DOI 10.1007/s00165-016-0358-2
BCS © 2016
Formal Aspects of Computing (2016) 28: 697–722

Formal Aspects of Computing

Reversible client/server interactions

Franco Barbanera[1], Mariangiola Dezani-Ciancaglini[2] and Ugo de'Liguoro[2]

[1] Dipartimento di Matematica e Informatica, Università di Catania, Viale A. Doria 6, 95125 Catania, Italy
[2] Università di Torino, Torino, Italy

Abstract. In the setting of *session behaviours*, we study an extension of the concept of compliance when a disciplined form of backtracking and of output skipping is present. After adding checkpoints to the syntax of session behaviours, we formalise the operational semantics via an LTS, and define natural notions of *checkpoint compliance* and *sub-behaviour*, which we prove to be both decidable. Then we extend the operational semantics with *skips* and we show the decidability of the obtained compliance.

Keywords: Client/server interaction, Session types, Behavioural semantics, Sub-behaviour, Semantics of subtyping, Coinduction.

1. Introduction

In human as well as automatic negotiations, an interesting feature is the ability of rolling back to some previous points of a conversation, undoing choices and possibly trying different paths. *Rollbacks* are familiar to the users of web browsers, and so are also the troubles that these might cause during "undisciplined" interactions. Clicking the "back" button, or going to some previous points in the chronology when we are in the middle of a transaction, say the booking of a flight, can be as smart as dangerous. In any case it is surely a behaviour that service programmers want to discipline. Also on the server's side one has to take care: a server discovering that a service becomes available after having started a conversation could take advantage from some kind of rollback. However, such a server would be quite unfair if the rollbacks were completely hidden from the client. In this scenario it is also useful to allow the skipping of already done outputs, like in the case of a logged client, who can avoid to send again the password.

Adding rollback and skip to interaction protocols requires sophisticated concepts of client/server compliance and sub-behaviour. In this paper we investigate protocols admitting a simple, though non trivial form of reversibility and skipping in the framework of the theory of contracts introduced in [CCLP06] and developed in a series of papers, see [CGP09] and the references there. We focus here on the scenario of client/server architectures, where services stored in a repository are queried by clients to establish two-sided communications.

Correspondence and offprint requests to: F. Barbanera. e-mail: barba@dmi.unict.it
This work was partially supported by ICT COST Action IC1201 BETTY, MIUR PRIN Project CINA Prot. 2010LHT4KM and Torino University/Compagnia San Paolo Project SALT.

More precisely, we consider the formalism of *session behaviours* as introduced in [BdL15] and [BH15] (where they were referred to as *session contracts*). This is a formalism interpreting the session types, introduced by Honda et al. in [HVK98], into a subset of CCS without τ. We extend the session behaviour syntax by means of markers that we call *checkpoints*; these are intended as pointers to the last place where either the client or the server can rollback at any time. We then extend the formalism by distinguishing already done outputs (by barring them): they are outputs which do not need to be resent, i.e. they can be *skipped*. First we investigate which constraints must be imposed to obtain a safe notion of client/server interaction in the scenario without skips, by defining a model in the form of an LTS, and by characterising the resulting concepts of compliance and sub-behaviour, which we prove to be both decidable. Then we add skipping and we prove decidability of the extended notion of compliance.

Before entering into the formal development of session behaviours with checkpoints and skips, we illustrate the basic concepts by discussing a few examples. A registered Client logs in to an online Server and then she asks the prices of either a bag or a belt. Let the action (input) login represent the receipt of login credentials, and the action price the receipt of a price. Dual actions represent offers, e.g. the coaction (output) $\overline{\mathsf{login}}$ represents the sending of credentials. Then the client behaviour is described by the process:

$$\mathsf{Client} = \overline{\mathsf{login}}.(\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price})$$

where dot is sequential composition and $\oplus$ is internal choice. In the standard contract formalism we say that a client ρ is *compliant* with a server σ, written $\rho \dashv\!\shortmid \sigma$, if all client communication actions are matched by the dual actions on the server side. According to this, the client will be not compliant with an online server behaving as:

$$\mathsf{Server} = \mathsf{login}.(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{suitcase}.\overline{\mathsf{price}})$$

where + is external choice. In fact the interaction represented by the parallel composition Client ∥ Server, that evolves by synchronising corresponding actions and coactions, might lead to:

$$(\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price}) \parallel (\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{suitcase}.\overline{\mathsf{price}})$$

This means that the client cannot ask the price of a belt.

Now consider the *dual* behaviour of ρ, dubbed $\overline{\rho}$, which is obtained by exchanging actions by the respective coactions, and internal by external choices. Then the dual of Client is:

$$\mathsf{Server}' = \mathsf{login}.(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{belt}.\overline{\mathsf{price}})$$

and clearly we get $\mathsf{Client} \dashv\!\shortmid \mathsf{Server}'$. In fact with the standard contract formalism we have, as expected, that $\rho \dashv\!\shortmid \overline{\rho}$, or equivalently that $\overline{\sigma} \dashv\!\shortmid \sigma$, since duality is involutive.

Taking a further step, let us consider a client that, after asking the price of a bag, wants to ask the price of a belt. This can be achieved by rolling back to the choice $\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price}$. Rollback is a feature present in some programming languages and in models of distributed computations as well, but in our context it is actually a new feature, that cannot be easily represented by usual process algebra operations [PU07]. In particular, the intriguing possibility of representing backtracking by means of recursion will be discussed in Remark 2.9, where it will be shown not to hold.

To express rollback we introduce the symbol '$\blacktriangle$' to mark the point where a session behaviour can backtrack to; we call such a marker a *checkpoint*. We suppose that a suitable mechanism keeps memory of the past, by recording the behaviour $\blacktriangle\sigma$ each time the checkpoint is traversed, namely when $\blacktriangle\sigma$ synchronises on some actions that σ is ready to do. For simplicity we assume that only one "past" can be recorded at any time, so that a new memorisation destroys the old one, leading to a model in which the client and the server can backtrack just to the lastly traversed checkpoint. This restriction is partially justified by the absence of interesting examples requiring rollbacks to other checkpoints.

Going on with our example, by adding some checkpoints to Client we get:

$$\mathsf{Client}' = \blacktriangle\overline{\mathsf{login}}.\blacktriangle(\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price})$$

The new client can undo most of the actions and choices, in order to keep the negotiation open as much as possible. But how should the server be redesigned to properly interact with it? Unfortunately, the apparently natural choice of taking the dual of Client′

$$\mathsf{Server}'' = {}_{\blacktriangle}\mathsf{login}.{}_{\blacktriangle}(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{belt}.\overline{\mathsf{price}})$$

fails. In fact, writing $\xrightarrow{\text{fw}}$ for the forward evolution step of a client/server system, and $\xrightarrow{\text{rollbk}}$ for the synchronous rollback, we have, among the possible interactions between Client' and Server'':

$$\begin{aligned}
{}_{\blacktriangle}\overline{\mathsf{login}}.{}_{\blacktriangle}(\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price}) \;\|\; {}_{\blacktriangle}\mathsf{login}.{}_{\blacktriangle}(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{belt}.\overline{\mathsf{price}}) &\xrightarrow{\text{fw}} \text{synchronising on } \mathsf{login} \text{ and } \overline{\mathsf{login}} \\
{}_{\blacktriangle}(\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price}) \;\|\; {}_{\blacktriangle}(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{belt}.\overline{\mathsf{price}}) &\xrightarrow{\text{fw}} \text{internal choice} \\
\overline{\mathsf{bag}}.\mathsf{price} \;\|\; {}_{\blacktriangle}(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{belt}.\overline{\mathsf{price}}) &\xrightarrow{\text{rollbk}} \text{rollback to the last traversed } {}_{\blacktriangle} \\
{}_{\blacktriangle}(\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price}) \;\|\; {}_{\blacktriangle}\mathsf{login}.{}_{\blacktriangle}(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{belt}.\overline{\mathsf{price}})
\end{aligned}$$

which is now in a stuck state.

The mismatch between external and internal choice is the effect of the asymmetry of the respective semantics in process algebra. The selection of a branch in an external choice is just one step; on the contrary the possible synchronisation on $\overline{\mathsf{bag}}$ comes *after* the internal choice has occurred. This has consequences with respect to the backtracking, since the checkpoint alignment fails.

In [BdL15] it has been proved that the dual of a server is the minimum client that complies with the server with respect to a natural (and efficiently decidable) ordering, and vice versa the dual of a client is the minimum compliant server. This is an essential feature of the theory, since it is supposed to model a scenario in which clients look for servers through a network, querying a service of a certain shape that is easier to find if we know its minimal form. To express this precisely, let us write $\rho \dashv\!\|^{\blacktriangle} \sigma$ to denote the compliance of ρ with σ in a setting with backtracking, where $\rho \dashv\!\|^{\blacktriangle} \sigma$ if ρ is compliant with σ in the standard sense and keeps being so also after any possible rollback. Then we put the requirement that in the new theory the following holds:

$$\forall\, \rho. \quad \rho \dashv\!\|^{\blacktriangle} \overline{\rho} \qquad \text{COMPLIANCE OF DUALS}$$

For this condition to hold we change the operational semantics of $\oplus$ by gluing the choice and the synchronisation over a coaction, that can be formalised by the rule:

$$\overline{a}.\sigma_1 \oplus \sigma_2 \xrightarrow{\overline{a}} \sigma_1$$

This has however the unpleasant consequence that $a \dashv\!\|^{\blacktriangle} \overline{a} \oplus \overline{b}$, since $a \parallel \overline{a} \oplus \overline{b}$ only reduces to $\mathbf{1} \parallel \mathbf{1}$. Notice that $a \not\dashv\!\| \overline{a} \oplus \overline{b}$, when the compliance $\dashv\!\|$ is defined according to the standard LTS [BdL15, BH15]. In general, we expect the compliance of behaviours with rollback to be conservative with respect to the compliance without rollback. More precisely we require:

$$\forall\, \rho, \sigma. \quad \rho \dashv\!\|^{\blacktriangle} \sigma \text{ implies } \mathit{erase}(\rho) \dashv\!\| \mathit{erase}(\sigma) \qquad \text{CONSERVATIVITY OF ERASURE}$$

where *erase* deletes all checkpoints. We will accomplish this by asking that any coaction has a corresponding action in reducing the parallel of internal and external choices.

The essence of this change is that rolling back has to be a synchronous action, and therefore it cannot be the effect of an internal choice, since the latter is unobservable. This is a general principle. Consider the interaction

$${}_{\blacktriangle}\overline{\mathsf{belt}}.\mathsf{price}.\overline{\mathsf{bag}}.\mathsf{price} \parallel {}_{\blacktriangle}\mathsf{belt}.{}_{\blacktriangle}\overline{\mathsf{price}}.\mathsf{bag}.\overline{\mathsf{price}} \tag{1}$$

It is the pair of a client willing to know the price of a belt *and* the price of a bag, and a server that can succeed by sending twice the price of the belt! The point is that the client has no way to be aware of what happened and to react according to her own policy, which is instead the case if both are forced to backtrack at the same time. For this to be guaranteed we require that the client and the server either both can or both cannot rollback in all configurations. In particular we assume that in the starting configurations both session behaviours of client and server are checkpointed.

We also observe that it is not necessarily the case that compliant behaviours show some correspondence between the respective checkpoints. For example it holds that:

$${}_{\blacktriangle}(\overline{\mathsf{discount}}.{}_{\blacktriangle}\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{bag}}.\mathsf{price}) \;\dashv\!\|^{\blacktriangle}\; {}_{\blacktriangle}(\mathsf{discount}.\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{bag}.\overline{\mathsf{price}}) \tag{2}$$

which makes sense, since the client ${}_{\blacktriangle}(\overline{\mathsf{discount}}.{}_{\blacktriangle}\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{bag}}.\mathsf{price})$ is asking for the price of a bag, with or without discount.

Notice that, if we use the operational semantics informally described above, we must rule out terms like ${}_{\blacktriangle}\overline{a}_1.\sigma_1 \oplus \overline{a}_2.\sigma_2$, where a single branch of a choice among two branches is checkpointed. The motivation can be illustrated by an example; let us consider the following behaviour with checkpoints:

${}_{\blacktriangle}e.({}_{\blacktriangle}\overline{a}.b \oplus \overline{c}.d)$.

In the present setting we are assuming that it is possible to rollback just to the last encountered checkpoint, and that restoring a "past" doesn't recover older checkpoints. This implies that all the informations for a rollback are "memorised" only until a new checkpoint is crossed. Then, after the action e, the side effect of choosing ${}_{\blacktriangle}\overline{a}.b$ should be that of memorising the latter as the "past", deleting ${}_{\blacktriangle}e.({}_{\blacktriangle}\overline{a}.b \oplus \overline{c}.d)$ and definitely discarding $\overline{c}.d$; on the other hand the choice of the latter branch would save the past ${}_{\blacktriangle}e.({}_{\blacktriangle}\overline{a}.b \oplus \overline{c}.d)$, to which we return in case of rollback. This has the undesirable effect that the actual past depends on an internal choice, which is hidden from the partner. Besides, the actual meaning of rolling back after the choice of ${}_{\blacktriangle}\overline{a}.b$ is not the undo of the choice, but to insist in doing the very same choice. This makes the rollback useless. Observe that we are not preventing a choice to be repeated after rollback: when restoring a past of the shape ${}_{\blacktriangle}\bigoplus_{i\in I}\overline{a}_i.\sigma_i$, there is no record of the branch previously taken. But whenever a rollback occurs to such a past, we ask that *all* choices are newly allowed. This is why we decided to rule out terms like ${}_{\blacktriangle}\overline{a}_1.\sigma_1 \oplus \overline{a}_2.\sigma_2$, imposing prefixes of summands of internal (and external) choices not to be checkpointed.

