set of tests that they are able to pass. In this setting, any test is defined as a unique process that runs in parallel with the tested service, namely all interactions with the observed service are handled by a unique, central process, i.e., the test. Therefore, we refer to the usual must-testing preorder as the *centralised preorder*. When shifting to a multiparty setting, the operations offered by a service are usually partitioned and each partner communicates with the service by using just one part of the interface. Moreover, different partners usually do not know each other and, therefore, they do not communicate directly with each other. As a direct consequence of this choice, the environment (i.e., the test) cannot establish any causal dependency between the actions that take place over different parts of the interface. Consider a scenario consisting of three processes σ,ρ_1 and ρ_2 in which σ and ρ_1 interact over a, σ and ρ_2 interact over b and ρ_1 and ρ_2 are totally independent. It turns out that it is impossible to find two uncoordinated processes ρ_1 and ρ_2 which are able to distinguish the following two implementations of σ: $a.b+a+b$ and $b.a+a+b$. Therefore, from the point of view of the multiparty setting described above the two implementations of σ should be considered equivalent. In this section we propose a notion of equivalence (obtained as a symmetric preorder) that equates processes that cannot be distinguished by contexts describing uncoordinated distributed processes.

We start by introducing the notion of uncoordinating observers.

Definition 5. *Let* $\{o_i\}_{i\in 0..n}$ *be a set of observers such that* $\mathtt{n}(o_i)\cap\mathtt{n}(o_j) = \{\checkmark\}$ *for all* $i \neq j$. *Then, we write* $\Pi_{\checkmark}^{i\in 0..n} o_i = o_0|_{\checkmark} \ldots |_{\checkmark} o_n$ *for the test representing the uncoordinated composition of* $\{o_i\}_{i\in 0..n}$*, which behaves as follows*

$$\frac{o_0 \xrightarrow{\alpha} o_0' \quad \alpha \neq \checkmark}{o_0|_{\checkmark} o_1 \xrightarrow{\alpha} o_0'|_{\checkmark} o_1} \qquad \frac{o_0 \xrightarrow{\checkmark} o_0' \quad o_1 \xrightarrow{\checkmark} o_1'}{o_0|_{\checkmark} o_1 \xrightarrow{\checkmark} o_0'|_{\checkmark} o_1'}$$

Note that the possible executions of a set of uncoordinated observers are all the possible interleavings of actions in $\mathcal{N}\cup\overline{\mathcal{N}}$ performed by the individual observers. The only action that is handled differently is $\checkmark$, i.e., a set of uncoordinated observers reports success only when every observer is able to report success (i.e., from the point of view of each partner the test is successful). In subsequent sections we will use the notion of ready sets extended over uncoordinated observers, which is defined as follows:

$$\frac{\sigma \Downarrow_R \quad \rho \Downarrow_S}{\sigma|_{\checkmark}\rho \Downarrow_{((R\cup S)\setminus\{\checkmark\})\cup(R\cap S)}}$$

Remark that a ready sets of $\sigma|_{\checkmark}\rho$ is just the union of a ready set of σ and a ready set of ρ for all actions but $\checkmark$. Action $\checkmark$ belongs to the ready set of the parallel composition when it is present in both ready sets (this reflects that $\checkmark$ is synchronized).

Below we introduce the notion of distributed preorder.

Definition 6 (Distributed (must) preorder $\sqsubseteq_{\mathbf{dmust}_{\mathbb{I}}}$). *Assume $\mathbb{I} = \{I_i\}_{i\in 0\ldots n}$ be a partition of $n(\sigma)$. We say $\sigma \sqsubseteq_{\mathbf{dmust}_{\mathbb{I}}} \rho$ iff, for all $\{o_i\}_{i\in 0,\ldots n}$ such that $n(o_i) \subseteq I_i \cup \{\checkmark\}$, σ **must** $\Pi^i_{\checkmark} o_i$ implies ρ **must** $\Pi^i_{\checkmark} o_i$, where $\Pi^i_{\checkmark}$ denotes the parallel composition of observers synchronized on $\checkmark$.*

Next result states that the distributed preorder is less discriminating than the centralized preorder.

Proposition 2. $\sigma \sqsubseteq_{\mathbf{must}} \rho$ *implies that, for every partition $\mathbb{I}$ of $n(\sigma)$,* $\sigma \sqsubseteq_{\mathbf{dmust}_{\mathbb{I}}} \rho$.

The proof of the above proposition is obtained indirectly with the notions introduced in the following sections (details are deferred to Section 6.1).

The converse of above proposition does not hold, i.e., $\sigma \sqsubseteq_{\mathbf{dmust}_{\mathbb{I}}} \rho$ does not imply $\sigma \sqsubseteq_{\mathbf{must}} \rho$. In fact, consider $\sigma = a.b+a+b$ and $\rho = b.a+a+b$. We show by contradiction that $\sigma \sqsubseteq_{\mathbf{dmust}_{\{\{a\},\{b\}\}}} \rho$. Suppose that $\sigma \not\sqsubseteq_{\mathbf{dmust}_{\{\{a\},\{b\}\}}} \rho$, then there exists $o = o^0|_{\checkmark} o^1$ s.t. $\mathtt{n}(o^1) \subseteq \{a,\checkmark\}$, $\mathtt{n}(o^2) \subseteq \{b,\checkmark\}$ and σ **must** o but ρ ~~must~~ o. Then, there exists a maximal computation $\rho|o = \rho_0|o_0 \xrightarrow{\tau} \ldots \rho_n|o_n \xrightarrow{\tau} \ldots$ and $o_i \stackrel{\checkmark}{\not\rightarrow}$ for all i. If the computation is finite, then the possible maximal computations have the following shape:

- $o \Longrightarrow o_n$ with $o_n \stackrel{\tau}{\not\rightarrow}$, $o_n \stackrel{\overline{a}}{\not\rightarrow}$, $o_n \stackrel{\overline{b}}{\not\rightarrow}$ and $o_j \stackrel{\checkmark}{\not\rightarrow}$ for all $j \leq n$, and $\rho_n = \rho$. In this case, there is also a maximal computation $\sigma|o = \sigma_0|o_0 \xrightarrow{\tau} \ldots \sigma_n|o_n$ with $o_n \stackrel{\tau}{\not\rightarrow}$, $o_j \stackrel{\checkmark}{\not\rightarrow}$ for all $j \leq n$, which contradicts the fact that σ **must** o.
- $o \stackrel{\overline{b}}{\Longrightarrow} o_n$ with $o_j \stackrel{\checkmark}{\not\rightarrow}$ for all $j \leq n$. In this case, $\rho_n = 0$ or $\rho_n = a$. For $\rho_n = 0$, it is immediate to check that there is also a maximal computation $\sigma|o = \sigma_0|o_0 \xrightarrow{\tau} \ldots \sigma_n|o_n$ with $o_n \stackrel{\tau}{\not\rightarrow}$ that transits the same states o_j and, hence, contradicts the fact that σ **must** o. Case $\rho_n = a$ implies $o_n \stackrel{\overline{a}}{\not\rightarrow}$. Since $o = o^1|o^2$ with $\mathtt{n}(o^1) \subseteq \{a,\checkmark\}$ and $\mathtt{n}(o^2) \subseteq \{b,\checkmark\}$ and $o \stackrel{\overline{b}}{\Longrightarrow} o_n \stackrel{\overline{a}}{\not\Longrightarrow}$ then $o^1 \stackrel{\overline{a}}{\not\Longrightarrow}$. Therefore, we can also build a maximal computation of $\sigma|o$ that contradicts σ **must** o.
- $o \stackrel{\overline{a}}{\Longrightarrow} o_n$ or $o \stackrel{\overline{a}\overline{b}}{\Longrightarrow} o_n$ follows analogously to the previous cases.

If computation is infinite, then the only possibility is $o_j \uparrow$ because ρ is finite. Then, it is easy to show that there exists also an infinite computation that contradicts σ **must** o. Therefore, $\sigma \sqsubseteq_{\mathbf{dmust}_{\mathbb{I}}} \rho$. It can be trivially checked that $\sigma \not\sqsubseteq_{\mathbf{must}} \rho$ (it suffices to consider the test $o = \overline{a}.\overline{b}.\checkmark$ and note that σ **must** o but ρ ~~must~~ o).

We remark that the characterization of uncoordinated observers corresponds to contexts with distributed control, in which any partner can rely on the behaviour of the other partners declared a priori (i.e., these contexts capture scenarios in which coordination between parties can be decided at design time). We remark that $a.b \not\approx_{\mathbf{dmust}_{\{\{a\},\{b\}\}}} b.a$. The observer $o = \overline{a}.\checkmark|_{\checkmark}\checkmark$ witnesses the fact that $a.b \not\sqsubseteq_{\mathbf{dmust}_{\{\{a\},\{b\}\}}} b.a$ while $o' = \checkmark|_{\checkmark}\overline{b}.\checkmark$ shows $b.a \not\sqsubseteq_{\mathbf{dmust}_{\{\{a\},\{b\}\}}} a.b$. In order to define a test that exhibits different behavior, partners in the tests cannot be chosen independently, e.g., once the component that interacts over a is fixed, the component interacting over b can be selected. Next section introduces a notion of process equivalence that leaves out also the assumption of design time coordination.