We discuss now the addition of skips, inspired by some previous work reported in [BdL14]. Let us consider the following pair of client and server:

$$\begin{aligned} \mathsf{Client}'' &=_{\blacktriangle} \overline{\mathsf{login}}.(\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price}) \\ \mathsf{Server}''' &=_{\blacktriangle} \mathsf{login}.{}_{\blacktriangle}(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{belt}.\overline{\mathsf{price}}) \end{aligned} \tag{3}$$

The interaction among them gets stuck with respect to the semantics discussed so far, since the (non well-designed) client Client'' is self-inflicting a new login whenever undoing the choice among a bag and a belt, while the server would normally save data from the former identification, as expressed by the checkpoints in Server'''. Such mismatches make the compliance relation we study exceedingly demanding.

In order to avoid such restrictions we relax the requirements defining compliance, and we consider a calculus where it is allowed to *skip* an already done output; for example the $\overline{\mathsf{login}}$ in the case above. We mark by a bar the outputs which have been already done and that may be skipped. Doing that enables Client'' and Server''' to be compliant.

On the other hand we observe that it would be unreasonable to skip outputs when the corresponding inputs are available, as this would result into a complete loss of control. For example we would have:

$$\begin{aligned} \mathsf{Client}' \parallel \mathsf{Server}'' &\xrightarrow{\mathsf{fw}} {}_{\blacktriangle}(\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price}) \parallel {}_{\blacktriangle}(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{belt}.\overline{\mathsf{price}}) \\ &\xrightarrow{\mathsf{rollbk}} {}_{\blacktriangle}\overline{\cancel{\mathsf{login}}}.{}_{\blacktriangle}(\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price}) \parallel {}_{\blacktriangle}\mathsf{login}.{}_{\blacktriangle}(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{belt}.\overline{\mathsf{price}}) \\ &\xrightarrow{\mathsf{skip}} {}_{\blacktriangle}(\overline{\mathsf{bag}}.\mathsf{price} \oplus \overline{\mathsf{belt}}.\mathsf{price}) \parallel {}_{\blacktriangle}\mathsf{login}.{}_{\blacktriangle}(\mathsf{bag}.\overline{\mathsf{price}} + \mathsf{belt}.\overline{\mathsf{price}}) \end{aligned}$$

In order not to get stuck, we should skip the input action login but, as discussed in [BdL14], allowing to skip inputs is unnatural.
We show how, although with some overhead, results that we establish about the calculus with rollback can be extended to the case with rollback and skips.

Outline In Sect. 2 we introduce the calculus with checkpoints and its operational semantics. In Sect. 3 we define compliance and we show its decidability by means of a syntax-directed formal system, which is proved to be sound and complete. The compliance relation induces a decidable sub-behaviour relation between servers which is the basis of query engines: this is the content of Sect. 4. In Sect. 5 and Sect. 6 we present the calculus with skips and we prove decidability of compliance in the new setting. Lastly Sect. 7 points to related works and some future developments.

This paper is an extended and revised version of [BDdL14] where we only gave a hint to the soundness and completeness proofs and we didn't consider neither the sub-behaviour relation nor the calculus with skips.

$$
\begin{array}{lcll}
\sigma, \rho & := & \mathbf{1} & \text{success} \\
& | & \sum_{i\in I} a_i.\sigma_i & \text{external choice} \\
& | & {}_{\blacktriangle}\sum_{i\in I} a_i.\sigma_i & \text{checkpointed external choice} \\
& | & \bigoplus_{i\in I} \overline{a}_i.\sigma_i & \text{internal choice} \\
& | & {}_{\blacktriangle}\bigoplus_{i\in I} \overline{a}_i.\sigma_i & \text{checkpointed internal choice} \\
& | & x & \text{variable} \\
& | & \mathsf{rec}\, x.\sigma & \text{recursion}
\end{array}
$$

Fig. 1. Syntax of raw behaviour expressions with checkpoints

2. Calculus

In this section we introduce the syntax of *session behaviours with checkpoints* and we describe their operational semantics by an LTS, as usual with process algebras. The calculus is obtained from *session behaviours*, as treated in [BdL15, BH15], by adding checkpoints. As explained in the Introduction, we allow checkpoints to occur only before internal or external choices, not before branches unless the sums consist of just one branch.

Definition 2.1 (*Session behaviours with checkpoints*)

(i) Let $\mathcal{N}$ (*names*) be some countable set of symbols and $\overline{\mathcal{N}} = \{\overline{a} \mid a \in \mathcal{N}\}$ (*conames*), with $\mathcal{N} \cap \overline{\mathcal{N}} = \emptyset$. The set $\mathcal{BE}$ of *raw session behaviours with checkpoints* is defined by the grammar of Fig. 1, where

- the set I is non-empty and finite;
- the names and the conames in choices are pairwise distinct;
- σ is not a variable in $\mathsf{rec}\, x.\sigma$.

(ii) The set SB of *session behaviours with checkpoints* is defined as the restriction of $\mathcal{BE}$ to closed expressions.

Notice that the restrictions in the previous definition (namely of external and internal choices to be made of summands with pairwise distinct prefixes, that are all input actions in external choices, and all output actions in internal choices) are present also in [BdL15, BH15], where they are justified. They are mainly due to the aim of using results on session behaviours in contexts where processes interactions are structured by means of session types [HVK98].
We remark that in our theory (as well in the theory of session behaviours in general [BdL15, BH15]), only closed terms are considered behaviours, since there is no real practical use for open terms.

When I is a singleton set, we just write $a.\sigma$ and $\overline{a}.\sigma$ for $\sum a.\sigma$ and $\bigoplus \overline{a}.\sigma$; also we omit the trailing $\mathbf{1}$ whenever possible.

Recursion in SB is guarded and hence contractive in the usual sense. We take the equi-recursive view of recursion, by equating $\mathsf{rec}\, x.\sigma$ with $\sigma[\mathsf{rec}\, x.\sigma/x]$. Hence there is no point in considering terms of the shape ${}_{\blacktriangle}\mathsf{rec}\, x.\sigma$.

In the following the notation ${}_{\triangle}\sigma$ will represent ambiguously σ and ${}_{\blacktriangle}\sigma$.

We use α to range over $\mathcal{N} \cup \overline{\mathcal{N}}$, with the convention $\overline{\alpha} = \begin{cases} \overline{a} & \text{if } \alpha = a, \\ a & \text{if } \alpha = \overline{a}. \end{cases}$

When no ambiguity can arise, we shall refer to *session behaviours with checkpoints* as simply *session behaviours* or *behaviours*.

A syntactical notion of *duality* on SB is easily obtained by extending the usual duality for contracts in such a way it leaves the checkpoints unchanged. Being such a notion formally defined by induction on the structure of (possibly open) expressions, we first define it on $\mathcal{BE}$. Duality for elements in SB is then inherited by restricting it to SB.

Definition 2.2 (*Duality*)

(i) Let $\sigma \in \mathcal{BE}$. The syntactic dual $\overline{\sigma}$ of σ is defined by the following clauses:

$$\overline{\mathbf{1}} = \mathbf{1} \qquad \overline{x} = x \qquad \overline{\mathsf{rec}\, x.\sigma} = \mathsf{rec}\, x.\overline{\sigma} \qquad \overline{{}_{\blacktriangle}\sigma} = {}_{\blacktriangle}\overline{\sigma}$$

$$\overline{\sum_{i\in I} a_i.\sigma_i} = \bigoplus_{i\in I} \overline{a}_i.\overline{\sigma_i} \qquad \overline{\bigoplus_{i\in I} \overline{a}_i.\sigma_i} = \sum_{i\in I} a_i.\overline{\sigma_i}$$

(ii) Let $\overline{(\cdot)} : \mathsf{SB} \to \mathsf{SB}$ be the restriction to SB of the duality function.

Notice that item (ii) in the previous definition is sound, since $\overline{\sigma} \in \mathsf{SB}$ if and only if $\sigma \in \mathsf{SB}$.

From now on, to avoid cumbersome definitions, any time an inductive definition on elements of SB is provided, it will be tacitly assumed to be the restriction to SB of the corresponding inductive definition on $\mathcal{BE}$.

Definition 2.2 closely mimics the duality operator on session types as defined e.g. in [HVK98]. As expected, $\overline{\overline{\sigma}} = \sigma$ for all σ. We remark that checkpoints are unaffected by the $\overline{\cdot}$ operation.

2.1. Operational semantics

To correctly define the operational semantics of the calculus, we have to record the checkpoint-prefixed behaviour whose checkpoint is the last one that has been *crossed* during the evolution of a behaviour. Therefore we introduce *configurations*, that is behaviours (σ) with pasts (γ) of the shape:

$$\gamma \prec \sigma$$

where γ is a checkpointed internal or external choice.

In the starting configuration, or just after a rollback has occurred, there is no further point to which the behaviour might roll back. We represent such a situation by writing $\circ \prec \sigma$, where '$\circ$' marks the fact that no-rollback is possible.

Definition 2.3 (*Configurations*)

(i) Let $\mathsf{SB}^{\blacktriangle}$ be the set of behaviours starting with ${}_{\blacktriangle}$, then $\mathsf{SB}^{\blacktriangle} \cup \{\circ\}$ is the set of the *pasts*, and we denote its elements by γ, δ, possibly with superscripts.

(ii) The set $\mathsf{SB}_{\prec}$ of *configurations* is defined by
$\mathsf{SB}_{\prec} = \{\gamma \prec \sigma \mid \gamma \in \mathsf{SB}^{\blacktriangle} \cup \{\circ\}, \sigma \in \mathsf{SB}\}$.

We are now in place to define the LTS of configurations. We do not need to consider recursive behaviours, since they are equated to non recursive ones in the equi-recursive view.

Definition 2.4 (*LTS for configurations*)

$$\gamma \prec \sum_{i\in I} a_i.\sigma_i \xrightarrow{a_k} \gamma \prec \sigma_k \ \ (k \in I)\ (+) \qquad \gamma \prec \bigoplus_{i\in I} \overline{a}_i.\sigma_i \xrightarrow{\overline{a}_k} \gamma \prec \sigma_k \ \ (k \in I)\ (\oplus)$$

$$\frac{\gamma \prec \sigma \xrightarrow{\alpha} \gamma \prec \sigma' \quad \alpha \in \mathcal{N} \cup \overline{\mathcal{N}}}{\gamma \prec {}_{\blacktriangle}\sigma \xrightarrow{\alpha} {}_{\blacktriangle}\sigma \prec \sigma'}\ (\blacktriangle) \qquad \gamma \prec \sigma \xrightarrow{\mathsf{rbk}} \circ \prec \gamma \ \ (\gamma \neq \circ) \qquad (\mathsf{rbk})$$

Notice that the rule for internal choice glues into just one step both the internal choice and the communication of a coname, becoming very similar to the rule for external choice. The reduction of client/server parallel compositions (Definition 2.6 below) will be only possible when all internal choices can be matched by the corresponding external choices, which has the effect of saving the CONSERVATIVITY OF ERASURE discussed in the Introduction. If the current behaviour has no checkpoint, the past (in either cases, $\circ$ or an element of $\mathsf{SB}^{\blacktriangle}$) is unaffected by the choice of a branch (rules ($\oplus$) and (+)).

Rule ($\blacktriangle$) says that, in the presence of a checkpoint, the forward reduction must update the behaviour at which it is possible to rollback, namely the past. The rollback action is implemented by Rule (rbk) and it is enabled only in case there is a past to roll back to, that is when $\gamma \neq \circ$. The rollback action updates the past to $\circ$, hence no further rollback is allowed unless after traversing a new checkpoint.

When composing in parallel clients and servers we have to consider the different nature of the reductions for internal and external choices. To this aim it is handy to collect the sets of names and conames prefixing the choices, as done in the following definition. Notice that the resulting sets only contain names, since each coname is mapped to the corresponding name.

Definition 2.5 (*The functions $\mathcal{A}^+(\cdot)$ and $\mathcal{A}^\oplus(\cdot)$*)
We define $\mathcal{A}^+(\cdot), \mathcal{A}^\oplus(\cdot) : \mathsf{SB} \to \mathcal{P}(\mathcal{N})$ by

$$\mathcal{A}^+(\mathbf{1}) = \mathcal{A}^+(\textstyle\bigoplus_{i\in I} \overline{a}_i.\sigma_i) = \emptyset \qquad \mathcal{A}^+(\textstyle\sum_{i\in I} a_i.\sigma_i) = \{a_i \mid i \in I\} \qquad \mathcal{A}^+({}_\blacktriangle\sigma) = \mathcal{A}^+(\sigma)$$
and
$$\mathcal{A}^\oplus(\mathbf{1}) = \mathcal{A}^\oplus(\textstyle\sum_{i\in I} a_i.\sigma_i) = \emptyset \qquad \mathcal{A}^\oplus(\textstyle\bigoplus_{i\in I} \overline{a}_i.\sigma_i) = \{a_i \mid i \in I\} \qquad \mathcal{A}^\oplus({}_\blacktriangle\sigma) = \mathcal{A}^\oplus(\sigma)$$

The interaction of a client with a server is modeled by the reduction of their parallel composition, that can be:

- Forward ($\xrightarrow{\tau}$), consisting of CCS style synchronisations, when the set of offered outputs is included in that of offered inputs (condition $\mathcal{A}^\oplus(\ ) \subseteq \mathcal{A}^+(\ )$), or
- Backward ($\xrightarrow{\mathsf{rbk}}$), where both behaviours synchronously go back to the respective last traversed checkpointed behaviours, if both have such.

Definition 2.6 (*Reduction of client and server pairs*)

$$\frac{\delta \prec \rho \xrightarrow{a} \delta' \prec \rho' \qquad \gamma \prec \sigma \xrightarrow{\overline{a}} \gamma' \prec \sigma' \qquad \mathcal{A}^\oplus(\sigma) \subseteq \mathcal{A}^+(\rho)}{\delta \prec \rho \parallel \gamma \prec \sigma \xrightarrow{\tau} \delta' \prec \rho' \parallel \gamma' \prec \sigma'} (+\|\oplus)$$

$$\frac{\delta \prec \rho \xrightarrow{\overline{a}} \delta' \prec \rho' \qquad \gamma \prec \sigma \xrightarrow{a} \gamma' \prec \sigma' \qquad \mathcal{A}^\oplus(\rho) \subseteq \mathcal{A}^+(\sigma)}{\delta \prec \rho \parallel \gamma \prec \sigma \xrightarrow{\tau} \delta' \prec \rho' \parallel \gamma' \prec \sigma'} (\oplus\|+)$$

$$\frac{\delta \prec \rho \xrightarrow{\mathsf{rbk}} \circ \prec \rho' \qquad \gamma \prec \sigma \xrightarrow{\mathsf{rbk}} \circ \prec \sigma'}{\delta \prec \rho \parallel \gamma \prec \sigma \xrightarrow{\mathsf{rbk}} \circ \prec \rho' \parallel \circ \prec \sigma'} (\mathsf{rbk})$$

In the following let $\longrightarrow$ be the union of $\xrightarrow{\tau}$ and $\xrightarrow{\mathsf{rbk}}$. We denote by $\xrightarrow{\tau}{}^*$, $\xrightarrow{\mathsf{rbk}}{}^*$ and $\xrightarrow{*}$ the reflexive and transitive closure of, respectively, $\xrightarrow{\tau}$, $\xrightarrow{\mathsf{rbk}}$ and $\longrightarrow$. We also use $\longrightarrow$ without specifying the resulting configuration and $\not\longrightarrow$ with the obvious meanings.

It is easy to verify that if $\circ \prec \rho \parallel \circ \prec \sigma \xrightarrow{\tau}{}^* \circ \prec \rho' \parallel \circ \prec \sigma'$, then $\rho \parallel \sigma$ reduces to $\rho' \parallel \sigma'$ in the calculi of [BdL15, BH15], by splitting in two steps each application of rule ($\oplus$). If $\rho \parallel \sigma$ reduces to $\rho' \parallel \sigma'$ in the calculi of [BdL15, BH15] we can find ρ'', σ'' such that both $\rho' \parallel \sigma'$ reduces to $\rho'' \parallel \sigma''$ and

$$\circ \prec \rho \parallel \circ \prec \sigma \xrightarrow{\tau}{}^* \circ \prec \rho'' \parallel \circ \prec \sigma''$$

Notice that $\rho'' \parallel \sigma''$ is different from $\rho' \parallel \sigma'$ only in case the last applied rule is an internal choice, which rule ($\oplus$) fuses with the communication of the coname.

Starting from a parallel composition, in which one of the two conditions holds

- both pasts are $\circ$ and both behaviours are checkpointed,
- both pasts are checkpointed behaviours,

only parallel compositions satisfying one of the two conditions can be reached, as formalised in the following definition and proposition. This assures that the client and the server either both can or both cannot roll back in all configurations, an essential property as discussed in the Introduction.

Definition 2.7 We say that the client/server parallel composition (of configurations) $\delta \prec \rho \parallel \gamma \prec \sigma$ is *nice* if:

1. either $\delta = \gamma = \circ$ and $\rho, \sigma \in \mathsf{SB}^{\blacktriangle}$;
2. or $\delta, \gamma \in \mathsf{SB}^{\blacktriangle}$.

Notice that the behaviours are arbitrary in a nice parallel composition in which the pasts are checkpointed behaviours (condition (2) of previous definition).

Proposition 2.8 *If $\delta \prec \rho \parallel \gamma \prec \sigma$ is nice and $\delta \prec \rho \parallel \gamma \prec \sigma \xrightarrow{*} \delta' \prec \rho' \parallel \gamma' \prec \sigma'$, then $\delta' \prec \rho' \parallel \gamma' \prec \sigma'$ is nice too.*

Proof A nice parallel composition which satisfies condition (1) of Definition 2.7 can only be reduced using rules $(+\|\oplus)$ or $(\oplus\|+)$, getting a nice parallel composition which satisfies condition (2) of Definition 2.7. □

A nice parallel composition which satisfies condition (2) of Definition 2.7 can be reduced using rules $(+\|\oplus)$ or $(\oplus\|+)$, getting a nice parallel composition which satisfies again the same condition. It can also be reduced using rule (rbk), getting a nice parallel composition which satisfies condition (1) of Definition 2.7.

Thanks to the previous proposition from now on we will only consider nice parallel compositions.

Remark **2.9** By looking at the operational semantics of behaviours one can notice that checkpoints and arbitrary rollbacks influence the evolution of behaviours in a way similar to the effect of recursion. For instance, the possible evolution of the behaviour ${}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b}$ can be mimicked by the following recursive expression without checkpoints:

$$\mathsf{rec}\, x.(\overline{a}.x \oplus \overline{a}.\mathsf{rec}\, y.\overline{b}.y)$$

However observe that the same name as prefix of distinct branches of an internal choice is not allowed, so this expression is not a behaviour.

As a further example, let us consider the behaviour ${}_{\blacktriangle}a.{}_{\blacktriangle}b$. In order to use recursion to mimic the way it could evolve, we should use an expression like

$$\mathsf{rec}\, x.(a.x \oplus a.\mathsf{rec}\, y.b.y)$$

This expression, besides having two branches with the same prefix, does not comply with the other syntax constraint on session behaviours, which imposes only input prefixes in internal choices.

Even by relaxing the syntax, the notion of compliance could not be faithfully represented using recursion instead of checkpoints and arbitrary rollbacks. For example ${}_{\blacktriangle}\overline{a} \dashv\!\!\mid^{\blacktriangle} {}_{\blacktriangle}a.{}_{\blacktriangle}b$, while

$$\mathsf{rec}\, x.\overline{a}.x \not\dashv \mathsf{rec}\, x.(a.x \oplus a.\mathsf{rec}\, y.b.y)$$

since as soon as the server chooses the right branch, no client's request can be satisfied.

Lastly observe that representing checkpoints by internal choices and recursion would make impossible to distinguish between an evolution depending on an internal choice and an evolution depending on a rollback.

Another question that naturally arises is whether recursion itself could be avoided and represented in terms of checkpoints and rollbacks. The answer is negative, since the evolution of $\mathsf{rec}\, x.(\overline{a}.\overline{b}.x)$ cannot be mimicked by a checkpointed expression. In fact

$${}_{\blacktriangle}\overline{a}.\overline{b}$$

produces also the sequence of outputs $\overline{a}\,\overline{a}\,\overline{a}\,\ldots$, that the recursive expression cannot produce.

3. Compliance

The compliance relation on standard session behaviours requires that for any sequence of interactions among a client ρ and a server σ, whenever a state is reached where no further communication (namely τ-reduction) is possible, all client's requests and offers are satisfied, that is the client ρ has evolved to $\mathbf{1}$ [BdL15, BH15]. Keeping the same definition in the case of session behaviours with rollbacks leads to the following:

Definition 3.1 (*Checkpoint Compliance Relation* $\dashv\!\vert\!\vert^{\blacktriangle}$)

(i) Let $\delta \prec \rho, \gamma \prec \sigma \in \mathsf{SB}_{\prec}$. We say that $\delta \prec \rho$ is *checkpoint compliant* with $\gamma \prec \sigma$, written $\delta \prec \rho \dashv\!\vert\!\vert^{\blacktriangle} \gamma \prec \sigma$, when for all $\delta' \prec \rho'$ and $\gamma' \prec \sigma'$:

$$\text{if} \quad \delta \prec \rho \parallel \gamma \prec \sigma \xrightarrow{*} \delta' \prec \rho' \parallel \gamma' \prec \sigma' \not\xrightarrow{\tau} \quad \text{then} \quad \rho' = \mathbf{1}.$$

(ii) Let $\rho, \sigma \in \mathsf{SB}^{\blacktriangle}$. We say that ρ is *checkpoint compliant* with σ (notation $\rho \dashv\!\vert\!\vert^{\blacktriangle} \sigma$) if $\circ \prec \rho \dashv\!\vert\!\vert^{\blacktriangle} \circ \prec \sigma$.

Roughly, when $\delta \prec \rho \dashv\!\vert\!\vert^{\blacktriangle} \gamma \prec \sigma$ holds, ρ and σ are compliant in the standard sense. Our definition is more demanding in case ρ and σ include checkpoints, asking that they keep on being compliant after any possible synchronous rollback. Therefore, if both δ and γ are different from $\circ$, then it must be $\circ \prec \delta \dashv\!\vert\!\vert^{\blacktriangle} \circ \prec \gamma$. Moreover, it can never be the case that one of them can perform a rollback and the other one cannot, even when ρ is in the success configuration. Notice that, by Lemma 2.8, we can safely restrict the technical treatment to *nice* parallel compositions, where either both δ and γ are equal to $\circ$ (no rollback is possible) or they are both checkpointed behaviours (the rollback is allowed).

It is easy to verify that Definition 3.1(ii) satisfies the COMPLIANCE OF DUALS and the CONSERVATIVITY OF ERASURE discussed in the Introduction. Namely, each session behaviour is checkpoint compliant with its dual, and if a client and a server are checkpoint compliant, then the client and the server obtained by erasing the checkpoints are compliant. The last property follows from the observation that all forward reductions are preserved when no rollback is allowed. More formally, let *erase*(σ) be the result of removing all checkpoints from σ:

Proposition 3.2 (Compliance of duals and conservativity of erasure) *For any* $\rho, \sigma \in \mathsf{SB}^{\blacktriangle}$*:*

1. $\rho \dashv\!\vert\!\vert^{\blacktriangle} \overline{\rho}$.
2. *If* $\rho \dashv\!\vert\!\vert^{\blacktriangle} \sigma$*, then* *erase*$(\rho) \dashv\!\vert\!\vert$ *erase*(σ).

Instead *erase*$(\rho) \dashv\!\vert\!\vert$ *erase*(σ) does not imply $\rho \dashv\!\vert\!\vert^{\blacktriangle} \sigma$, take for example $\rho = {}_{\blacktriangle}a.b$ and $\sigma = {}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b}$.

In the following, when ambiguity cannot arise, we shall simply say *compliance*/*compliant* instead of *checkpoint compliance*/*compliant*.

We define a function $\mathbf{p}$ denoting the effect of traversing a checkpoint on the "past" of a configuration.

Definition 3.3 (*The function* $\mathbf{p}$) The function $\mathbf{p} : \mathsf{SB}^{\blacktriangle} \cup \{\circ\} \times \mathsf{SB} \to \mathsf{SB}^{\blacktriangle} \cup \{\circ\}$ is defined by

$$\mathbf{p}(\gamma, {}_{\triangle}\sigma) = \begin{cases} {}_{\blacktriangle}\sigma & \text{if} \quad \triangle = \blacktriangle \\ \gamma & \text{otherwise.} \end{cases}$$

Forward reduction in Definition 2.4 can be shortly written in terms of the function $\mathbf{p}$:

Lemma 3.4 *Let* $k \in I$.

$$\gamma \prec {}_{\triangle}(\textstyle\sum_{i\in I} a_i.\sigma_i) \xrightarrow{a_k} \mathbf{p}(\gamma, {}_{\triangle}(\textstyle\sum_{i\in I} a_i.\sigma_i)) \prec \sigma_k$$
$$\gamma \prec {}_{\triangle}(\textstyle\bigoplus_{i\in I} \overline{a}_i.\sigma_i) \xrightarrow{\overline{a}_k} \mathbf{p}(\gamma, {}_{\triangle}(\textstyle\bigoplus_{i\in I} \overline{a}_i.\sigma_i)) \prec \sigma_k$$

Proof By cases, according to whether the ${}_{\triangle}$'s are ${}_{\blacktriangle}$ or not, using Definition 3.3. □

As a first step in the study of compliance, we provide a coinductive definition of $\dashv\!\vert\!\vert^{\blacktriangle}$. Let us define a new relation $\dashv\!\vert\!\vert^{\blacktriangle}_{co}$ as follows.

Definition 3.5 Let $\{\dashv\!\vert\!\vert^{\blacktriangle}_k \mid k \in \mathbb{N}\}$ be the family of relations over $\mathsf{SB}_{\prec}$ such that $\dashv\!\vert\!\vert^{\blacktriangle}_0 = \mathsf{SB}_{\prec} \times \mathsf{SB}_{\prec}$ and $\delta \prec \rho \dashv\!\vert\!\vert^{\blacktriangle}_{k+1} \gamma \prec \sigma$ if either:

1. $\rho = \mathbf{1}$ and $\delta = \gamma = \circ$; or
2. $\rho = \mathbf{1}$ and $\delta, \gamma \neq \circ$ and $\circ \prec \delta \dashv\!\vert\!\vert^{\blacktriangle}_k \circ \prec \gamma$; or
3. $\rho \neq \mathbf{1}$ and $\delta \prec \rho \parallel \gamma \prec \sigma \xrightarrow{\tau}$ and

$$[\delta \prec \rho \parallel \gamma \prec \sigma \longrightarrow \delta' \prec \rho' \parallel \gamma' \prec \sigma'] \text{ implies } \delta' \prec \rho' \dashv\!\vert\!\vert^{\blacktriangle}_k \gamma' \prec \sigma' \text{ for all } \delta' \prec \rho', \gamma' \prec \sigma'.$$

Then we define $\dashv\!\vert\!\vert^{\blacktriangle}_{co} = \bigcap_{k\in\mathbb{N}} \dashv\!\vert\!\vert^{\blacktriangle}_k$.