4 Local observers

In this section we explore a notion of equivalence that equates processes that cannot be distinguished by contexts that have only a partial view of the tested processes. As for the distributed observers, the targeted scenario is that of a service with a partitioned interface that interacts with two or more independent partners by using dedicated ports. In addition, observers are unaware of the design choices made by the other parties. Consider the scenario consisting of the three process σ,ρ and ψ where σ and ρ interact over a,b and σ and ψ interact over c. Consider the following two implementations for σ: $\sigma_1 = a.c+b.d$ and $\sigma_2 = a.d+b.c$. Note that $\sigma_1 \not\approx_{\mathbf{dmust}_{\{\{a,b\},\{c,d\}\}}} \sigma_2$. In fact they can be distinguished, e.g., by the distributed observer $o_1 = \overline{a}.\checkmark|_{\checkmark}\overline{c}.\checkmark$ ($\sigma_1 \not\sqsubseteq_{\mathbf{dmust}_{\{\{a,b\},\{c,d\}\}}} \sigma_2$) and $o_2 = \overline{b}.\checkmark|_{\checkmark}\overline{c}.\checkmark$ ($\sigma_2 \not\sqsubseteq_{\mathbf{dmust}_{\{\{a,b\},\{c,d\}\}}} \sigma_1$). Nevertheless, as far as ρ is concerned both implementations are equivalent since they allow ρ to select either a or b. Assuming that ψ is designed without a priori knowledge of the particular choices that will be made by ρ (e.g., $\rho = \overline{a}\oplus\overline{b}$), both implementations of σ are equivalent also from its perspective. In fact, ψ should be prepared to synchronize either over c or d. That is, the expected behaviour of σ when observing just the channels c and d can be described as $c\oplus d$. Basically, this behaviour corresponds to the abstraction that hides all actions over channels that are not observed.

Definition 7 **(Abstraction).** *Let $V \subseteq \mathcal{N}$ be a set of observable ports. We write $\sigma\mid_V$ for the abstraction of σ over V, which behaves as follows:*

$$\frac{\sigma \xrightarrow{\alpha} \sigma' \quad \alpha \in V}{\sigma\mid_V \xrightarrow{\alpha} \sigma'\mid_V} \qquad \frac{\sigma \xrightarrow{\alpha} \sigma' \quad \alpha \notin V}{\sigma\mid_V \xrightarrow{\tau} \sigma'\mid_V}$$

Abstraction corresponds to the usual notion of hidding in calculi like CCS [9].

Definition 8 **(Local must preorder $\sqsubseteq_{\mathbf{lmust}_V}$).** *Let $V \subseteq \mathcal{N}$ be a set of observable ports. We say $\sigma \sqsubseteq_{\mathbf{lmust}_V} \rho$ iff $\sigma\mid_V$ **must** o implies $\sigma\mid_V$ **must** o.*

The following two results show that the local preorder is less discriminating than the centralized preorder and that the local preorder is less discriminating than the distributed preorder, respectively. They are obtained as a consequence of other results presented in subsequent sections (details are deferred to Section 6.2).

Proposition 3. $\sigma \sqsubseteq_{\mathbf{must}} \rho$ *implies* $\sigma \sqsubseteq_{\mathbf{lmust}_V} \rho$ *for all V.*

Note that the converse does not hold, i.e., $\sigma \sqsubseteq_{\mathbf{lmust}_V} \rho$ does not imply $\sigma \sqsubseteq_{\mathbf{must}} \rho$. It is easy to check that $a.b \sqsubseteq_{\mathbf{lmust}_{\{a\}}} b.a$ but $a.b \not\sqsubseteq_{\mathbf{must}} b.a$.

Proposition 4. $\rho \sqsubseteq_{\mathbf{dmust}_{\{V,\Sigma\setminus V\}}} \sigma$ *implies* $\rho \sqsubseteq_{\mathbf{lmust}_V} \sigma$ *for all V.*

The converse does not hold, i.e., $\rho \sqsubseteq_{\mathbf{lmust}_V} \sigma$ does not imply $\rho \sqsubseteq_{\mathbf{dmust}_{\{V,\Sigma\setminus V\}}} \sigma$. It is easy to check that $a.c+b.d \sqsubseteq_{\mathbf{lmust}_{\{c,d\}}} a.d+b.c$ but $a.c+b.d \not\sqsubseteq_{\mathbf{dmust}_{\{\{a,b\},\{c,d\}\}}} a.d+b.c$ (as illustrated at the begining of this section).

5 Contractual must

In this section we present an extended notion of must preorder, called *contractual preorder*, which is parametric with respect to a set of contracts C (noted $\sqsubseteq^{C}_{\mathbf{must}}$). According to the contractual preorder, $\sigma \sqsubseteq^{C}_{\mathbf{must}} \rho$ if the service ρ passes all tests that are passed by σ, provided the tests are in C. We show that classical must preorder implies contractual preorder and that the two preorders coincide when C is the set of all possible tests. As we will see in subsequent sections, this generalised preorder allows for less discriminating notions of testing in which observers are, e.g., the description of uncoordinated multiparty contexts or contexts that are unable to observe the complete behaviour of the process under test.

Definition 9 (contractual preorder). *Let* σ *and* ρ *be two processes, and the* contract C *be a set of observers, namely* $C \subseteq O$*.*

$$\sigma \sqsubseteq^{C}_{\mathbf{must}} \rho \quad \textit{iff} \quad \forall o \in C : \sigma \ \mathbf{must}\ o \textit{ implies } \rho \ \mathbf{must}\ o.$$

Definition 10 (equivalence). $\sigma \approx^{C}_{\mathbf{must}} \rho$ *iff* $\sigma \sqsubseteq^{C}_{\mathbf{must}} \rho$ *and* $\rho \sqsubseteq^{C}_{\mathbf{must}} \sigma$.

Example 1. As an example, consider the two processes $\overline{a}$ and $\overline{a}+\overline{b}$, and the contract $C = \{a.\checkmark \oplus b.\checkmark\}$. It holds that $\overline{a} \not\approx^{\{a.\checkmark \oplus b.\checkmark\}}_{\mathbf{must}} \overline{a}+\overline{b}$. In fact, $\overline{a} \sqsubseteq^{\{a.\checkmark \oplus b.\checkmark\}}_{\mathbf{must}} \overline{a}+\overline{b}$ but $\overline{a}+\overline{b} \not\sqsubseteq^{\{a.\checkmark \oplus b.\checkmark\}}_{\mathbf{must}} \overline{a}$. Indeed, $\overline{a}+\overline{b}$ **must** $a.\checkmark \oplus b.\checkmark$ since there are two possible sequences of transitions stemming from $\overline{a}+\overline{b}\,|\,a.\checkmark \oplus b.\checkmark$:

1. $\overline{a}+\overline{b}\,|\,a.\checkmark \oplus b.\checkmark \xrightarrow{\tau} \overline{a}+\overline{b}\,|\,a.\checkmark \xrightarrow{\tau} \checkmark$ and
2. $\overline{a}+\overline{b}\,|\,a.\checkmark \oplus b.\checkmark \xrightarrow{\tau} \overline{a}+\overline{b}\,|\,b.\checkmark \xrightarrow{\tau} \checkmark$

and in both cases, after two steps an action $\checkmark$ can be taken. On the other side, $\overline{a}\ \not\mathbf{must}\ a.\checkmark \oplus b.\checkmark$, because the sequence of transitions

$$\overline{a}+\overline{b}\,|\,a.\checkmark \oplus b.\checkmark \xrightarrow{\tau} \overline{a}+\overline{b}\,|\,b.\checkmark \not\xrightarrow{\tau}$$

is such that the observer can never exhibit $\checkmark$. Following the same reasoning, we can show that $\overline{a} \not\approx_{\mathbf{must}} \overline{a}+\overline{b}$. Nevertheless, $\overline{a}$ and $\overline{a}+\overline{b}$ cannot be distinguished if we restrict to observers which do not act on b. Hence, for instance, $\overline{a} \approx^{\{a.\checkmark\}}_{\mathbf{must}} \overline{a}+\overline{b}$, since the only admissible test is $a.\checkmark$. On the other side, $\overline{a} \not\sqsubseteq^{\{a.\checkmark\}}_{\mathbf{must}} \overline{a} \oplus \overline{b}$. Indeed, $\overline{a}$ **must** $a.\checkmark$ while $\overline{a} \oplus \overline{b}\ \not\mathbf{must}\ a.\checkmark$, since there is a sequence $a.\checkmark\,|\,\overline{a} \oplus \overline{b} \xrightarrow{\tau} a.\checkmark\,|\,\overline{b} \not\rightarrow$. Conversely, the fact that $\overline{a} \oplus \overline{b}\ \not\mathbf{must}\ a.\checkmark$ implies that $\overline{a} \oplus \overline{b} \sqsubseteq^{\{a.\checkmark\}}_{\mathbf{must}} \overline{a}$.

Below we formally state the desired relationship between classical must preorder and contractual preorder. The proofs of the following results are straightforward.

Proposition 5. $\rho \sqsubseteq^{C}_{\mathbf{must}} \sigma$ *implies* $\rho \sqsubseteq^{C'}_{\mathbf{must}} \sigma$ *for all* $C' \subseteq C$*.*

Proposition 6. $\sqsubseteq_{\mathbf{must}} = \sqsubseteq^{O}_{\mathbf{must}}$.