Lemma 3.6 *The relation* $\dashv\!\Vdash^{\blacktriangle}_{co}$ *and the compliance relation* $\dashv\!\Vdash^{\blacktriangle}$ *coincide.*

Proof The inclusion $\dashv\!\Vdash^{\blacktriangle}_{co} \subseteq \dashv\!\Vdash^{\blacktriangle}$ is immediate. Vice versa let k be the minimal natural number such that $\delta \prec \rho \not\dashv\!\Vdash^{\blacktriangle}_{k} \gamma \prec \sigma$. Then there is a reduction

$$\delta \prec \rho \parallel \gamma \prec \sigma \longrightarrow \delta_1 \prec \rho_1 \parallel \gamma_1 \prec \sigma_1 \longrightarrow \cdots \longrightarrow \delta_{k-1} \prec \rho_{k-1} \parallel \gamma_{k-1} \prec \sigma_{k-1}$$

of length $k-1$ such that $\delta_i \prec \rho_i \parallel \gamma_i \prec \sigma_i \xrightarrow{\tau}$ for all $i < k-1$ (but note that not necessarily $\delta_i \prec \rho_i \parallel \gamma_i \prec \sigma_i \xrightarrow{\tau} \delta_{i+1} \prec \rho_{i+1} \parallel \gamma_{i+1} \prec \sigma_{i+1}$) and $\delta_{k-1} \prec \rho_{k-1} \not\dashv\!\Vdash^{\blacktriangle}_{1} \gamma_{k-1} \prec \sigma_{k-1}$. Therefore $\rho_{k-1} \neq \mathbf{1}$ and $\delta_{k-1} \prec \rho_{k-1} \parallel \gamma_{k-1} \prec \sigma_{k-1} \not\xrightarrow{\tau}$, which implies $\delta \prec \rho \not\dashv\!\Vdash^{\blacktriangle} \gamma \prec \sigma$. □

Next we define a formal system that we shall prove to axiomatically characterise the checkpoint compliance relation. The system is inspired by the coinductive axiomatisation of subtyping of the arrow and recursive-types in [BH98].

Definition 3.7 (*The formal system* $\rhd$) The judgment of the formal system $\rhd$ are expressions of the form $\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$, where Γ is an environment, i.e. a finite set $\Gamma = \{\delta_i \prec \rho_i \dashv^{\blacktriangle} \gamma_i \prec \sigma_i\}_{i \in I}$.
The rules of the formal system are given in Fig. 2, where in writing $\gamma \prec \delta$ we assume $\delta \in \mathsf{SB}$ (hence $\delta \neq \circ$). The symbol $\dashv^{\blacktriangle}$ is used to denote the formal counterpart of $\dashv\!\Vdash^{\blacktriangle}$.

As usual we write $\rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ for $\emptyset \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$. In Fig. 2 $\Gamma, \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ is short for $\Gamma \cup \{\delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma\}$. Moreover all rules but (Hyp) are in fact two, for example in case of rule (Ax) we have:

$$\frac{}{\Gamma \rhd \circ \prec \mathbf{1} \dashv^{\blacktriangle} \circ \prec \sigma}\ (\mathrm{Ax}_1) \qquad \frac{\Gamma \rhd \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma}{\Gamma \rhd \delta \prec \mathbf{1} \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\mathrm{Ax}_2)$$

Observe that, in case of rule (Ax_1), no rollback is possible on both sides, so $\rho = \mathbf{1}$ suffices to conclude that ρ is compliant with σ. On the contrary in case of rule (Ax_2) we have also to check that the "pasts" δ and γ are compliant, since a rollback might occur. Note that, by restricting to nice parallel compositions, if $\delta \neq \circ$, then also $\gamma \neq \circ$ and vice versa. A similar remark applies to the other rules, where the use of the notation with **p** helps to treat shortly the various cases with either checkpointed or non checkpointed behaviours.

Example **3.8** The following is a derivation of the judgment $\rhd {}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b} \dashv^{\blacktriangle} {}_{\blacktriangle}(a.b + b)$, where
$\Gamma_1 = \{\circ \prec {}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b} \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}(a.b + b)\}$,
$\Gamma_2 = \Gamma_1,\ {}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b} \prec {}_{\blacktriangle}\overline{b} \dashv^{\blacktriangle} {}_{\blacktriangle}(a.b + b) \prec b$,
$\Gamma_3 = \Gamma_2,\ \circ \prec {}_{\blacktriangle}\overline{b} \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}(a.b + b)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_3 \rhd \circ \prec {}_{\blacktriangle}\overline{b} \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}(a.b + b)}\,(\mathrm{Hyp})}{\Gamma_3 \rhd {}_{\blacktriangle}\overline{b} \prec \mathbf{1} \dashv^{\blacktriangle} {}_{\blacktriangle}(a.b + b) \prec \mathbf{1}}\,(\mathrm{Ax})}{\Gamma_2 \rhd \circ \prec {}_{\blacktriangle}\overline{b} \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}(a.b + b)}\,(\oplus\cdot+)}{\Gamma_2 \rhd {}_{\blacktriangle}\overline{b} \prec \mathbf{1} \dashv^{\blacktriangle} {}_{\blacktriangle}(a.b + b) \prec \mathbf{1}}\,(\mathrm{Ax}) \qquad \dfrac{}{\Gamma_2 \rhd \circ \prec {}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b} \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}(a.b + b)}\,(\mathrm{Hyp})}{\Gamma_1 \rhd {}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b} \prec {}_{\blacktriangle}\overline{b} \dashv^{\blacktriangle} {}_{\blacktriangle}(a.b + b) \prec b}\,(\oplus\cdot+)}{\rhd \circ \prec {}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b} \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}(a.b + b)}\,(\oplus\cdot+)$$

By the Soundness property we shall prove below, such a derivation implies that ${}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b} \dashv\!\Vdash^{\blacktriangle} {}_{\blacktriangle}(a.b + b)$.

Remark **3.9** In case of derivations with checkpointed behaviours in the conclusions (as it is our case, since the behaviours we consider result from reducing nice parallel compositions), the axiom (Ax) with $\delta = \gamma = \circ$, that is (Ax_1), is actually never applied, and the leaves of a derivation are always instances of the axiom (Hyp). In fact in any derivation with height greater than 0 with a (Ax_1) leaf, the judgments in the path from the (Ax_1) leaf to the conclusion have all the pasts equal to $\circ$ and the behaviours in the conclusions are not checkpointed.

The intuition is that recursive behaviours show an evolution that somewhat resembles that of checkpointed behaviours, even if, as discussed in Remark 2.9, the former ones cannot actually been defined in terms of the latter ones, and vice versa. In the formal systems for behaviour compliance, as those in [BdL15, BDLdL16], and those dealing with recursive types in general [BH98], axioms like (Hyp) are used to deal with recursion only.

$$\frac{\text{either } \delta = \gamma = \circ \text{ or } \Gamma \rhd \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma}{\Gamma \rhd \delta \prec \mathbf{1} \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\textsc{Ax}) \qquad \frac{}{\Gamma, \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\textsc{Hyp})$$

$$\frac{\forall j \in J.\ \Gamma' \rhd \mathbf{p}(\delta, \rho) \prec \rho_j \dashv^{\blacktriangle} \mathbf{p}(\gamma, \sigma) \prec \sigma_j \qquad \text{either } \delta = \gamma = \circ \text{ or } \Gamma' \rhd \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma}{\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (+ \cdot \oplus)$$

where $\Gamma' = \Gamma,\ \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ and $\rho = {}_{\triangle 1}(\sum_{i \in I \cup J} a_i.\rho_i)$ and $\sigma = {}_{\triangle 2}(\bigoplus_{j \in J} \overline{a}_j.\sigma_j)$

$$\frac{\forall i \in I.\ \Gamma' \rhd \mathbf{p}(\delta, \rho) \prec \rho_i \dashv^{\blacktriangle} \mathbf{p}(\gamma, \sigma) \prec \sigma_i \qquad \text{either } \delta = \gamma = \circ \text{ or } \Gamma' \rhd \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma}{\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\oplus \cdot +)$$

where $\Gamma' = \Gamma,\ \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ and $\rho = {}_{\triangle 1}(\bigoplus_{i \in I} \overline{a}_i.\rho_i)$ and $\sigma = {}_{\triangle 2}(\sum_{j \in I \cup J} a_j.\sigma_j)$

Fig. 2. The formal system $\rhd$ for checkpoint compliance

We do not show this property, since it does not affect in any way the formalism and the related proofs, and consider rule ($\textsc{Ax}_1$) instead, showing that the system is sound and complete even with respect to non nice compositions.

3.1. Soundness and completeness

It is handy to extend the compliance relation to judgments.

Definition 3.10 (*Judgment semantics*) We write

- $\models \Gamma$ if $\delta' \prec \rho' \dashv\!\!\!\!\mid^{\blacktriangle} \gamma' \prec \sigma'$ for all $\delta' \prec \rho' \dashv^{\blacktriangle} \gamma' \prec \sigma' \in \Gamma$;
- $\Gamma \models \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ if $\models \Gamma$ implies $\delta \prec \rho \dashv\!\!\!\!\mid^{\blacktriangle} \gamma \prec \sigma$.

To facilitate the proofs below, it is convenient to consider a stratified version of Definition 3.10.

Definition 3.11 (*Stratified judgment semantics*) We write

- $\models_k \Gamma$ if $\delta' \prec \rho' \dashv\!\!\!\!\mid^{\blacktriangle}_k \gamma' \prec \sigma'$ for all $\delta' \prec \rho' \dashv^{\blacktriangle} \gamma' \prec \sigma' \in \Gamma$;
- $\Gamma \models_k \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ if $\models_k \Gamma$ implies $\delta \prec \rho \dashv\!\!\!\!\mid^{\blacktriangle}_k \gamma \prec \sigma$,

where $k \geq 0$.

Observing that $\dashv\!\!\!\!\mid^{\blacktriangle}_{k+1} \subseteq \dashv\!\!\!\!\mid^{\blacktriangle}_k$, we have that $\models_{k+1} \Gamma$ implies $\models_k \Gamma$. Also it is immediate to verify that the following holds:

Fact 3.12 If $\Gamma \models_k \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ for all k, then $\Gamma \models \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$.

The opposite implication of Fact 3.12 does not hold, as shown in the following example.
Consider $\Gamma = \{\circ \prec {}_{\blacktriangle}\overline{a}.c \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}a\}$ and $\circ \prec {}_{\blacktriangle}b \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}c$. Clearly $\circ \prec {}_{\blacktriangle}b \not\dashv\!\!\!\!\mid^{\blacktriangle} \circ \prec {}_{\blacktriangle}c$, moreover it is easy to check that $\not\models \Gamma$. However, $\models_1 \Gamma$. In fact $\circ \prec {}_{\blacktriangle}\overline{a}.c \dashv\!\!\!\!\mid^{\blacktriangle}_1 \circ \prec {}_{\blacktriangle}a$ (since, trivially, $c \dashv\!\!\!\!\mid^{\blacktriangle}_0 \mathbf{1}$). So, $\Gamma \models \circ \prec {}_{\blacktriangle}b \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}c$ holds simply because $\not\models \Gamma$, whereas $\Gamma \models_k \circ \prec {}_{\blacktriangle}b \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}c$ for all k does not hold, since $\models_1 \Gamma$ but $(\circ \prec {}_{\blacktriangle}b) \not\dashv\!\!\!\!\mid^{\blacktriangle}_1 (\circ \prec {}_{\blacktriangle}c)$.

As a matter of fact the best we can say is that if $\Gamma \models \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$, then $\Gamma \models_k \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ for all but finitely many k. However we don't have to bother about this, because in the the next proofs we need only Fact 3.12 and the fact that $\delta \prec \rho \dashv\!\!\mid^{\blacktriangle}_k \gamma \prec \sigma$ for all k if and only if $\delta \prec \rho \dashv\!\!\mid^{\blacktriangle} \gamma \prec \sigma$ (which is true by Lemma 3.6).

We can now show that the formal system is sound with respect to the judgment semantics.

Theorem 3.13 (Soundness) *If* $\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$, *then* $\Gamma \models \delta \prec \rho \dashv\!\!\mid^{\blacktriangle} \gamma \prec \sigma$.

Proof By Fact 3.12 it suffices to prove that:

$\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ implies $\Gamma \models_k \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ for all k.

We proceed by simultaneous induction over the derivation $\mathcal{D}$ of $\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ and over k. Since $\Gamma \models_0 \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ trivially holds, we shall keep implicit the case $k = 0$. We distinguish the possible cases of the last rule in $\mathcal{D}$.

Case (Ax). Then either $\mathcal{D}$ consists of the inference:

$$\frac{}{\Gamma \rhd \circ \prec \mathbf{1} \dashv^{\blacktriangle} \circ \prec \sigma}\ (\mathrm{Ax}_1)$$

and the thesis is immediate since $\circ \prec \mathbf{1} \dashv\!\!\mid^{\blacktriangle}_k \circ \prec \sigma$ for all k; or $\mathcal{D}$ ends by:

$$\frac{\Gamma \rhd \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma}{\Gamma \rhd \delta \prec \mathbf{1} \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\mathrm{Ax}_2)$$

For $k > 0$, let $\models_k \Gamma$; then $\models_{k-1} \Gamma$ and, by induction over $\mathcal{D}$, $\Gamma \models_{k-1} \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma$. Hence we have $\circ \prec \delta \dashv\!\!\mid^{\blacktriangle}_{k-1} \circ \prec \gamma$, that implies $\delta \prec \mathbf{1} \dashv\!\!\mid^{\blacktriangle}_k \gamma \prec \sigma$ by definition, and lastly $\Gamma \models_k \delta \prec \mathbf{1} \dashv^{\blacktriangle} \gamma \prec \sigma$ as required.