Corollary 1. $\rho \sqsubseteq_{\mathbf{must}} \sigma$ *implies* $\rho \sqsubseteq^{C}_{\mathbf{must}} \sigma$ *for all* C*.*

6 Interesting Contract Languages

In this section we show how to recast the notions of distributed and local preorder into the framework of contractual testing by defining suitable contract languages. Firstly, we show that the distributed preorder corresponds to a contractual preorder in which possible contexts are closed by name swapping, i.e., if a contract allows a particular computation, then it also allows any possible permutation of actions of the original computation that respect the relative order of the actions of the same partition of the interface. Secondly, we show that the local preorder can be characterized as a contractual preorder in which contracts are closed with respect to arbitrary occurrences of hidden actions, i.e., a contract allows one particular computation if it allows any other computation that interleaves the original actions with an arbitrary number of hidden actions. Moreover, we prove the results claimed in Sections 3 and 4 about the discriminating power of the proposed preorder just by showing inclusion of contract languages.

6.1 Observers closed under name swapping

Definition 11. *Let σ be a process and $\mathbb{I} = \{I_i\}_{i \in 0 \ldots n}$ a partition of $N \in \mathcal{N}$ s.t. $n(\sigma) \subseteq N$. Then, σ is closed over name swapping with respect to $\mathbb{I}$ (written $\sigma \in \mathtt{Swap}_{\mathbb{I}}$) whenever $\forall \alpha \in I_i \cup \overline{I_i}, \beta \in I_j \cup \overline{I_j}$ with $i \neq j$ the following conditions hold:*

1. *if $\sigma \stackrel{\alpha}{\Longrightarrow}$ and $\sigma \stackrel{\beta}{\Longrightarrow}$ then $\exists R$ s.t. $\sigma \Downarrow_R$ and $\{\alpha, \beta\} \subseteq R$*
2. *$\sigma \Downarrow_R$ iff $\sigma \stackrel{\alpha}{\Longrightarrow} \sigma'$, $\sigma' \Downarrow_{R'}$ and $R \backslash (I_i \cup \overline{I_j} \cup \{\checkmark\}) = R' \backslash (I_i \cup \overline{I_i} \cup \{\checkmark\})$.*
3. *if $\sigma \stackrel{\alpha\beta}{\Longrightarrow} \sigma'$ then $\sigma \stackrel{\beta\alpha}{\Longrightarrow} \sigma'$ and $\sigma' \in \mathtt{Swap}_{\mathbb{I}}$.*

We call $\mathtt{Swap}_{\mathbb{I}}$ the set of all observers closed under name swapping w.r.t. $\mathbb{I}$.

The first condition above stands for the fact that an observer closed under name swapping cannot exhibit simultaneous, dependent choices into different partitions of the interface. Basically, we want to avoid contracts like $\sigma = a \oplus b$ when a and b belong to different partners. Note that σ describes a multiparty contract in which one partner decides to execute a and the other decides not to execute b at the same time or vice versa, i.e., the decision of one partner is conditioned by the decision taken by the other. Differently, we allow contracts like $\rho = (a \oplus 0) \oplus (a.b + b.a)$, which describes the independent behaviour of two partners, one that executes a and other that non-deterministically decide to execute b or terminate. The second condition states that the execution of the action of a partner does not affect the behaviour of the remaining partners. In other words, the internal choices of a partner have no effect on the behaviour observed over the remaining parts of the interface. In this way we avoid contracts like $\sigma = a + b$ when a and b are in different partitions. Note that σ describes two partners, which are ready to execute a and b, respectively. Nevertheless, after the execution of, e.g., a the action b is not available anymore. Finally, the third condition above states that the computation of two consecutive actions on different parts of the interface can be mimicked by the computation that exhibits the two performed actions in a different order, namely, one partner cannot force the order in which actions over different parts of the interface are performed.

Example 2. Some examples:

- $a.b \in \mathtt{Swap}_{\{\{a,b\}\}}$
- $a.b \notin \mathtt{Swap}_{\{\{a\},\{b\}\}}$ (Conditions 2 and 3 do not hold)
- $a.b + b.a \in \mathtt{Swap}_{\{\{a\},\{b\}\}}$
- $a.b \oplus b.a \notin \mathtt{Swap}_{\{\{a\},\{b\}\}}$ (Condition 1 does not hold)

The remainder of this section is devoted to formally state the correspondence of distributed observers and contextual testing for the language of observers closed by name swapping. The proofs of the following results are reported in [2].

Observers closed by name swapping as distributed observers We start by showing that any $\sigma \in \mathtt{Swap}_{\mathbb{I}}$ actually describes a distributed uncoordinated observer, which is obtained by combining the projections of σ over each part of the interface.

Definition 12. *Let $\sigma \in \mathtt{Swap}_{\mathbb{I}}$ with $\mathbb{I} = \{I_i\}_{i \in 0\ldots n}$. For $j \leq n$, we define $\mathbb{I}(\sigma)_j$, which is the projection of σ over I_j, as follows*

$$\mathbb{I}(\sigma)_j = \bigoplus_{R \in \mathbb{R}_j(\sigma)} \Sigma_{\alpha \in R} \alpha . \oplus_{\rho \in \sigma(\alpha)} \mathbb{I}(\rho)_j \underbrace{\oplus\, \Omega}_{if\ \sigma\uparrow}$$

$$\text{where } \mathbb{R}_j(\sigma) = \{R \cap (I_j \cup \overline{I_j} \cup \{\checkmark\}) | \sigma \Downarrow_R\}$$

The ready sets of a projection are the projections of the ready sets of the contract (i.e., they are in $\mathbb{R}_j(\sigma)$). Then, for each performed action α, the projection is able to select internally one of the original continuations of the contract σ after performing α. The last term says that a projection diverges when the contract diverges. We first remark that $\mathbb{I}(\sigma)_j$ is a well-defined contract, because the number of ready sets in $\mathbb{R}_j(\sigma)$, each ready sets is finite and the set $\sigma(\alpha)$ is finite by Proposition 1. Additionally, note that $\mathbb{I}(\sigma)_j \uparrow$ iff $\sigma \uparrow$ since both $\mathbb{R}_j(\sigma)$ and $\sigma(\alpha)$ are finite $\forall \sigma, \alpha$ (see Proposition 1).

The following technical results will be used when proving the correspondence of observers closed by name swapping and distributed contexts.

Proposition 7. *Let $\sigma \in \mathtt{Swap}_{\mathbb{I}}$ s.t. $\sigma \stackrel{\alpha}{\Longrightarrow} \sigma'$ with $\alpha \in (I_i \cup \overline{I_i} \cup \{\checkmark\})$. Then, $\mathbb{I}(\sigma)_i \stackrel{\alpha}{\Longrightarrow} \mathbb{I}(\sigma')_i$.*

Proof. Since $\sigma \stackrel{\alpha}{\Longrightarrow}$, there exists $R \in \mathbb{R}_i(\sigma)$ s.t. $a \in R$. Moreover, $\sigma' \in \sigma(\alpha)$. Therefore,

$$\mathbb{I}(\sigma)_i \stackrel{\tau}{\Longrightarrow} \Sigma_{\alpha \in R} \alpha . \oplus_{\rho \in \sigma(\alpha)} \mathbb{I}(\rho)_i \stackrel{\alpha}{\Longrightarrow} \oplus_{\rho \in \sigma(\alpha)} \mathbb{I}(\rho)_i \stackrel{\tau}{\Longrightarrow} \mathbb{I}(\sigma')_i.$$

Proposition 8. *Let $\sigma \in \mathtt{Swap}_{\mathbb{I}}$ s.t. $\sigma \stackrel{\alpha}{\Longrightarrow}$ with $\alpha \in (I_i \cup \overline{I_i})$. Then, $\forall j \neq i$: there exists σ' s.t. $\sigma \stackrel{\alpha}{\Longrightarrow} \sigma'$ and*

1. $\mathbb{I}(\sigma)_j \stackrel{\phi}{\Longrightarrow}$ *iff* $\mathbb{I}(\sigma')_j \stackrel{\phi}{\Longrightarrow}$.
2. $\mathbb{I}(\sigma)_j \stackrel{\phi}{\Longrightarrow} \rho$ *and* $\rho \Downarrow_R$ *iff* $\mathbb{I}(\sigma')_j \stackrel{\phi}{\Longrightarrow} \rho'$ *and* $\rho' \Downarrow_R$.

Proof. ($\Rightarrow$) By induction on the lenght of $|\phi| = n$.

- Base case $n = 0$: (1) is immediate. (2) From Definition 12 we note that $\mathbb{I}(\sigma')_j \Downarrow_R$ iff $R \in \mathbb{R}_j(\sigma')$ and $\mathbb{I}(\sigma)_j \Downarrow_R$ iff $R \in \mathbb{R}_j(\sigma)$. Then, we show $\mathbb{R}_j(\sigma') = \mathbb{R}_j(\sigma)$. Case ($\subseteq$): $R \in \mathbb{R}_j(\sigma')$ iff $\exists R' : \sigma' \Downarrow_{R'}$ and $R = R' \cap (I_j \cup \overline{I_j} \cup \{\checkmark\})$. By Definition 11(2), $\exists Q' : \sigma \Downarrow_{Q'}$ and $Q' \backslash I_i = R' \backslash I_i$. Moreover, $Q = Q' \cap (I_j \cup \overline{I_j} \cup \{\checkmark\}) \in \mathbb{R}_j(\sigma)$ by Definition 12. Since, $Q' \backslash (I_j \cup \overline{I_j} \cup \{\checkmark\}) = R' \backslash (I_j \cup \overline{I_j} \cup \{\checkmark\})$ we can conclude $R = R' \cap (I_j \cup \overline{I_j} \cup \{\checkmark\}) = Q' \cap (I_j \cup \overline{I_j} \cup \{\checkmark\}) = Q \in \mathbb{R}_j(\sigma)$. Case ($\supseteq$) is analogous.
- Inductive Step.