Case (Hyp). Then $\mathcal{D}$ consists of the inference:

$$\frac{}{\Gamma, \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\mathrm{Hyp})$$

where $\Gamma, \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma \models_k \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ holds trivially for all k.

Case $(+ \cdot \oplus)$. If $\delta = \gamma = \circ$, then $\mathcal{D}$ ends by:

$$\frac{\forall\, j \in J.\ \Gamma' \rhd \mathbf{p}(\circ, \rho) \prec \rho_j \dashv^{\blacktriangle} \mathbf{p}(\circ, \sigma) \prec \sigma_j}{\Gamma \rhd \circ \prec \rho \dashv^{\blacktriangle} \circ \prec \sigma}\ (+ \cdot \oplus)$$

where $\Gamma' = \Gamma,\ \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ and $\rho = {}_{\triangle 1}(\sum_{i \in I \cup J} a_i.\rho_i)$ and $\sigma = {}_{\triangle 2}(\bigoplus_{j \in J} \overline{a}_j.\sigma_j)$.

We have to prove that $\Gamma \models_k \circ \prec \rho \dashv^{\blacktriangle} \circ \prec \sigma$ for all k.

Let $k > 0$; let us assume, by induction over k, $\Gamma \models_{k-1} \circ \prec \rho \dashv^{\blacktriangle} \circ \prec \sigma$. If $\models_k \Gamma$, then $\models_{k-1} \Gamma$, which implies $\circ \prec \rho \dashv\!\!\mid^{\blacktriangle}_{k-1} \circ \prec \sigma$ and hence $\models_{k-1} \Gamma'$, since $\Gamma' = \Gamma, \circ \prec \rho \dashv^{\blacktriangle} \circ \prec \sigma$. By induction over $\mathcal{D}$ we know that $\Gamma' \models_h \mathbf{p}(\circ, \rho) \prec \rho_j \dashv^{\blacktriangle} \mathbf{p}(\circ, \sigma) \prec \sigma_j$ for all $j \in J$ and for all h, hence $\Gamma' \models_{k-1} \mathbf{p}(\circ, \rho) \prec \rho_j \dashv^{\blacktriangle} \mathbf{p}(\circ, \sigma) \prec \sigma_j$.

Combining this with $\models_{k-1} \Gamma'$ we get $\mathbf{p}(\circ, \rho) \prec \rho_j \dashv\!\!\mid^{\blacktriangle}_{k-1} \mathbf{p}(\circ, \sigma) \prec \sigma_j$ for all $j \in J$. The one step reducts of $\circ \prec \rho \parallel \circ \prec \sigma$ are exactly

$$\circ \prec {}_{\triangle 1}(\textstyle\sum_{i \in I \cup J} a_i.\rho_i) \parallel \circ \prec {}_{\triangle 2}(\bigoplus_{j \in J} \overline{a}_j.\sigma_j) \xrightarrow{\tau} \mathbf{p}(\delta, \rho) \prec \rho_j \parallel \mathbf{p}(\gamma, \sigma) \prec \sigma_j$$

for all $j \in J$. So we conclude $\circ \prec \rho \dashv\!\!\mid^{\blacktriangle}_k \circ \prec \sigma$ as desired.

Otherwise $\mathcal{D}$ ends by:

$$\frac{\forall\, j \in J.\ \Gamma' \rhd \mathbf{p}(\delta, \rho) \prec \rho_j \dashv^{\blacktriangle} \mathbf{p}(\gamma, \sigma) \prec \sigma_j \qquad \Gamma' \rhd \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma}{\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (+ \cdot \oplus)$$

where Γ', ρ and σ are as in the previous case, and both $\delta, \gamma \neq \circ$. Reasoning as before, if $\models_k \Gamma$, then $\models_{k-1} \Gamma'$ and hence $\mathbf{p}(\delta, \rho) \prec \rho_j \dashv\!\!\mid^{\blacktriangle}_{k-1} \mathbf{p}(\gamma, \sigma) \prec \sigma_j$ for all $j \in J$. We get $\Gamma' \models_{k-1} \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma$ by induction over $\mathcal{D}$, which implies $\circ \prec \delta \dashv\!\!\mid^{\blacktriangle}_{k-1} \circ \prec \gamma$. Since $\mathbf{p}(\delta, \rho) \prec \rho_j \parallel \mathbf{p}(\gamma, \sigma) \prec \sigma_j$ for $j \in J$ and $\circ \prec \delta \parallel \circ \prec \gamma$ are all the one step reducts of $\delta \prec \rho \parallel \gamma \prec \sigma$, we conclude $\delta \prec \rho \dashv\!\!\mid^{\blacktriangle}_k \gamma \prec \sigma$ by definition of $\dashv\!\!\mid^{\blacktriangle}_k$.

The proof for the remaining cases of rule $(\oplus \cdot +)$ are similar. □

$\mathbf{Prove}(\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma)$

$\mathbf{if}\ \ \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma \in \Gamma\ \ \mathbf{then}\ \ \dfrac{}{\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\textsc{Hyp})$

$\mathbf{else}\ \mathbf{if}\ \ \rho = \mathbf{1}\ \ \mathbf{then}$

$\quad \mathbf{if}\ \ \delta = \circ\ \mathbf{or}\ \gamma = \circ\ \ \mathbf{then}\ \ \dfrac{}{\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\textsc{Ax})$

$\quad \mathbf{else\ let}\ \ \mathcal{D} = \mathbf{Prove}\,(\Gamma \rhd \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma) \neq \mathbf{fail}\ \ \mathbf{in}\ \ \dfrac{\mathcal{D}}{\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\textsc{Ax})$

$\mathbf{else}\ \mathbf{if}\ \ \rho = {\triangle_1}(\sum_{i \in I \cup J} a_i.\rho_i)\ \ \mathbf{and}\ \sigma = {\triangle_2}(\bigoplus_{j \in J} \overline{a}_j.\sigma_j)\ \ \mathbf{then}$

$\quad \mathbf{let}\ \ \Gamma' = \Gamma, \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma\ \ \mathbf{in}$

$\quad\ \mathbf{foreach}\ j \in J\ \ \mathbf{let}\ \ \mathcal{D}_j = \mathbf{Prove}\,(\Gamma' \rhd \mathbf{p}(\delta, \rho) \prec \rho_j \dashv^{\blacktriangle} \mathbf{p}(\gamma, \sigma) \prec \sigma_j) \neq \mathbf{fail}\ \ \mathbf{in}$

$\quad\quad \mathbf{if}\ \ \delta = \circ\ \mathbf{or}\ \gamma = \circ\ \ \mathbf{then}\ \ \dfrac{\mathcal{D}_j \quad (\forall j \in J)}{\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (+ \cdot \oplus)$

$\quad\quad \mathbf{else\ let}\ \ \mathcal{D} = \mathbf{Prove}\,(\Gamma' \rhd \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma) \neq \mathbf{fail}\ \ \mathbf{in}\ \ \dfrac{\mathcal{D}_j \quad (\forall j \in J) \qquad \mathcal{D}}{\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (+ \cdot \oplus)$

$\mathbf{else}\ \mathbf{if}\ \ \rho = {\triangle_1}(\bigoplus_{i \in I} \overline{a}_i.\rho_i)\ \ \mathbf{and}\ \sigma = {\triangle_2}(\sum_{j \in I \cup J} a_j.\sigma_j)\ \ \mathbf{then}$

$\quad \mathbf{let}\ \ \Gamma' = \Gamma, \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma\ \ \mathbf{in}$

$\quad\ \mathbf{foreach}\ i \in I\ \ \mathbf{let}\ \ \mathcal{D}_i = \mathbf{Prove}\,(\Gamma' \rhd \mathbf{p}(\delta, \rho) \prec \rho_i \dashv^{\blacktriangle} \mathbf{p}(\gamma, \sigma) \prec \sigma_i) \neq \mathbf{fail}\ \ \mathbf{in}$

$\quad\quad \mathbf{if}\ \ \delta = \circ\ \mathbf{or}\ \gamma = \circ\ \ \mathbf{then}\ \ \dfrac{\mathcal{D}_i \quad (\forall i \in I)}{\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\oplus \cdot +)$

$\quad\quad \mathbf{else\ let}\ \ \mathcal{D} = \mathbf{Prove}\,(\Gamma' \rhd \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma) \neq \mathbf{fail}\ \ \mathbf{in}\ \ \dfrac{\mathcal{D}_i \quad (\forall i \in I) \qquad \mathcal{D}}{\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma}\ (\oplus \cdot +)$

$\mathbf{else}\ \ \mathbf{fail}$

Fig. 3. The algorithm **Prove**

We now establish the completeness of the axiomatic system and decidability of derivability (and therefore of compliance) by means of the proof reconstruction algorithm **Prove** of Fig. 3.

Given a judgment $\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$, if the algorithm **Prove** terminates, then it either returns a derivation $\mathcal{D}$ with conclusion $\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$, or it returns **fail**. As a matter of fact we prove in Lemma 3.21 that **Prove** always terminates.

The **Prove** algorithm tries to construct a proof for a given judgment by recursively proceeding bottom-up, each time applying the only possible rule that has the given judgment as conclusion, once it has been checked that rule (Hyp) does not apply. The algorithm fails as soon as the current judgment cannot be the conclusion of any rule.

Recall that, as discussed in Remark 3.9, due to the presence of rollbacks and how they affect the operational semantics of behaviours, rule (Hyp) can apply also in absence of recursion.

We put the algorithm **Prove** at work on the compliant client and server defined by (2) in the Introduction. In this example, as well as in the following ones, we use bg, bt, dsc, pr as short for bag, belt, discount, price, respectively.

Example **3.14** Let $\rho = {}_{\blacktriangle}(\overline{\mathsf{dsc}}.{}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bg}}.\mathsf{pr})$ and $\sigma = {}_{\blacktriangle}(\mathsf{dsc}.\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bg}.\overline{\mathsf{pr}})$.
The following is the result of **Prove**$(\delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma)$, where

$\Gamma_1 = \{\circ \prec \rho \dashv^{\blacktriangle} \circ \prec \sigma\}$, $\Gamma_2 = \Gamma_1,\ \rho \prec {}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \dashv^{\blacktriangle} \sigma \prec \mathsf{bg}.\overline{\mathsf{pr}}$, $\Gamma_3 = \Gamma_2,\ {}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \prec \mathsf{pr} \dashv^{\blacktriangle} \sigma \prec \overline{\mathsf{pr}}$,
$\Gamma_4 = \Gamma_3,\ \circ \prec {}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \dashv^{\blacktriangle} \circ \prec \sigma$, $\Gamma_5 = \Gamma_1,\ \rho \prec \mathsf{pr} \dashv^{\blacktriangle} \sigma \prec \overline{\mathsf{pr}}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_4 \rhd {}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \prec \mathsf{pr} \dashv^{\blacktriangle} \sigma \prec \overline{\mathsf{pr}}}\,(\textsc{Hyp})}{\Gamma_3 \rhd \circ \prec {}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \dashv^{\blacktriangle} \circ \prec \sigma}}{\Gamma_3 \rhd {}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \prec \mathbf{1} \dashv^{\blacktriangle} \sigma \prec \mathbf{1}}\,(\textsc{Ax}) \quad \mathcal{D}_3}{\Gamma_2 \rhd {}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \prec \mathsf{pr} \dashv^{\blacktriangle} \sigma \prec \overline{\mathsf{pr}}}\,(+\cdot\oplus) \quad \mathcal{D}_2}{\Gamma_1 \rhd \rho \prec {}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \dashv^{\blacktriangle} \sigma \prec \mathsf{bg}.\overline{\mathsf{pr}}}\,(\oplus\cdot+) \qquad \dfrac{\dfrac{\dfrac{}{\Gamma_5 \rhd \circ \prec \rho \dashv^{\blacktriangle} \circ \prec \sigma}\,(\textsc{Hyp})}{\Gamma_5 \rhd \rho \prec \mathbf{1} \dashv^{\blacktriangle} \sigma \prec \mathbf{1}}\,(\textsc{Ax}) \quad \mathcal{D}_5}{\Gamma_1 \rhd \rho \prec \mathsf{pr} \dashv^{\blacktriangle} \sigma \prec \overline{\mathsf{pr}}}\,(+\cdot\oplus)$$

$$\rhd \circ \prec \rho \dashv^{\blacktriangle} \circ \prec \sigma \qquad (\oplus\cdot+)$$

where $\mathcal{D}_i$ $(i = 2, 5)$ is the derivation $\dfrac{}{\Gamma_i \rhd \circ \prec \rho \dashv^{\blacktriangle} \circ \prec \sigma}\,(\textsc{Hyp})$

and $\mathcal{D}_3$ is the derivation $\dfrac{\dfrac{}{\Gamma_4 \rhd {}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \prec \mathsf{pr} \dashv^{\blacktriangle} \sigma \prec \overline{\mathsf{pr}}}\,(\textsc{Hyp})}{\Gamma_3 \rhd \circ \prec {}_{\blacktriangle}\overline{\mathsf{bg}}.\mathsf{pr} \dashv^{\blacktriangle} \circ \prec \sigma}\,(\oplus\cdot+)$

It is not difficult to show that the algorithm **Prove** builds a derivation every time it does not fail.

Lemma 3.15 *If* ***Prove*** $(\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma) = \mathcal{D} \neq$ ***fail****, then* $\mathcal{D}$ *is a derivation of* $\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$.

Proof By construction and by induction over the tree of the recursive calls of **Prove**, which is finite if the execution terminates. □

The following example shows that the algorithm **Prove** fails on the client/server parallel composition (1) of the Introduction. For sake of readability, we omit the part "$\Gamma \rhd$" in the arguments of **Prove**.