  $\mathbb{I}(\sigma)_j \xRightarrow{\beta}\xRightarrow{\phi'}$: Then, $\mathbb{I}(\sigma)_j \xRightarrow{\beta}\xRightarrow{\phi'}$. From

  $$\mathbb{I}(\sigma)_j = \bigoplus_{R \in \mathbb{R}_j(\sigma)} \Sigma_{\gamma \in R} \gamma. \oplus_{\kappa \in \sigma(\gamma)} \mathbb{I}(\kappa)_j \;\oplus\; \Omega \;\xRightarrow{\beta}\; \psi$$

  we conclude that there exist $R' \in \mathbb{R}_j(\sigma)$, $\beta \in R'$ and $\psi = \oplus_{\kappa \in \sigma(\beta)} \mathbb{I}(\kappa)_j$. Then, it must hold that

  $$\psi \xRightarrow{\tau} \mathbb{I}(\kappa_\psi)_j \xRightarrow{\phi'} \text{ with} \kappa_\psi \in \sigma(\beta) \tag{1}$$

  In addition, $\beta \in R'$ implies $\exists R$ s.t. $\sigma \Downarrow_R$ and $\beta \in R$, $\beta \xRightarrow{\beta}$. As $\sigma \in \mathtt{Swap}_{\mathbb{I}}$, $\sigma \xRightarrow{\beta}$ and $\sigma \xRightarrow{\alpha}$, then $\sigma \Downarrow_R$ with $\{\alpha, \beta\} \subseteq R$ by Definition 11(1), and $\sigma \xRightarrow{\alpha} \sigma''$ s.t. $\sigma'' \Downarrow R''$ with $R'' \backslash (I_j \cup \overline{I_j} \cup \{\checkmark\}) = R \backslash (I_j \cup \overline{I_j} \cup \{\checkmark\})$ by Definition 11(2). This implies that there exists $R'' \in \mathbb{R}_j(\sigma'')$ and $\beta \in R''$. Therefore

  $$\mathbb{I}(\sigma'')_j = \bigoplus_{R \in \mathbb{R}_j(\sigma'')} \Sigma_{\gamma \in R} \gamma. \oplus_{\kappa \in \sigma''(\gamma)} \mathbb{I}(\rho)_j \;\oplus\; \Omega \;\xRightarrow{\beta}\; \psi' = \oplus_{\kappa \in \sigma''(\beta)} \mathbb{I}(\kappa)_j \tag{2}$$

  Note that $\kappa \in \sigma''(\beta)$ implies $\kappa \in \sigma(\alpha\beta)$. By Definition 11(3), $\kappa \in \sigma(\alpha\beta)$ implies $\kappa \in \sigma(\beta\alpha)$. Therefore, for any possible choice of $\kappa_\psi \in \sigma(\beta)$ in Equation 1, it holds that $\kappa_\psi \xRightarrow{\alpha}$ (otherwise σ does not satisfy Definition 11 (2)). Then, for any possible $\kappa_\psi \xRightarrow{\alpha} \kappa'_\psi$ we use inductive hypothesis to conclude that (1) $\mathbb{I}(\kappa_\psi)_j \xRightarrow{\phi'}$ iff $\mathbb{I}(\kappa'_\psi)_j \xRightarrow{\phi'}$ and (2) $\mathbb{I}(\kappa_\psi)_j \xRightarrow{\phi'} \rho$ and $\rho \Downarrow_R$ iff $\mathbb{I}(\kappa'_\psi)_j \xRightarrow{\phi'} \rho'$ and $\rho' \Downarrow_R$.
  Then, the proof is concluded by noting that $\sigma(\beta\alpha) = \kappa'_\psi \in \sigma(\alpha\beta) = \sigma''(\beta)$, hence $\kappa'_\psi \in \sigma''(\beta)$. Finally, for ψ' in Equation 2 can reduce as follows $\psi' \xRightarrow{\tau} \mathbb{I}(\kappa'_\psi)_j$.

  ($\Leftarrow$) Then, $\mathbb{I}(\sigma')_j \xRightarrow{\beta}\xRightarrow{\phi'}$. From

  $$\mathbb{I}(\sigma')_j = \bigoplus_{R \in \mathbb{R}_j(\sigma')} \Sigma_{\gamma \in R} \gamma. \oplus_{\kappa \in \sigma'(\gamma)} \mathbb{I}(\kappa)_j \;\oplus\; \Omega \;\xRightarrow{\beta}\; \psi'$$

  we conclude that there exist $R' \in \mathbb{R}_j(\sigma')$, $\beta \in R'$ and $\psi' = \oplus_{\kappa \in \sigma'(\beta)} \mathbb{I}(\kappa)_j$. By the definition of $\mathbb{R}_j(\sigma')$, $\beta \in R'$ implies $\exists R$ s.t. $\sigma' \Downarrow_R$ and $\beta \in R$. By Definition 11(2), there exists R'' s.t. $\sigma \Downarrow R''$ with $R'' \backslash (I_j \cup \overline{I_j} \cup \{\checkmark\}) = R \backslash (I_j \cup \overline{I_j} \cup \{\checkmark\})$. Therefore,

$\sigma \overset{\alpha}{\Longrightarrow}$ and $\sigma \overset{\beta}{\Longrightarrow}$ and hence, $\sigma \Downarrow_R$ with $\{\alpha, \beta\} \subseteq R$ by Definition 11(1). Therefore, there exists $R'' \in \mathbb{R}_j(\sigma)$ with $\beta \in R''$. Consequently,

$$\mathbb{I}(\sigma)_j = \bigoplus_{R \in \mathbb{R}_j(\sigma)} \Sigma_{\gamma \in R} \gamma. \oplus_{\kappa \in \sigma(\gamma)} \mathbb{I}(\kappa)_j \;\oplus\; \Omega \overset{\beta}{\Longrightarrow} \oplus_{\kappa \in \sigma'(\beta)} \mathbb{I}(\kappa)_j$$

Then, the proof is completed as in the previous case.

Given $\phi \in \mathcal{S}^*$ and $S \subseteq \mathcal{S}$, $\phi \downharpoonright_S$ denotes the projection of ϕ over S, which is inductively defined as follows,

$$\begin{array}{rcl} \varepsilon \downharpoonright_S & = & \varepsilon \\ (\alpha\phi) \downharpoonright_S & = \phi \downharpoonright_S & \text{when } a \notin S \\ (\alpha\phi) \downharpoonright_S & = \alpha(\phi \downharpoonright_S) & \text{when } a \in S \end{array}$$

Proposition 9. *Let* $\sigma \in \mathtt{Swap}_{\mathbb{I}}$ *and* $\sigma \overset{\phi}{\Longrightarrow}$ *then:*

1. $\mathbb{I}(\sigma)_i \overset{\phi \downharpoonright_{I_i \cup \overline{I_i} \cup \{\checkmark\}}}{\Longrightarrow}$.
2. $\sigma \overset{\phi}{\Longrightarrow} \rho$ *and* $\rho \Downarrow_R$ *implies* $\mathbb{I}(\sigma)_i \overset{\phi \downharpoonright_{I_i \cup \overline{I_i} \cup \{\checkmark\}}}{\Longrightarrow} \rho'$ *and* $\rho' \Downarrow_{R \cap (I_i \cup \overline{I_i} \cup \{\checkmark\})}$.

Proof. By induction on $|\phi|$.

- Base case $\phi = \varepsilon$: (1) is immediate. (2) Follows by definition of $\mathbb{I}(\sigma)_i$.
- Inductive step $\phi = \alpha\phi'$ and $\sigma \overset{\alpha}{\Longrightarrow} \psi \overset{\phi'}{\Longrightarrow}$. There are two cases:
  - $\alpha \in I_i \cup \overline{I_i} \cup \{\checkmark\}$: by Proposition 7, $\mathbb{I}(\sigma)_i \overset{\alpha}{\Longrightarrow} \mathbb{I}(\psi)_i$. Then, the proof is completed by using inductive hypothesis.
  - $\alpha \notin I_i \cup \overline{I_i} \cup \{\checkmark\}$: Note that $\phi \downharpoonright_{I_i \cup \overline{I_i} \cup \{\checkmark\}} = \phi' \downharpoonright_{I_i \cup \overline{I_i} \cup \{\checkmark\}}$. By inductive hypothesis, (1) $\mathbb{I}(\psi)_i \overset{\phi' \downharpoonright_{I_i \cup \overline{I_i} \cup \{\checkmark\}}}{\Longrightarrow}$ and (2) $\psi \overset{\phi'}{\Longrightarrow} \rho'$ and $\rho' \Downarrow_R$ implies

    $$\mathbb{I}(\psi)_i \overset{\phi' \downharpoonright_{I_i \cup \overline{I_i} \cup \{\checkmark\}}}{\Longrightarrow} \psi' \quad \text{and} \quad \psi' \Downarrow_{R \cap (I_i \cup \overline{I_i} \cup \{\checkmark\})} .$$

    The proof is completed by using Proposition 8 to conclude (1) $\mathbb{I}(\sigma)_i \overset{\phi' \downharpoonright_{I_i \cup \overline{I_i} \cup \{\checkmark\}}}{\Longrightarrow}$ and (2) $\mathbb{I}(\sigma)_i \overset{\phi' \downharpoonright_{I_i \cup \overline{I_i} \cup \{\checkmark\}}}{\Longrightarrow} \rho'$ and $\rho' \Downarrow_{R \cap (I_i \cup \overline{I_i} \cup \{\checkmark\})}$.