Example **3.16** According to the algorithm of Fig. 3, in order to get the result of

Prove $(\circ \prec {}_{\blacktriangle}\overline{\mathsf{bt}}.\mathsf{pr}.\overline{\mathsf{bg}}.\mathsf{pr} \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}\mathsf{bt}.{}_{\blacktriangle}\overline{\mathsf{pr}}.\mathsf{bg}.\overline{\mathsf{pr}})$

we recursively proceed to look for the result of

Prove $({}_{\blacktriangle}\overline{\mathsf{bt}}.\mathsf{pr}.\overline{\mathsf{bg}}.\mathsf{pr} \prec \mathsf{pr}.\overline{\mathsf{bg}}.\mathsf{pr} \dashv^{\blacktriangle} {}_{\blacktriangle}\mathsf{bt}.{}_{\blacktriangle}\overline{\mathsf{pr}}.\mathsf{bg}.\overline{\mathsf{pr}} \prec {}_{\blacktriangle}\overline{\mathsf{pr}}.\mathsf{bg}.\overline{\mathsf{pr}})$

which immediately calls

Prove $({}_{\blacktriangle}\overline{\mathsf{bt}}.\mathsf{pr}.\overline{\mathsf{bg}}.\mathsf{pr} \prec \overline{\mathsf{bg}}.\mathsf{pr} \dashv^{\blacktriangle} {}_{\blacktriangle}\overline{\mathsf{pr}}.\mathsf{bg}.\overline{\mathsf{pr}} \prec \mathsf{bg}.\overline{\mathsf{pr}})$

Notice how the system takes care that the second checkpoint of the server has been traversed, by updating its past. The algorithm then proceeds by calling

Prove $({}_{\blacktriangle}\overline{\mathsf{bt}}.\mathsf{pr}.\overline{\mathsf{bg}}.\mathsf{pr} \prec \mathsf{pr} \dashv^{\blacktriangle} {}_{\blacktriangle}\overline{\mathsf{pr}}.\mathsf{bg}.\overline{\mathsf{pr}} \prec \overline{\mathsf{pr}})$

This last call produces, in turn, the call of

Prove $({}_{\blacktriangle}\overline{\mathsf{bt}}.\mathsf{pr}.\overline{\mathsf{bg}}.\mathsf{pr} \prec \mathbf{1} \dashv^{\blacktriangle} {}_{\blacktriangle}\overline{\mathsf{pr}}.\mathsf{bg}.\overline{\mathsf{pr}} \prec \mathbf{1})$,

that results in the following failing call (giving the overall failure of the algorithm)

Prove $(\circ \prec {}_{\blacktriangle}\overline{\mathsf{bt}}.\mathsf{pr}.\overline{\mathsf{bg}}.\mathsf{pr} \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle}\overline{\mathsf{pr}}.\mathsf{bg}.\overline{\mathsf{pr}}) =$ **fail**

since $\mathcal{A}^{\oplus}({}_{\blacktriangle}\overline{\mathsf{bt}}.\mathsf{pr}.\overline{\mathsf{bg}}.\mathsf{pr}) \cap \mathcal{A}^{+}({}_{\blacktriangle}\overline{\mathsf{pr}}.\mathsf{bg}.\overline{\mathsf{pr}}) = \emptyset$.

Notice that the algorithm detects the failure that occurs after a rollback that takes place after the synchronisations on the names $\mathsf{bt}, \mathsf{pr}, \mathsf{bg}, \mathsf{pr}$. As a matter of fact, a failure would occur also in case of a rollback taking place just after the synchronisations on the names bg, pr. In order to detect the latter synchronisation failure instead of the former one, in the main clauses of the algorithm **Prove** the recursive calls when δ and γ are not $\circ$ should be such that the derivation $\mathcal{D}$ is computed before the derivations $\mathcal{D}_j$s (or $\mathcal{D}_i$s).

The following lemma assures that a failure of the algorithm **Prove** can only happen if the configurations are not compliant.

Lemma 3.17 *If* ***Prove***$(\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma) =$ ***fail****, then* $\delta \prec \rho \not\asymp^{\blacktriangle} \gamma \prec \sigma$.

Proof Observe that if **Prove**$(\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma) =$ **fail**, then $\delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma \notin \Gamma$. This will be tacitly assumed in all cases below.

Let k be the maximum number of recursive calls of the terminating execution of

Prove $(\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma)$

returning **fail**. Then we prove by induction over k, that there exists h (actually greater than k) such that $\delta \prec \rho \not\asymp^{\blacktriangle}_h \gamma \prec \sigma$. This suffices since $\dashv^{\blacktriangle} = \bigcap_{k\in\mathbb{N}} \dashv^{\blacktriangle}_k$ by Lemma 3.6.

If $k = 0$, then **Prove**$(\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma) =$ **fail** implies $\rho \neq \mathbf{1}$ (otherwise either **Prove** succeeds or there is at least one recursive call) and $\sigma = \mathbf{1}$ (again because otherwise there would be at least one recursive call of **Prove**). But then $\delta \prec \rho \parallel \gamma \prec \mathbf{1} \not\xrightarrow{\tau}$ which implies $\delta \prec \rho \not\asymp^{\blacktriangle}_h \gamma \prec \mathbf{1} \not\xrightarrow{\tau}$ for any $h > 0$.

Let $k > 0$. If $\rho \neq \mathbf{1}$ and $\sigma = \mathbf{1}$, then we reason as in the base case. Otherwise the negative result of the computation depends on the failure of some recursive calls. Since all cases are similar, we consider for example the case when $\rho = {}_{\triangle I}(\sum_{i\in I\cup J} a_i.\rho_i)$ and $\sigma = {}_{\triangle 2}(\bigoplus_{j\in J} \overline{a}_j.\sigma_j)$, and $\delta, \gamma \neq \circ$. Then either **Prove**$(\Gamma' \rhd \mathbf{p}(\delta,\rho) \prec \rho_j \dashv^{\blacktriangle} \mathbf{p}(\gamma,\sigma) \prec \sigma_j) =$ **fail** for some $j \in J$, or **Prove**$(\Gamma' \rhd \circ \prec \delta \dashv^{\blacktriangle} \circ \prec \gamma) =$ **fail**, where $\Gamma' = \Gamma, \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$. The maximum number of recursive calls in these computations is $\leq k-1$. By induction hypothesis there exists h such that either $\mathbf{p}(\delta,\rho) \prec \rho_j \not\asymp^{\blacktriangle}_h \mathbf{p}(\gamma,\sigma) \prec \sigma_j$ or $\circ \prec \delta \not\asymp^{\blacktriangle}_h \circ \prec \gamma$. In both cases we have $\delta \prec \rho \not\asymp^{\blacktriangle}_{h+1} \gamma \prec \sigma$. □

The following example illustrates how recursive terms are treated in our formal system. Recall that we identify $\mathsf{rec}\, x.\sigma$ with $\sigma[\mathsf{rec}\, x.\sigma/x]$.

Example **3.18** The following is the result of **Prove**$(\rhd \circ \prec {}_{\blacktriangle}\overline{a}.\mathsf{rec}\, x.\overline{b}.x \dashv^{\blacktriangle} \circ \prec \mathsf{rec}\, x.{}_{\blacktriangle}(a.x + b.x))$.

$$\dfrac{\dfrac{\dfrac{}{\Gamma_2 \rhd {}_{\blacktriangle}\overline{a}.\mathsf{rec}\, x.\overline{b}.x \prec \mathsf{rec}\, x.\overline{b}.x \dashv^{\blacktriangle} \mathsf{rec}\, x.{}_{\blacktriangle}(a.x + b.x) \prec \mathsf{rec}\, x.{}_{\blacktriangle}(a.x + b.x)}\,(\textsc{Hyp}) \quad \mathcal{D}}{\Gamma_1 \rhd {}_{\blacktriangle}\overline{a}.\mathsf{rec}\, x.\overline{b}.x \prec \mathsf{rec}\, x.\overline{b}.x \dashv^{\blacktriangle} \mathsf{rec}\, x.{}_{\blacktriangle}(a.x + b.x) \prec \mathsf{rec}\, x.{}_{\blacktriangle}(a.x + b.x)}\,(\oplus\cdot+)}{\rhd \circ \prec {}_{\blacktriangle}\overline{a}.\mathsf{rec}\, x.\overline{b}.x \dashv^{\blacktriangle} \circ \prec \mathsf{rec}\, x.{}_{\blacktriangle}(a.x + b.x)}\,(\oplus\cdot+)$$

where $\mathcal{D}$ is $\dfrac{}{\Gamma_2 \rhd \circ \prec {}_{\blacktriangle}\overline{a}.\mathsf{rec}\, x.\overline{b}.x \dashv^{\blacktriangle} \circ \prec \mathsf{rec}\, x.{}_{\blacktriangle}(a.x + b.x)}\,(\textsc{Hyp})$
and where

$$\Gamma_1 = \{\circ \prec {}_{\blacktriangle}\overline{a}.\mathsf{rec}\, x.\overline{b}.x \dashv^{\blacktriangle} \circ \prec \mathsf{rec}\, x.{}_{\blacktriangle}(a.x + b.x)\}$$
$$\Gamma_2 = \Gamma_1, {}_{\blacktriangle}\overline{a}.\mathsf{rec}\, x.\overline{b}.x \prec \mathsf{rec}\, x.\overline{b}.x \dashv^{\blacktriangle} \mathsf{rec}\, x.{}_{\blacktriangle}(a.x + b.x) \prec \mathsf{rec}\, x.{}_{\blacktriangle}(a.x + b.x)$$

We show now the termination of **Prove**. This proof is inspired by the decidability proof for subtyping recursive types in the π-calculus [PS96].

We define the set of sub-behaviours of a behaviour as expected.

Definition 3.19 The function $\mathsf{Sub} : \mathsf{SB} \to \mathcal{P}(\mathsf{SB})$ is coinductively given by:

$$\begin{aligned}
\mathsf{Sub}(\mathbf{1}) &= \{\mathbf{1}\}\\
\mathsf{Sub}(\textstyle\sum_{i\in I} a_i.\sigma_i) &= \{\textstyle\sum_{i\in I} a_i.\sigma_i\} \cup \bigcup_{i\in I} \mathsf{Sub}(\sigma_i)\\
\mathsf{Sub}(\textstyle\bigoplus_{i\in I} \overline{a}_i.\sigma_i) &= \{\textstyle\bigoplus_{i\in I} \overline{a}_i.\sigma_i\} \cup \bigcup_{i\in I} \mathsf{Sub}(\sigma_i).
\end{aligned}$$

Since we assume $\mathsf{rec}\, x.\sigma = \sigma\{\mathsf{rec}\, x.\sigma/x\}$, behaviours containing recursive subterms are infinite terms, hence the coinductive character of Sub; in particular we have that $\mathsf{Sub}(\mathsf{rec}\, x.\sigma) = \mathsf{Sub}(\sigma\{\mathsf{rec}\, x.\sigma/x\})$. On the other hand, being recursion guarded, σ is always a regular tree. Hence:

Fact 3.20 For any σ, the set $\mathsf{Sub}(\sigma)$ is well defined and finite.

Lemma 3.21 *For all judgments* $\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ *the execution of* ***Prove*** $(\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma)$ *terminates.*

Proof Let $\mathsf{Sub}(\circ) = \{\circ\}$, extending Sub to $\mathsf{SB} \cup \{\circ\}$. Then given a judgment $\delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ we set:

$$\mathsf{Sub}(\delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma) = \{\delta' \prec \rho' \dashv^{\blacktriangle} \gamma' \prec \sigma' \mid \delta' \in \mathsf{Sub}(\delta) \cup \mathsf{Sub}(\rho) \cup \{\circ\}, \rho' \in \mathsf{Sub}(\rho) \cup \mathsf{Sub}(\delta), \\ \gamma' \in \mathsf{Sub}(\gamma) \cup \mathsf{Sub}(\sigma) \cup \{\circ\}, \sigma' \in \mathsf{Sub}(\sigma) \cup \mathsf{Sub}(\gamma)\}.$$

Fact 3.20 implies that $\mathsf{Sub}(\delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma)$ is finite. On the other hand, by direct inspection of the rules of the system in Fig. 2, we find that all $\delta' \prec \rho' \dashv^{\blacktriangle} \gamma' \prec \sigma'$ occurring in the premises of a derivation showing $\delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ belong to the set $\mathsf{Sub}(\delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma)$.

Now, if **Prove** $(\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma)$ would not terminate, then there would be an infinite sequence of nested calls **Prove**$(\Gamma_0 \rhd \delta_0 \prec \rho_0 \dashv^{\blacktriangle} \gamma_0 \prec \sigma_0)$, **Prove**$(\Gamma_1 \rhd \delta_1 \prec \rho_1 \dashv^{\blacktriangle} \gamma_1 \prec \sigma_1), \ldots$, where $\Gamma_0 \rhd \delta_0 \prec \rho_0 \dashv^{\blacktriangle} \gamma_0 \prec \sigma_0$ is just $\Gamma \rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$, and the sequence $\Gamma_0, \Gamma_1, \ldots$ is such that $\Gamma_{i+1} = \Gamma_i \cup \{\delta_i \prec \rho_i \dashv^{\blacktriangle} \gamma_i \prec \sigma_i\}$ for all i. Since **Prove** begins by checking $\delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma \in \Gamma$ and it returns in the positive case, non termination would only be possible if $\Gamma_i \subset \Gamma_{i+1}$ for infinitely many i, contradicting the fact that each Γ_i is a subset of the union of Γ and $\mathsf{Sub}(\delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma)$, which are both finite sets. □

Theorem 3.22 (Completeness) *If* $\delta \prec \rho \dashv\!\shortmid^{\blacktriangle} \gamma \prec \sigma$, *then* $\rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$ *is derivable.*

Proof The hypothesis $\delta \prec \rho \dashv\!\shortmid^{\blacktriangle} \gamma \prec \sigma$ implies that **Prove**$(\rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma) \neq$ **fail**, by Lemma 3.17. Since the execution of **Prove**$(\rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma)$ terminates by Lemma 3.21, we conclude by Lemma 3.15 that it produces a derivation with conclusion $\rhd \delta \prec \rho \dashv^{\blacktriangle} \gamma \prec \sigma$. □

Corollary 3.23 *The relation* $\dashv\!\shortmid^{\blacktriangle}$ *is decidable.*

Proof By definition $\rho \dashv\!\shortmid^{\blacktriangle} \sigma$ is equivalent to $\circ \prec \rho \dashv\!\shortmid^{\blacktriangle} \circ \prec \sigma$. By Theorems 3.13 and 3.22 $\circ \prec \rho \dashv\!\shortmid^{\blacktriangle} \circ \prec \sigma$ is equivalent to the derivability of $\rhd \circ \prec \rho \dashv^{\blacktriangle} \circ \prec \sigma$, which is decidable by means of **Prove**. □

4. The sub-behaviour relation

In the theory of session behaviours (and contracts in general) the compliance relation induces a preorder $\preceq$ formalising the notion of (server) *substitutivity*. The relation $\sigma \preceq \sigma'$ holds whenever, for any client ρ, if $\rho \dashv\!\shortmid \sigma$, then $\rho \dashv\!\shortmid \sigma'$. Here we adapt the definition of the sub-behaviour relation to the behaviours with checkpoints and to the ▲-compliance relation, obtaining a relation that we call ▲-*sub-behaviour* and dub $\leq^{\blacktriangle}$.

Definition 4.1 (▲-*Sub-behaviour*) The binary relation $\sigma \leq^{\blacktriangle} \sigma'$ over $\mathsf{SB}^{\blacktriangle}$ is defined by

$\sigma \leq^{\blacktriangle} \sigma'$ if $\rho \dashv\!\shortmid^{\blacktriangle} \sigma$ implies $\rho \dashv\!\shortmid^{\blacktriangle} \sigma'$ for all $\rho \in \mathsf{SB}^{\blacktriangle}$.

As in the case of behaviours without checkpoints, a behaviour with more external choices and fewer internal choices is "bigger than" a behaviour with fewer external choices and more internal choices. Formally

$$\begin{array}{ll} {}_{\blacktriangle}\sum_{i\in I}\, a_i.\sigma_i & \leq^{\blacktriangle} {}_{\blacktriangle}\sum_{i\in I\cup J}\, a_i.\sigma'_i \\ {}_{\blacktriangle}\bigoplus_{i\in I\cup J}\, \overline{a}_i.\sigma_i & \leq^{\blacktriangle} {}_{\blacktriangle}\bigoplus_{i\in I}\, \overline{a}_i.\sigma'_i \end{array}$$

where we assume $\sigma_i \leq^{\blacktriangle} \sigma'_i$ for all $i \in I$. A simple example is ${}_{\blacktriangle}a.(\overline{b} \oplus \overline{c}) \leq^{\blacktriangle} {}_{\blacktriangle}(a.\overline{b} + d)$.

The addition of checkpoints produces behaviours which are incomparable in general, even in case their erasures be identical. For example ${}_{\blacktriangle}a.b$ is compliant with ${}_{\blacktriangle}\overline{a}.\overline{b}$, but not with ${}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b}$, while ${}_{\blacktriangle}a.{}_{\blacktriangle}b$ is compliant with ${}_{\blacktriangle}\overline{a}.{}_{\blacktriangle}\overline{b}$, but not with ${}_{\blacktriangle}\overline{a}.\overline{b}$.

Nevertheless we can show decidability of the ▲-sub-behaviour relation. This proof will be obtained as a corollary of the property that the dual of a session-behaviour is actually the minimum among its servers with

respect to $\leq^{\blacktriangle}$. For any theory of subcontracts this *duals as minima* result is quite relevant, since the possibility of implementing contract-based query engines relies on it (see the Introduction of [Pad10] for a detailed description of the use of subcontract relations in search engines).

Lemma 4.2 *For all* $\rho, \sigma, \omega \in \mathsf{SB}^{\blacktriangle}$*:*

$$[\,\rho \dashv\!\shortmid^{\blacktriangle} \omega \;\; \textit{and} \;\; \overline{\omega} \dashv\!\shortmid^{\blacktriangle} \sigma\,] \;\; \textit{imply} \;\; \rho \dashv\!\shortmid^{\blacktriangle} \sigma.$$

Proof

It is easy to verify that an alternative definition of $\dashv\!\shortmid^{\blacktriangle}$ is the following one.
Let $\delta \prec \rho \dashv\!\shortmid^{\blacktriangle} \gamma \prec \sigma$ *if*

(1) $\delta \prec \rho \parallel \gamma \prec \sigma \stackrel{\tau}{\not\longrightarrow}$ *implies* $\rho = \mathbf{1}$*;*

(2) $\delta \prec \rho \parallel \gamma \prec \sigma \stackrel{\beta}{\longrightarrow} \delta' \prec \rho' \parallel \gamma' \prec \sigma'$ *implies* $\delta' \prec \rho' \;\dashv\!\shortmid^{\blacktriangle}\; \gamma' \prec \sigma'$*, where* $\beta \in \{\tau, \mathsf{rbk}\}$.

Therefore it is enough to show that

$$\{(\delta \prec \rho, \gamma \prec \sigma) \in \mathsf{SB}_{\prec} \times \mathsf{SB}_{\prec} \mid \exists \omega.\vartheta.\ \delta \prec \rho \dashv\!\shortmid^{\blacktriangle} \vartheta \prec \omega \;\&\; \overline{\vartheta} \prec \overline{\omega} \dashv\!\shortmid^{\blacktriangle} \gamma \prec \sigma\}$$

satisfies the above conditions (extending the duality operation to $\overline{\circ} = \circ$).

(1) $\delta \prec \rho \parallel \gamma \prec \sigma \stackrel{\tau}{\not\longrightarrow}$ implies $\rho = \mathbf{1}$. If we had, by contradiction, $\rho \neq \mathbf{1}$, let $\delta \prec \rho \stackrel{a}{\longrightarrow}$ for some a (the case in which $\delta \prec \rho \stackrel{\overline{a}}{\longrightarrow}$ can be treated similarly). From $\delta \prec \rho \dashv\!\shortmid^{\blacktriangle} \vartheta \prec \omega$ we then get $\vartheta \prec \omega \stackrel{\overline{a}}{\longrightarrow}$ and hence $\overline{\vartheta} \prec \overline{\omega} \stackrel{a}{\longrightarrow}$. So from $\overline{\vartheta} \prec \overline{\omega} \dashv\!\shortmid^{\blacktriangle} \gamma \prec \sigma$, we get $\gamma \prec \sigma \stackrel{\overline{a}}{\longrightarrow}$. Notice that we have $\mathcal{A}^{+}(\rho) \supseteq \mathcal{A}^{\oplus}(\sigma)$, since $\mathcal{A}^{+}(\rho) \supseteq \mathcal{A}^{\oplus}(\omega)$ and $\mathcal{A}^{+}(\overline{\omega}) \supseteq \mathcal{A}^{\oplus}(\sigma)$. So we could conclude $\delta \prec \rho \parallel \gamma \prec \sigma \stackrel{\tau}{\longrightarrow}$.

(2) $\delta \prec \rho \parallel \gamma \prec \sigma \stackrel{\beta}{\longrightarrow} \delta' \prec \rho' \parallel \gamma' \prec \sigma'$ with $\beta \in \{\tau, \mathsf{rbk}\}$ implies $\delta' \prec \rho' \dashv\!\shortmid^{\blacktriangle} \vartheta' \prec \omega'$ and $\overline{\vartheta'} \prec \overline{\omega'} \dashv\!\shortmid^{\blacktriangle} \gamma' \prec \sigma'$ for some ω', ϑ'.
If $\beta = \mathsf{rbk}$ we have actually to show that
$\delta \prec \rho \parallel \gamma \prec \sigma \stackrel{\mathsf{rbk}}{\longrightarrow} \circ \prec \delta \parallel \circ \prec \gamma$ implies $\circ \prec \delta \dashv\!\shortmid^{\blacktriangle} \circ \prec \omega'$ and $\overline{\circ} \prec \overline{\omega'} \dashv\!\shortmid^{\blacktriangle} \circ \prec \gamma$ for some ω'.
From $\delta \prec \rho \parallel \gamma \prec \sigma \stackrel{\mathsf{rbk}}{\longrightarrow} \circ \prec \delta \parallel \circ \prec \gamma$ we can infer that $\delta, \gamma \in \mathsf{SB}^{\blacktriangle}$. Being $\delta \prec \rho \dashv\!\shortmid^{\blacktriangle} \vartheta \prec \omega$ also $\vartheta \in \mathsf{SB}^{\blacktriangle}$. Therefore $\delta \prec \rho \parallel \vartheta \prec \omega \stackrel{\mathsf{rbk}}{\longrightarrow} \circ \prec \delta \parallel \circ \prec \vartheta$ and $\overline{\vartheta} \prec \overline{\omega} \parallel \gamma \prec \sigma \stackrel{\mathsf{rbk}}{\longrightarrow} \circ \prec \overline{\vartheta} \parallel \circ \prec \gamma$. This implies $\circ \prec \delta \dashv\!\shortmid^{\blacktriangle} \circ \prec \vartheta$ and $\circ \prec \overline{\vartheta} \dashv\!\shortmid^{\blacktriangle} \circ \prec \gamma$. We can then choose $\omega' = \vartheta$ in order to get what we need.
If $\beta = \tau$ we have actually to show that
$\delta \prec \rho \parallel \gamma \prec \sigma \stackrel{\tau}{\longrightarrow} \delta \prec \rho' \parallel \gamma \prec \sigma'$ implies
$\delta \prec \rho' \dashv\!\shortmid^{\blacktriangle} \vartheta' \prec \omega'$ and $\overline{\vartheta'} \prec \overline{\omega'} \dashv\!\shortmid^{\blacktriangle} \gamma \prec \sigma'$ for some ω', ϑ'. We proceed by cases, according to the form of ρ.
Let $\rho =_{\triangle 1} \sum_{i \in I} a_i.\rho_i$ and $\rho' = \rho_k$ and $\delta' = \mathbf{p}(\delta, \rho)$. Then $\sigma =_{\triangle 2} \bigoplus_{j \in J} \overline{a_j}.\sigma_j$ and $\sigma' = \sigma_k$ and $\gamma' = \mathbf{p}(\gamma, \sigma)$.
Now, from $\delta \prec \rho \dashv\!\shortmid^{\blacktriangle} \vartheta \prec \omega$ and $\overline{\vartheta} \prec \overline{\omega} \dashv\!\shortmid^{\blacktriangle} \gamma \prec \sigma$, we can infer that $\omega =_{\triangle 3} \bigoplus_{l \in L} \overline{a_l}.\omega_l$ and $J \subseteq L \subseteq I$.
Notice that by construction $\overline{\mathbf{p}(\vartheta, \omega)} = \mathbf{p}(\overline{\vartheta}, \overline{\omega})$. So in order to obtain what we need we can simply choose $\vartheta' = \mathbf{p}(\vartheta, \omega)$ and $\omega' = \omega_k$.
The case $\rho =_{\triangle} \bigoplus_{i \in I} \overline{a_i}.\rho_i$ can be treated similarly.

□

Proposition 4.3 *Let* $\omega \in \mathsf{SB}^{\blacktriangle}$*. Then* $\overline{\omega}$ *is the minimum server of* ω*, i.e.*

$$\omega \dashv\!\shortmid^{\blacktriangle} \sigma \;\; \textit{implies} \;\; \overline{\omega} \leq^{\blacktriangle} \sigma \;\; \textit{for all} \; \sigma \in \mathsf{SB}^{\blacktriangle}.$$

Proof We observe that $\omega \dashv\!\shortmid^{\blacktriangle} \overline{\omega}$. Hence it remains to show the minimality property with respect to $\leq^{\blacktriangle}$.
Let σ be a server ω is compliant with, i.e. $\omega \dashv\!\shortmid^{\blacktriangle} \sigma$. Now, in order to show $\overline{\omega} \leq^{\blacktriangle} \sigma$, let ρ be a client compliant with $\overline{\omega}$, i.e. $\rho \dashv\!\shortmid^{\blacktriangle} \overline{\omega}$. Then we have $\rho \dashv\!\shortmid^{\blacktriangle} \overline{\omega}$ and $\omega \dashv\!\shortmid^{\blacktriangle} \sigma$. By Lemma 4.2 and being duality an involution we get what we need, that is the compliance of ρ with σ. □

We are finally in place to establish the expected relation between subtyping, compliance and duality.

Theorem 4.4 $\sigma \leq^{\blacktriangle} \sigma'$ *if and only if* $\overline{\sigma} \dashv\!\shortmid^{\blacktriangle} \sigma'$.

Proof ($\Rightarrow$) Let $\overline{\sigma} \not\dashv\!\shortmid^{\blacktriangle} \sigma'$. Since we have $\overline{\sigma} \dashv\!\shortmid^{\blacktriangle} \sigma$, we get $\sigma \not\leq^{\blacktriangle} \sigma'$.
($\Leftarrow$) Let $\overline{\sigma} \dashv\!\shortmid^{\blacktriangle} \sigma'$. Then, by Proposition 4.3, we get $\sigma = \overline{\overline{\sigma}} \leq^{\blacktriangle} \sigma'$. □

By Theorem 4.4 and the decidability of $\dashv\!\vert^{\blacktriangle}$ we can now conclude:

Corollary 4.5 *The relation $\leq^{\blacktriangle}$ is decidable.*

5. Calculus with skips

In this section we modify the calculus of behaviours to allow the skip of certain outputs occurring after a rollback, in particular those already done and which have no corresponding input anymore. To represent the fact that already done outputs can be skipped, we bar them as in $\overline{\not a}$. Then we define session behaviours as in Figure 1 adding barred outputs. In the following $\overline{\overline{a}}$ stands for either $\overline{a}$ or $\overline{\not a}$. We use $\mathsf{SB}_{\mathtt{skp}}$ to denote the set of *session behaviours with checkpoints and skips*. From now on we call just *behaviours* the expressions in $\mathsf{SB}_{\mathtt{skp}}$.