Proposition 10. *Let* $\sigma \in \mathtt{Swap}_{\mathbb{I}}$ *and* $\mathbb{I}(\sigma)_i \overset{\phi}{\Longrightarrow} \rho$. *Then,* $\sigma \overset{\phi}{\Longrightarrow} \sigma'$ *and* $\rho = \mathbb{I}(\sigma')_i$.

Proof. By straightforward induction on the lenght $|\phi|$.

Proposition 11. *Let* $\sigma \in \mathtt{Swap}_{\mathbb{I}}$. *Then,* $\sigma \overset{\phi}{\Longrightarrow}$ *iff* $\Pi^i_{\checkmark} \mathbb{I}(\sigma)_i \overset{\phi}{\Longrightarrow}$.

Proof. ($\Rightarrow$) By induction on $|\phi|$. Base case $n = 0$ is immediate. For the inductive step $\phi = \alpha\phi'$, $\sigma \overset{\alpha}{\Longrightarrow} \sigma'' \overset{\phi'}{\Longrightarrow} \sigma'$ there are two cases:

– $\alpha \neq \checkmark \in I_j \cup \overline{I_j}$. By Proposition 7, $\mathbb{I}(\sigma)_j \stackrel{\alpha}{\Longrightarrow} \mathbb{I}(\sigma'')_j$. Therefore,

$$\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i \stackrel{\alpha}{\Longrightarrow} \mathbb{I}(\sigma'')_i|_{\checkmark}\Pi^{i\neq j}_{\checkmark}\mathbb{I}(\sigma)_i.$$

By inductive hypothesis, $\sigma'' \stackrel{\phi'}{\Longrightarrow}$ implies $\Pi^i_{\checkmark}\mathbb{I}(\sigma'')_i \stackrel{\phi'}{\Longrightarrow}$. Finally, the proof is completed by using repeatedly of Proposition 10 to conclude that $\Pi^i_{\checkmark}\mathbb{I}(\sigma'')_i \stackrel{\phi'}{\Longrightarrow}$ iff $\mathbb{I}(\sigma'')_i|\Pi^{i\neq j}_{\checkmark}\mathbb{I}(\sigma)_i \stackrel{\phi'}{\Longrightarrow}$.

– $a = \checkmark$. By definition of $\mathbb{I}(\sigma)_i$, $\mathbb{I}(\sigma)_i \stackrel{\checkmark}{\longrightarrow}$ for all i. Then, $\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i \stackrel{\checkmark}{\longrightarrow} \Pi^i_{\checkmark}\mathbb{I}(\sigma'')_i$. The proof is completed by inductive hypothesis.

($\Leftarrow$) The proof follows by induction on the length of $|\phi|$ and is analogous to the previous case.

Proposition 12. *Let* $\sigma \in \mathtt{Swap}_{\mathbb{I}}$ *and* $\sigma \stackrel{\phi}{\Longrightarrow} \sigma'$. *Then,* $\sigma' \Downarrow_R$ *iff* $(\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i) \stackrel{\phi}{\Longrightarrow} \psi$ *and* $\psi \Downarrow_R$.

Proof. ($\Rightarrow$) By induction on $|\phi|$.

– Base case $\phi = \varepsilon$: By definition, $(\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i) \Downarrow_R$ with $R = (\bigcup_i(R_i\setminus\{\checkmark\}))\cup\{\bigcap_i R_i\}$ and $\mathbb{I}(\sigma)_i \Downarrow_{R_i}$. Note that $R_i = (R'' \cap (I_i \cup \overline{I_i} \cup \{\checkmark\}))$ for $\sigma \Downarrow_{R''}$. Since $\mathbb{I}$ is a partition of $\mathtt{n}(\sigma)$, we conclude that $R = R''$.

– Inductive step $\phi = \alpha\phi'$, $\sigma \stackrel{\alpha}{\Longrightarrow} \sigma'' \stackrel{\phi'}{\Longrightarrow} \sigma'$. By Proposition 7, $\mathbb{I}(\sigma)_j \stackrel{\alpha}{\Longrightarrow} \mathbb{I}(\sigma'')_j$ Therefore,

$$\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i \stackrel{\alpha}{\Longrightarrow} \mathbb{I}(\sigma'))_i|\Pi^{i\neq j}_{\checkmark}\mathbb{I}(\sigma)_i.$$

Then, the proof is completed by using Proposition 10(2) and the definition of ready sets.

($\Leftarrow$) The proof follows by induction on the length of $|\phi|$ and is analogous to ($\Rightarrow$).

The following result states that any observer described as contract closed by name swapping can be written also as a distributed observer.

Lemma 1. *Let* $\sigma \in \mathtt{Swap}_{\mathbb{I}}$. *Then,* ρ **must** σ *iff* ρ **must**$\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i$

Proof. ($\Rightarrow$) By contradiction. Assume ρ ~~**must**~~ $\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i$. Then, there is a maximal computation $C \equiv \rho|\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i = \rho_0|P_0 \stackrel{\tau}{\longrightarrow} \ldots\rho_n|P_n \stackrel{\tau}{\longrightarrow} \ldots$ s.t. $P_j \stackrel{\checkmark}{\not\longrightarrow}$ for all j. There are two cases:

– Computation is finite, i.e., $\rho_n|P_n \stackrel{\tau}{\not\longrightarrow}$: Hence $\rho \stackrel{\phi}{\Longrightarrow} \rho_n$ and $\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i = P_0 \stackrel{\overline{\phi}}{\Longrightarrow}$. By Proposition 10, $\sigma \stackrel{\overline{\phi}}{\Longrightarrow}$. Therefore, we can build the computation $C' \equiv \rho|\sigma = \rho_0|\sigma_0 \stackrel{\tau}{\longrightarrow} \ldots\rho_0|\sigma_n$. First we note that C' is maximal. Otherwise, $\rho_n|\sigma_n \stackrel{\tau}{\longrightarrow}$, which implies $\rho_n \stackrel{\alpha}{\longrightarrow}$ and $\sigma_n \stackrel{\overline{\alpha}}{\longrightarrow}$. By Proposition 10, $P_n \stackrel{\overline{a}}{\longrightarrow}$ and hence $\rho_n|P_n \stackrel{\tau}{\longrightarrow}$ which contradicts the assumption that C is maximal. Since ρ **must** σ, it should be the case that some $\sigma_j \stackrel{\checkmark}{\longrightarrow}$. Since $\sigma \stackrel{\phi'}{\Longrightarrow} \sigma_j$ and $\sigma_j \stackrel{\checkmark}{\longrightarrow}$, we know, by Proposition 12, that $\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i \stackrel{\phi'}{\Longrightarrow} P_j$ and $P_j = \Pi^i_{\checkmark}\mathbb{I}(\sigma)_i(\phi') \stackrel{\checkmark}{\longrightarrow}$, which contradict the hypothesis that $P_j \stackrel{\checkmark}{\not\longrightarrow}$ for all j.

– Computation is infinite. Then $\rho \stackrel{\phi}{\Longrightarrow}$ and $\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i \stackrel{\overline{\phi}}{\Longrightarrow}$ with ϕ infinite. There are two cases: (i) there exists n s.t. $\rho_n \uparrow$ or $P_n \uparrow$, then reasoning as in the previous case we can exhibit a computation that contradicts ρ **must** σ. (ii) $\rho_0|P_0$ interacts infinitely often. By using Proposition 10, we can show that for any finite prefix ϕ' of ϕ, $\sigma \stackrel{\phi'}{\Longrightarrow} \sigma'$ and $\sigma' \Downarrow_R$ iff $\Pi^i_{\checkmark}\mathbb{I}(\sigma)_i) \stackrel{\phi}{\Longrightarrow}' \psi$ and $\psi \Downarrow_R$. Therefore, there exists an infinite computation of $\rho|\sigma$ that transits the terms $\sigma_0, \sigma_1, \ldots$ that has the same ready sets of $P_0, P_1, \ldots$. Hence, $\sigma_i \stackrel{\checkmark}{\not\to}$ for all i. This contradicts the assumption that ρ **must** σ.

($\Leftarrow$) Follows analogously to ($\Rightarrow$).

Distributed observers as contracts closed by name swapping We now show that for any distributed observer we can define an equivalent contract that is closed by name swapping, which describes all the possible interleavings in the execution of the distributed context.

Definition 13. *Let $\mathbb{I} = \{I_i\}_{i\in 0\ldots n}$ be a partition of $N \subset \mathcal{N}$ and $\{o_i\}_{i\in 0,\ldots n}$ a family of observers such that $n(o_i) \subseteq I_i \cup \overline{I_i} \cup \{\checkmark\}$. Then, the merge of $\{o_i\}_{i\in 0,\ldots n}$ is*

$$\mathbb{I}(\{o_i\}_{i\in 0\ldots n}) = \bigoplus_{R\in\mathbb{R}(\{o_i\})} \Sigma_{\gamma\in R}\gamma.\oplus_{\kappa\in\{o_i\}_{i\in 0\ldots n}(\gamma)}\mathbb{I}(\kappa) \quad \overbrace{\oplus\,\Omega}^{\textit{if some } o_i\uparrow}$$

$$\textit{where } \mathbb{R}(\{o_i\}_{i\in 0\ldots n}) = \{(\cup_{i\in 0\ldots n}(R_i\backslash\{\checkmark\}))\cup(\cap_{i\in 0\ldots n}R_i) \mid o_i \Downarrow_{R_i}\} \textit{ and}$$
$$\{o_i\}_{i\in 0\ldots n}(\gamma) = \{\{o'_i\}_{i\in 0\ldots n}|\gamma = \checkmark, o'_i \in o_i(\checkmark)\} \cup \{\{o_0, \ldots o'_j \ldots, o_n\}|\gamma \neq \checkmark, o'_j \in o_j(\gamma)\}\}$$

Basically, the ready sets of the merge corresponds to the combination of the ready sets of all distributed observers which are represented by $\mathbb{R}(\{o_i\}_{i\in 0\ldots n})$. We remark that $\checkmark$ is in a ready set of the merge only when it is in all the ready sets that are being merged. Similarly, the continuation for an action γ in some ready set is the internal choice of one merge describing the behaviour of the system after performing γ, which are denoted by $\kappa \in \{o_i\}_{i\in 0\ldots n}(\gamma)$ (all observers must perform γ when $\gamma = \checkmark$, otherwise only component performs γ). We remark also that $\mathbb{I}(\{o_i\}_{i\in 0\ldots n})$ diverges only when some o_i diverges.

Proposition 13. $\mathbb{I}(\{o_i\}_{i\in 0\ldots n})$ *is closed over name swapping with respect to* $\mathbb{I}$.

Proof. 1. Assume $\alpha \in I_i \cup \overline{I_i}$, $\beta \in I_j \cup \overline{I_j}$ and $i \neq j$. Note that $\mathbb{I}(\{o_i\}_{i\in 0\ldots n}) \stackrel{\alpha}{\Longrightarrow}$ and $\mathbb{I}(\{o_i\}_{i\in 0\ldots n}) \stackrel{\beta}{\Longrightarrow}$ imply that there exist R_i, R_j with $\alpha \in R_i$, $\beta \in R_j$, $o_i \Downarrow_{R_i}$ and $o_j \Downarrow_{R_j}$. Therefore, there exists R s.t. $\{\alpha, \beta\} \subseteq (R_i \cup R_j)\backslash\{\checkmark\} \subseteq R$ and $R \in \mathbb{R}(\{o_i\}_{i\in 0\ldots n})$.

2. Let $\alpha \in I_j \cup \overline{I_i}$ and $(\mathbb{I}(\{o_i\}_{i\in 0\ldots n})) \stackrel{\alpha}{\Longrightarrow}$. Then, $\mathbb{I}(\{o_i\}_{i\in 0\ldots n}) \Downarrow_R$ implies $R \in \mathbb{R}(\{o_i\}_{i\in 0\ldots n})$. It is easy to check that $\forall R \in \mathbb{R}(\{o_i\}_{i\in 0\ldots n})$ there exists $R' \in \mathbb{R}(o'_j, \{o_i\}_{i\neq j})$ with $\o_j \stackrel{\alpha}{\Longrightarrow} \o'_j$ s.t. $R\backslash(I_j \cup \overline{I_j} \cup \{\checkmark\}) = R'\backslash(I_j \cup \overline{I_j} \cup \{\checkmark\})$. Therefore, $(\mathbb{I}(\{o_i\}_{i\in 0\ldots n})) \stackrel{\alpha}{\Longrightarrow}$ ψ and $\psi \Downarrow_{R'}$.

3. It is immediate to check that $(\mathbb{I}(\{o_i\}_{i\in 0\ldots n})) \stackrel{\alpha\beta}{\Longrightarrow} \psi$ implies $(\mathbb{I}(\{o_i\}_{i\in 0\ldots n})) \stackrel{\beta\alpha}{\Longrightarrow} \psi$. Then, the proof is completed by showing that the set $\{\mathbb{I}(\{o_i\}_{i\in 0\ldots n})|\mathtt{n}(o_i) \subseteq I_i\}$ is closed under reduction.

Proposition 14. $\mathbb{I}(\{o_i\}_{i\in 0\ldots n}) \Downarrow_R$ *iff* $\Pi^i_{\checkmark} o_i \Downarrow_R$.

Proof. Follows immediately by noting that $\mathbb{I}(\{o_i\}_{i\in 0\ldots n}) \Downarrow_R$ iff $R \in \mathbb{R}(\{o_i\})$ iff $\Pi^i_{\checkmark} o_i \Downarrow_R$.

Proposition 15. $\Pi^i_{\checkmark} o_i \stackrel{\alpha}{\Longrightarrow} o$ *iff* $\mathbb{I}(\{o_i\}_{i\in 0\ldots n}) \stackrel{\alpha}{\Longrightarrow} P$ *and* $o = \Pi^i_{\checkmark} o'_i$ *and* $P = \mathbb{I}(\{o'_i\}_{i\in 0\ldots n})$

Proof. ($\Rightarrow$) There are two cases:

1. $a \neq \checkmark \in I_j \cup \overline{I_j}$: Then $o = o'_j|_{\checkmark} \Pi^{i\neq j}_{\checkmark} o_i$ with $\phi_j \stackrel{\alpha}{\Longrightarrow} o'_j$. Since $o_j \stackrel{\alpha}{\rightarrow} o'_j$, there exists R_j s.t. $o_j \Downarrow_{R_j}$ and $a \in R_j$. Therefore, there exists $R \in \mathbb{R}(\{o_i\})$ s.t. $a \in R$. Moreover, $\mathbb{I}(\{o_i\}_{i\in 0\ldots n}) \stackrel{\alpha}{\Longrightarrow} \mathbb{I}(o'_j, \{o_i\}_{i\neq j})$.
2. $a = \checkmark$: Then $o'_i = o_i(\checkmark)$ for all i. It is easy to check that $P = \mathbb{I}(\{o_i(\checkmark)\}_{i\in 0\ldots n})$.

($\Leftarrow$) Follows analogously.

Lemma 2. *Let* $\sigma \in \mathtt{Swap}_{\mathbb{I}}$. *Then,* ρ **must** $\Pi^i_{\checkmark} o_i$ *iff* ρ **must** $\mathbb{I}(\{o_i\}_{i\in 0\ldots n})$

Proof. The proof follows by contradiction analogously to Lemma 1 (in this case we rely on Propositions 14 and 15).

We can now show that the distributed preorder coincides with the contractual preorder for observers closed by name swapping.

Theorem 1. $\sqsubseteq_{\mathbf{dmust}_{\mathbb{I}}} = \sqsubseteq^{\mathtt{Swap}_{\mathbb{I}}}_{\mathbf{must}}$.

Proof.

($\subseteq$) By contradiction. Assume $\sigma \sqsubseteq_{\mathbf{dmust}_{\mathbb{I}}} \rho$ and $\sigma \not\sqsubseteq^{\mathtt{Swap}_{\mathbb{I}}}_{\mathbf{must}} \rho$. Then, there exists $o \in \mathtt{Swap}_{\mathbb{I}}$ s.t. σ **must** o and ρ $\not$**must** o. By Lemma 1, σ **must**$\Pi^i_{\checkmark} \mathbb{I}(o)_i$ and ρ $\not$**must** $\Pi^i_{\checkmark} \mathbb{I}(o)_i$, which contradicts $\sigma \sqsubseteq_{\mathbf{dmust}_{\mathbb{I}}} \rho$.

($\supseteq$) Follows analogously by using Lemma 2.

Remark Note that Proposition 2 can be obtained as a corollary of the above result and Proposition 5 after noting that $\mathtt{Swap}_{\mathbb{I}} \subseteq O$ for all $\mathbb{I}$.

6.2 Noisy observers

We introduce the notion of *noisy observer*, namely a contract that allows one particular computation if it allows any other computation that interleaves the original actions with an arbitrary number of hidden actions. The set of noisy observers will be used as the contract language that is suited to characterise the local preorder introduced in §4 in terms of contractual testing.

Definition 14. *Let o be an observer and $H \subseteq \mathcal{N}$ be a set of (noisy) channels. We say that o is a* noisy observer *over H whenever the following two conditions hold:*

1. $o \stackrel{\alpha}{\Longrightarrow} o$ *for all* $\alpha \in H \cup \overline{H}$
2. $o \stackrel{\alpha}{\Longrightarrow} o'$ *and* $\alpha \notin H \cup \overline{H}$ *then either o' is a noisy observer over H or* $\forall R : o' \Downarrow R$ *implies* $R = \emptyset$.

We call $\mathtt{Noisy}_H$ *the set of all noisy observers over* H.

The first condition above stands for the fact that a noisy observer after performing an action in H (namely, a *hidden* action) remains in the same state. The reason for this condition is to avoid contracts that allow distinguishing processes that "behave" the same over visible actions while performing possibly different invisible actions. For instance, for $H = \{a\}$ we want to rule out contracts like $a.a.b.\checkmark$ in which the number of synchronizations over a is relevant to exhibit an action $\checkmark$. Conversely, we allow contracts like $\mathbf{rec}_X(b.(\mathtt{rec}_Y.(a.Y+0)+a.X)$ in which an occurrence of action b can be interleaved by an arbitrary number of a's. The second condition requires that a noisy observer after exhibiting a visible action becomes either a noisy observer or an inert process. According to this condition, we discard contracts like $\mathbf{rec}_X(b.c.(\mathtt{rec}_Y.(a.Y+0)+a.X)$ in which performing or not an invisible action after b is crucial.

Example 3. Some examples:

- $a.b \in \mathtt{Noisy}_\emptyset$
- $0 \not\in \mathtt{Noisy}_{\{a\}}$ (Condition 1 is violated)
- $\mathbf{rec}_X(a.X \oplus b.\mathbf{rec}_X a.X) \in \mathtt{Noisy}_{\{a\}}$

Below we formally state the correspondence of local observers and contextual testing for the language of noisy observers (Corollary 3). The proofs of the following results are reported in [2]. We start by introducing the notion of canonical noisy observers of an arbitrary observer σ with respect to H, which basically makes σ a noisy observer over H.

Definition 15 (canonical noisy observers). *Let $V \subseteq \mathcal{N}$ be a set of observable ports. The* canonical noisy observer *of an observer o with respect to a set H of names such that $H \cap V = \emptyset$, written $\mathtt{nf}_V^H(o)$, is defined as follows:*

$$\mathtt{nf}_V^H(o) = \begin{cases} \mathbf{rec}_X.\,(\bigoplus_{R\in\mathtt{Ready}(\sigma)}(\Sigma_{\alpha\in R}\alpha.\bigoplus_{\rho\in\sigma(\alpha)}\mathtt{nf}_V^H(\rho)) + \Sigma_{\beta\in H\cup\overline{H}}\beta.X) & \text{if } o\downarrow \\ \mathbf{rec}_X.\,(\bigoplus_{R\in\mathtt{Ready}(\sigma)}(\Sigma_{\alpha\in R}\alpha.\bigoplus_{\rho\in\sigma(\alpha)}\mathtt{nf}_V^H(\rho)) + \Sigma_{\beta\in H\cup\overline{H}}\beta.X) \oplus \Omega & \text{if } o\uparrow \end{cases}$$

where $\mathtt{Ready}(\sigma) = \{R | \sigma \Downarrow_R\}$

Proposition 16. $\mathtt{nf}_V^H(o) \in \mathtt{Noisy}_H$.

Proof. (Hint) By induction on the structure of o.

Proposition 17. *If σ **must** o and $o \in \mathtt{Noisy}_H$ with $(n(\sigma)\setminus V) \subseteq H$ and $H \cap V = \emptyset$ for some V, then σ **must** $\mathtt{nf}_V^H(o)$.*

Proof. (Hint) By contradiction. The proof exploits the fact that o and $\mathtt{nf}_V^H(o)$ have the same traces if $o \in \mathtt{Noisy}_H$.

Proposition 18. *If σ **must** $\mathtt{nf}_V^H(o)$ for some $o \in \mathtt{Noisy}_H$ then σ **must** o.*

Proof. (Hint) By contradiction. The proof exploits the fact that o and $\mathtt{nf}_V^H(o)$ have the same traces if $o \in \mathtt{Noisy}_H$.

Proposition 19. *Let V be a set of visible ports. For σ a process such that $\mathtt{n}(\sigma) \subseteq V$ and o an observer, for every configuration $\sigma|o = \sigma_0|o_0 \to \ldots$, if $o_n \xrightarrow{\checkmark}$ then $o_0 \xRightarrow{\phi} o_n$ and $n(\phi) \subseteq V$.*

Proof. (Hint) By induction of the length of configurations $\sigma|o = \sigma_0|o_0 \to \ldots$.

Proposition 20. *If $\sigma \lfloor_V$ **must** o then σ **must** $\mathtt{nf}_V^H(o)$, for all H such that $(n(\sigma) \setminus V) \subseteq H$ and $H \cap V = \emptyset$.*

Proof. By contradiction. Suppose that there exists H such that $n(\sigma) \subseteq H$ and $H \cap V = \emptyset$ and σ $\cancel{\textbf{must}}$ $\mathtt{nf}_V^H(o)$, namely there exists a maximal interaction $C \equiv \sigma|\mathtt{nf}_V^H(o) = \sigma_0|o'_0 \xrightarrow{\tau} \ldots \xrightarrow{\tau} \sigma_k|o'_k \xrightarrow{\tau} \ldots$ such that (i) C is finite, $\sigma_k|o'_k$ is the last term and $o'_n \stackrel{\checkmark}{\not\to}$ for every $k \geq n \geq 0$ or (ii) C is infinite and $o'_n \stackrel{\checkmark}{\not\to}$ for every $n \geq 0$.

- C is finite. Hence $\sigma \xRightarrow{\phi} \sigma_k$ and $\mathtt{nf}_V^H(o) \xRightarrow{\overline{\phi}} o'_k$. Therefore, we can build a computation $C' \equiv \sigma \lfloor_V |o = \sigma'_0|o_0 \to \ldots \to \sigma_h|o_h$ with $h \leq k$, as $\sigma \lfloor_V$ can only synchronise on actions in V. First note that C' is maximal: otherwise, $\sigma'_h|o_h \to$, i.e., $\sigma'_h \xrightarrow{\alpha}$ and $o_h \xrightarrow{\overline{\alpha}}$ with $\alpha \in V$ (without lost of generality we can assume that any synchronisation over names $\alpha \in V$ is included in the trace ϕ) and, hence $\sigma_k \xRightarrow{\phi'}$ and $o'_k \xRightarrow{\overline{\phi'}}$ with $\phi' \lfloor_V = \alpha$ which contradicts the assumption that C is maximal. Since $\sigma \lfloor_V$ **must** o, it should be the case that some $o_j \xrightarrow{\checkmark}$. However, by Proposition 19, $o_j \xrightarrow{\checkmark}$ implies $o_0 \xRightarrow{\phi''} o_j$ and $n(\phi'') \subseteq V$. Then, there exists $i \leq k$ such that $o'_i \xrightarrow{\checkmark}$, which contradicts the (absurd) hypothesis that $o'_j \stackrel{\checkmark}{\not\to}$ for all $j \leq k$.
- C is infinite. Then $\sigma \xRightarrow{\phi}$ and $\mathtt{nf}_V^H(o) \xRightarrow{\overline{\phi}}$ with ϕ infinite. Therefore, we can build a computation $C' \equiv \sigma \lfloor_V |o = \sigma'_0|o_0 \to \ldots \to \sigma_h|o_h$ with $h \leq k$, as $\sigma \lfloor_V$ can only synchronise on actions in V. If C' is finite the proof proceeds much like in the previous case. If C' is infinite, by the fact that $\sigma \lfloor_V$ **must** o, $o_i \stackrel{\checkmark}{\not\to}$ for some $i \geq 0$. Necessarily, $o_0 \xRightarrow{\phi''} o_i$ and $n(\phi'') \subseteq V$. Then, $o'_l \xrightarrow{\checkmark}$, for some $l \geq 0$, which contradicts the absurd hypothesis $o'_j \stackrel{\checkmark}{\not\to}$ for all $j \geq 0$.

Proposition 21. *If σ **must** $\mathtt{nf}_V^H(o)$ for some o and $(n(\sigma) \setminus V) \subseteq H$ then $\sigma \lfloor_V$ **must** $\mathtt{nf}_V^H(o)$.*

Proof. By contradiction. Suppose that $\sigma \lfloor_V \cancel{\textbf{must}}$ $\mathtt{nf}_V^H(o)$, namely there exists a maximal interaction $C \equiv \sigma \lfloor_V |\mathtt{nf}_V^H(o) = \sigma'_0|o'_0 \xrightarrow{\tau} \ldots \xrightarrow{\tau} \sigma'_k|o'_k \xrightarrow{\tau} \ldots$ such that (i) C is finite, $\sigma'_k|o'_k$ is the last term and $o'_n \stackrel{\checkmark}{\not\to}$ for every $k \geq n \geq 0$ or (ii) C is infinite and $o_n \stackrel{\checkmark}{\not\to}$ for every $n \geq 0$.

- C is finite. Hence, $\sigma' \xRightarrow{\phi} \sigma'_k$ and $\mathtt{nf}_V^H(o) \xRightarrow{\overline{\phi}} o'_k$. Therefore, we can build a computation $C' \equiv \sigma|\mathtt{nf}_V^H(o) = \sigma_0|o'_0 \to \ldots \to \sigma_h|o'_h \to \ldots$. There are two cases.
  - If C' is finite with last term $\sigma_h|o'_h$, without loss of generality we can assume that $\sigma_h \stackrel{\alpha}{\not\to}$ and $o'_h \stackrel{\overline{\alpha}}{\not\to}$ with $\alpha \in H$ (since, we can assume any synchornisation over names in H to be included in ϕ). Moreover it holds that $h \geq k$, as $\sigma \lfloor_V$ can

only synchronise on actions in V. Necessarily, it cannot happen that $\sigma_h \stackrel{\phi'}{\Longrightarrow}$ and $o'_h \stackrel{\phi'}{\not\Longrightarrow}$, with $\phi' \downharpoonright_V = \beta$ and $\beta \in V$ as it would imply that $\sigma'_k | o'_k$ can eventually synchronise over β, thus contradicting the fact that C is maximal. By observing the fact that the only additional actions in C' wrt to C are in H and by definition of $\mathtt{Noisy}_H$ we can conclude that $o'_j \stackrel{\checkmark}{\not\to}$ for all $j \leq h$, which is a contradiction.
  - If C' is infinite, again we exploit the fact that the only additional actions in C are in H and the definition of $\mathtt{Noisy}_H$ to reach the contradiction that $o'_i \stackrel{\checkmark}{\not\to}$ for all $i \geq 0$.
- C is infinite and $o_n \stackrel{\checkmark}{\not\to}$ for every $n \geq 0$. As shown above, we can build the configuration $C' \equiv \sigma | \mathtt{nf}^H_V(o) = \sigma_0 | o'_0 \to \ldots$ which necessarily is infinite. By proceeding as in the previous case we reach a contradiction.

Corollary 2. $\sigma \sqsubseteq_{\mathbf{lmust}_V} \rho$ *iff* $\sigma \sqsubseteq^{\{\mathtt{nf}^H_V(o)\}}_{\mathbf{must}} \rho$ *with* $H = n(\sigma + \rho) \setminus V$.

Proof. Immediate consequence of Propositions 20 and 21.

Corollary 3. $\sigma \sqsubseteq_{\mathbf{lmust}_V} \rho$ *iff* $\sigma \sqsubseteq^{\mathtt{Noisy}_H}_{\mathbf{must}} \rho$ *with* $H = n(\sigma + \rho) \setminus V$.

Proof.
($\Rightarrow$) Suppose σ **must** o, with $o \in \mathtt{Noisy}_H$. By Proposition 17, σ **must** $\mathtt{nf}^H_V(o)$. By Proposition 21, $\sigma \downharpoonright_V$ **must** $\mathtt{nf}^H_V(o)$. By hypothesis, $\rho \downharpoonright_V$ **must** $\mathtt{nf}^H_V(o)$. By Proposition 20, ρ **must** $\mathtt{nf}^H_V(\mathtt{nf}^H_V(o))$. Hence, by applying twice Proposition 18, ρ **must** o.
($\Leftarrow$) Suppose $\sigma \downharpoonright_V$ **must** o. By Proposition 20, σ **must** $\mathtt{nf}^H_V(o)$. By hypothesis, ρ **must** $\mathtt{nf}^H_V(o)$. By Proposition 21, $\rho \downharpoonright_V$ **must** $\mathtt{nf}^H_V(o)$. By Proposition 19, $\rho \downharpoonright_V$ **must**o.

Remark Note that Proposition 3 can be obtained as a corollary of the above result and Proposition 5 after noting that $\mathtt{Noisy}_H \subseteq O$ for all H.

We dedicate the last part of this section to show that noisy observers (and therefore, local preorder) are less discriminating than observers closed under name swapping (and hence, distributed preorder). We start by showing that $\mathtt{nf}^H_V(o)$ is closed over name swapping with respect to visibles and hidden names.

Proposition 22. $\mathtt{nf}^H_V(o) \in \mathtt{Swap}_{\{H,V\}}$.

Proof. We actually show that the set $\{\oplus_i \mathtt{nf}^H_V(o_i)\} \subseteq \mathtt{Swap}_{\{H,V\}}$. 1) Follows straightforwardly because $\oplus_i \mathtt{nf}^H_V(o_i) \Downarrow R$ implies $o \Downarrow R'$ and $R = R' \cup H$. 2) There are two cases: (a) if $\alpha \in H$ then it follows immediately since $\oplus_i \mathtt{nf}^H_V(o_i) \stackrel{\alpha}{\Longrightarrow} \mathtt{nf}^H_V(o_i)$, (b) $\alpha \in V$ and $\oplus_i \mathtt{nf}^H_V(o_i) \stackrel{\alpha}{\Longrightarrow} o'$. Note by definition of $\oplus_i \mathtt{nf}^H_V(o_i)$ that $o' \Downarrow R$ implies $H \subseteq R$. 3) It is immediate that $\oplus_i \mathtt{nf}^H_V(o_i) \stackrel{\alpha\beta}{\Longrightarrow} o'$ implies $\oplus_i \mathtt{nf}^H_V(o_i) \stackrel{\beta\alpha}{\Longrightarrow} o'$. Then, the proof is completed by showing that the set $\{\oplus_i \mathtt{nf}^H_V(o_i)\}$ is closed under reductions.

Remark Note that Proposition 3 can be obtained as a consequence of the results presented in the previous sections. In fact, $\{\mathtt{nf}^H_V(o)\} \subseteq \mathtt{Swap}_{\{V,H\}}$ by Proposition 22. By Corollary 2, $\sigma \sqsubseteq_{\mathbf{lmust}_V} \rho$ iff $\sigma \sqsubseteq^{\{\mathtt{nf}^H_V(o)\}}_{\mathbf{must}} \rho$. By Proposition 5, $\sigma \sqsubseteq^{\{\mathtt{nf}^H_V(o)\}}_{\mathbf{must}} \rho$ implies $\sigma \sqsubseteq^{\mathtt{Swap}_{\{V,H\}}}_{\mathbf{must}} \rho$, which implies $\sigma \sqsubseteq_{\mathbf{dmust}_{\{V,H\}}} \rho$ by Theorem 1.

7 Conclusions

In this paper we have explored different refinements of the must testing preorder tailored to capture behaviour equivalences in multiparty settings. In particular, we considered two different settings in which contexts of a service are represented by processes exhibiting distributed control. The first variant, called *distributed (must) preorder*, corresponds to multiparty contexts without runtime communication but coordinated design choices. The second one, called *local (must) preorder*, requires that parties are completely independent and none of them can assume any particular behaviour about the partners. We have shown that such notions can be recast into a parameterized version of the must testing preorder, called *contractual* preorder, in which tests are taken from a precise subset of all possible observers (this subset is the parameter of the preorder). The parameter can be seen as the specification of a "contract" that any context observes while interacting with the tested service. As expected, the discriminating power of the induced equivalences decreases as contract languages became smaller. We have shown that the distributed preorder corresponds to a particular language of observers, called $\mathtt{Swap}_{\mathbb{I}}$, whose specification states that the order of execution of events over different parts of the interface can be swapped. Similarly, we associated the local preorder with a class of observers named $\mathtt{Noisy}_H$, that can always execute actions over set H without compromising its behaviour. Interestingly, we show that $\mathtt{Noisy}_H \subseteq \mathtt{Swap}_{\mathbb{I}}$, which implies that local preorder is less discriminating than distributed preorder (in other words, more coordinated contexts hava more discriminating power).

As future work, we plan to establish a formal connection of the notions presented here with the theory of contracts developed in [8,3,4]. In this respect, we would like to study whether the notions of distributed, local and, in general, contractual preorders provide refined versions of the compliance and subcontract relations that could be used efficiently to solve problems like, for instance, searching for services. In this sense, it would be worth studying the definition of suitable deduction systems for the subcontract relations induced by the new preorders. Since contractual preorder makes explicit the assumptions about the contexts in which a particular service should be used, the study of coercions (or filters) looks like a promising mechanism for computing the minimal adjustments required by a service in order to fulfill a particular contract.

References

1. M. Bravetti and G. Zavattaro. Towards a unifying theory for choreography conformance and contract compliance. In *SC*, volume 4829 of *Lect. Notes in Comput. Sci.*, pages 34–50. Springer Verlag, 2007.
2. M. G. Buscemi, R. De Nicola, and H. Melgratti. Contractual testing. Available at `http://eprints.imtlucca.it/1261/`.
3. G. Castagna, N. Gesbert, and L. Padovani. A theory of contracts for web services. In *POPL*, pages 261–272, 2008.
4. G. Castagna, N. Gesbert, and L. Padovani. A theory of contracts for web services. *ACM Trans. Program. Lang. Syst.*, 31(5), 2009.
5. R. De Nicola and M. Hennessy. Testing equivalences for processes. *Theor. Comput. Sci.*, 34:83–133, 1984.

6. R. De Nicola and M. Hennessy. CCS without tau's. In *TAPSOFT, Vol.1*, volume 249 of *Lect. Notes in Comput. Sci.*, pages 138–152. Springer, 1987.
7. K. Honda, V. T. Vasconcelos, and M. Kubo. Language primitives and type discipline for structured communication-based programming. In *ESOP*, volume 1381 of *Lect. Notes in Comput. Sci.*, pages 122–138. Springer Verlag, 1998.
8. C. Laneve and L. Padovani. The Must preorder revisited. In *CONCUR*, volume 4703 of *Lect. Notes in Comput. Sci.*, pages 212–225, 2007.
9. R. Milner. *Communication and Concurrency*. Prentice Hall International, 1989.
10. K. Schmidt. Controllability of open workflow nets. In *EMISA*, volume 75 of *LNI*, pages 236–249. GI, 2005.
11. K. Takeuchi, K. Honda, and M. Kubo. An interaction-based language and its typing system. In *PARLE*, volume 817 of *Lect. Notes in Comput. Sci.*, pages 398–413. Springer Verlag, 1994.