In the operational semantics of the calculus we have to record not only the last encountered behaviour that was prefixed by a checkpoint in the interaction leading to the current behaviour (as before), but also the sequence of executed inputs and outputs (trace). As usual a *trace* μ is a finite sequence of actions and coactions, where the outputs can be either barred or not. We say that μ is a trace of σ if $\sigma \xrightarrow{\mu}$, according to the LTS of ordinary behaviours. Therefore we will consider configurations of the shape:

$$(\sigma, \mu) \prec \sigma'$$

where σ is a checkpointed internal or external choice, and $\sigma \xrightarrow{\mu} \sigma'$ if σ becomes σ' after performing all the actions in μ and in the given order. Let $\mathsf{SB}^{\blacktriangle}_{\mathtt{skp}}$ be the set of behaviours starting with $\blacktriangle$ and TR be the set of traces. Then $(\mathsf{SB}^{\blacktriangle}_{\mathtt{skp}} \times \mathsf{TR}) \cup \{\circ\}$ is the set of the "pasts with skips", and we denote by χ, ζ, possibly with superscripts, its elements.

To formalise the LTS of clients and servers we record the outputs that can be skipped by means of a function which, applied to a behaviour and to a trace of it, returns the behaviour where all the executed coactions have been barred. Given a pair (σ, μ) such that μ is a trace of σ, we define the function $\mathsf{b}(\sigma, \mu)$ by induction on μ, using ϵ to denote the empty trace:

$$\begin{array}{ll}
\mathsf{b}(\sigma, \epsilon) & = \sigma \\
\mathsf{b}(\blacktriangle\sigma, \mu) & = \blacktriangle\mathsf{b}(\sigma, \mu) \\
\mathsf{b}(\sum_{i\in I} a_i.\sigma_i,\ a_k\ \mu) & = \sum_{i\in I\setminus k} a_i.\sigma_i + a_k.\mathsf{b}(\sigma_k, \mu) \\
\mathsf{b}(\bigoplus_{i\in I} \overline{\overline{a}}_i.\sigma_i,\ \overline{a}_k\ \mu) & = \bigoplus_{i\in I\setminus k} \overline{\overline{a}}_i.\sigma_i \oplus \overline{\not a}_k.\mathsf{b}(\sigma_k, \mu)
\end{array}$$

where, if μ is a trace of σ, then $k \in I$ in both cases; also k is unique such, by definition of behaviours. The equi-recursive treatment of $\mathsf{rec}\, x.\sigma$ implies $\mathsf{b}(\mathsf{rec}\, x.\sigma, \mu) = \mathsf{b}(\sigma[\mathsf{rec}\, x.\sigma/x], \mu)$ if $\mu \neq \epsilon$, and $\mathsf{b}(\mathsf{rec}\, x.\sigma, \mu) = \mathsf{rec}\, x.\sigma$, otherwise.

We define $\blacktriangle^{?}\sigma = \begin{cases} \sigma & \text{if} \quad \sigma = \blacktriangle\sigma' , \\ \blacktriangle\sigma & \text{otherwise.} \end{cases}$

We are now ready to give the LTS of clients and servers.

Definition 5.1 (*Reduction of session behaviours with skips*) Let $k \in I$.

$$\circ \prec \textstyle\sum_{i\in I} a_i.\sigma_i \xrightarrow{a_k} \circ \prec \sigma_k \ (\circ +) \qquad \circ \prec \bigoplus_{i\in I} \overline{\overline{a}}_i.\sigma_i \xrightarrow{\overline{a}_k} \circ \prec \sigma_k \ (\circ \oplus)$$

$$(\sigma, \mu) \prec \textstyle\sum_{i\in I} a_i.\sigma_i \xrightarrow{a_k} (\sigma, \mu\ a_k) \prec \sigma_k \ (+) \qquad (\sigma, \mu) \prec \bigoplus_{i\in I} \overline{\overline{a}}_i.\sigma_i \xrightarrow{\overline{a}_k} (\sigma, \mu\ \overline{\overline{a}}_k) \prec \sigma_k \ (\oplus)$$

$$\chi \prec \blacktriangle\textstyle\sum_{i\in I} a_i.\sigma_i \xrightarrow{a_k} (\blacktriangle\sum_{i\in I} a_i.\sigma_i,\ a_k) \prec \sigma_k \ (\blacktriangle +) \qquad \chi \prec \blacktriangle\bigoplus_{i\in I} \overline{\overline{a}}_i.\sigma_i \xrightarrow{\overline{a}_k} (\blacktriangle\bigoplus_{i\in I} \overline{\overline{a}}_i.\sigma_i,\ \overline{\overline{a}}_k) \prec \sigma_k \ (\blacktriangle \oplus)$$

$$\circ \prec \blacktriangle(\textstyle\bigoplus_{i\in I} \overline{\overline{a}}_i.\sigma_i \oplus \overline{\not a}.\sigma) \xrightarrow{\mathtt{skp}} \circ \prec \blacktriangle^{?}\sigma \ (\mathtt{skp}) \qquad (\sigma, \mu) \prec \rho \xrightarrow{\mathsf{rbk}} \circ \prec \mathsf{b}(\sigma, \mu) \ (\mathsf{rbk})$$

Since we only allow to skip barred outputs, the starting configuration of the first application of rule (skp) must be the result of a rollback. This implies that the past must be $\circ$ and the current behaviour must be checkpointed. In the new configuration we preserve the past and the checkpoint. This choice can be illustrated by looking at reduction rules of client/server pairs and so we will discuss it after defining them.

We extend the function $\mathcal{A}^+(\cdot)$ of Definition 2.5 just by ignoring bars. The interaction of a client with a server is modelled by the reduction of their parallel composition, that can also involve barred outputs in synchronisations and be forward, skipping an already done output when no synchronisation is possible (condition $\mathcal{A}^{\oplus}(\ ) \cap \mathcal{A}^+(\ ) = \emptyset$ in rules (skp$\|$) and ($\|$skp)). Therefore we allow barred outputs in rules ($+\|\oplus$) and ($\oplus\|+$) of Definition 2.6 and we add the following rules:

$$\frac{\circ \prec \rho \xrightarrow{\text{skp}} \circ \prec \rho' \quad \mathcal{A}^{\oplus}(\rho) \cap \mathcal{A}^+(\sigma) = \emptyset}{\circ \prec \rho \parallel \circ \prec \sigma \xrightarrow{\text{skp}} \circ \prec \rho' \parallel \circ \prec \sigma} \ (\text{skp}\|)$$

$$\frac{\circ \prec \sigma \xrightarrow{\text{skp}} \circ \prec \sigma' \quad \mathcal{A}^{\oplus}(\sigma) \cap \mathcal{A}^+(\rho) = \emptyset}{\circ \prec \rho \parallel \circ \prec \sigma \xrightarrow{\text{skp}} \circ \prec \rho \parallel \circ \prec \sigma'} \ (\|\text{skp})$$

In the following we will use $\longrightarrow$ for the union of $\xrightarrow{\tau}$, $\xrightarrow{\text{rbk}}$ and $\xrightarrow{\text{skp}}$ and similarly of the other arrows denoting reductions.

Without skips, starting from nice client/server parallel compositions, only nice client/server parallel compositions can be reached (Proposition 2.8). This assures that the client and the server either both can or both cannot roll back in all configurations, an essential property as discussed in the Introduction. So we designed rule (skp) preserving this property.

Let us extend Definition 2.7 to the present calculus just replacing $\mathsf{SB}^{\blacktriangle}$ with $\mathsf{SB}^{\blacktriangle}_{\text{skp}}$. Then rules ($\|$skp) and (skp$\|$) can only be applied to nice parallel compositions which satisfy condition (1) of Definition 2.7 producing parallel compositions which satisfy the same condition. Therefore we can consider only nice client/server parallel compositions also when dealing with skips.

Example **5.2** We show a possible reduction sequence of the client server pair defined in example (3) of the Introduction.

$$\begin{aligned}
\circ \prec \mathsf{Client}'' \parallel \circ \prec \mathsf{Server}''' &\xrightarrow{\tau} (\mathsf{Client}'', \overline{\mathsf{lg}}) \prec (\overline{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}) \parallel (\mathsf{Server}''', \mathsf{lg}) \prec {}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}) \\
&\xrightarrow{\tau} (\mathsf{Client}'', \overline{\mathsf{lg}}\,\overline{\mathsf{bg}}) \prec \mathsf{pr} \parallel ({}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{lg}\,\mathsf{bg}) \prec \overline{\mathsf{pr}} \\
&\xrightarrow{\text{rbk}} \circ \prec {}_{\blacktriangle}\overline{\not{\mathsf{lg}}}.(\overline{\not{\mathsf{bg}}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}) \parallel \circ \prec {}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}) \\
&\xrightarrow{\text{skp}} \circ \prec {}_{\blacktriangle}(\overline{\not{\mathsf{bg}}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}) \parallel \circ \prec {}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}) \\
&\xrightarrow{\tau} ({}_{\blacktriangle}(\overline{\not{\mathsf{bg}}}.\mathsf{pr} \oplus \overline{\mathsf{bg}}.\mathsf{pr}), \overline{\not{\mathsf{bt}}}) \prec \mathsf{pr} \parallel ({}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bg}) \prec \overline{\mathsf{pr}} \\
&\xrightarrow{\tau} ({}_{\blacktriangle}(\overline{\not{\mathsf{bg}}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}), \overline{\not{\mathsf{bg}}}\,\mathsf{pr}) \prec \mathbf{1} \parallel ({}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bg}\,\overline{\mathsf{pr}}) \prec \mathbf{1}
\end{aligned}$$

Notice how, after the skipping of $\overline{\not{\mathsf{lg}}}$, the subsequent barred output $\overline{\not{\mathsf{bg}}}$ performed by the client is not skipped, since there is a corresponding input in the server. Of course the above reduction could go on by means a further rollback (this time, however, without causing any subsequent skip reduction).

For what concerns implementation issues related to the (skp) transition of processes, we recall that we are in a context with synchronous communications, which necessarily requires a sort of handshaking protocol. In presence of rollbacks and skips, the communication protocol must take care of the possibility that an output action has been previously "barred". In such a case the output will be skipped if no corresponding input action is found.

6. Compliance with skips

We denote the set of configurations by $\mathsf{SB}^{\prec}_{\mathtt{skp}}$, i.e. $\mathsf{SB}^{\prec}_{\mathtt{skp}} = \{\chi \prec \sigma \mid \chi \in (\mathsf{SB}^{\blacktriangle}_{\mathtt{skp}} \times \mathsf{TR}) \cup \{\circ\}, \sigma \in \mathsf{SB}\}$.

Definition 6.1 ($\blacktriangle_{\mathtt{skp}}$*-Compliance Relation* $\dashv\!\!\|^{\blacktriangle_{\mathtt{skp}}}$) (i) Let $\zeta \prec \rho, \chi \prec \sigma \in \mathsf{SB}^{\prec}_{\mathtt{skp}}$. We say that $\zeta \prec \rho$ is $\blacktriangle_{\mathtt{skp}}$*-compliant with* $\chi \prec \sigma$, written $\zeta \prec \rho \dashv\!\!\|^{\blacktriangle_{\mathtt{skp}}} \chi \prec \sigma$, when for all $\zeta' \prec \rho'$ and $\chi' \prec \sigma'$:

$$\text{if } \zeta \prec \rho \parallel \chi \prec \sigma \xrightarrow{*} \zeta' \prec \rho' \parallel \chi' \prec \sigma' \not\xrightarrow{\tau,\mathtt{skp}} \text{ then } \rho' = \mathbf{1},$$

where $\xrightarrow{\tau,\mathtt{skp}} = \xrightarrow{\tau} \cup \xrightarrow{\mathtt{skp}}$.

(ii) Let $\rho, \sigma \in \mathsf{SB}^{\blacktriangle}_{\mathtt{skp}}$. We say that ρ is $\blacktriangle_{\mathtt{skp}}$- *compliant* with σ (notation $\rho \dashv\!\!\|^{\blacktriangle_{\mathtt{skp}}} \sigma$) if $\circ \prec \rho \dashv\!\!\|^{\blacktriangle_{\mathtt{skp}}} \circ \prec \sigma$.

Also $\blacktriangle_{\mathtt{skp}}$-compliance satisfies the CONSERVATIVITY OF ERASURE discussed in the Introduction. So similarly to Proposition 3.2 we get

Proposition 6.2 *$\rho \dashv\!\!\|^{\blacktriangle_{\mathtt{skp}}} \sigma$ implies $\mathit{erase}(\rho) \dashv\!\!\| \mathit{erase}(\sigma)$ for all ρ, σ.*

The example given after Proposition 3.2 shows that the vice versa does not hold. Moreover $\dashv\!\!\|^{\blacktriangle_{\mathtt{skp}}}$ does not imply $\dashv\!\!\|^{\blacktriangle}$; take for example $\rho =_{\blacktriangle} \overline{a}.\overline{b}$ and $\sigma =_{\blacktriangle} a_{\blacktriangle} b$. Instead $\dashv\!\!\|^{\blacktriangle}$ implies $\dashv\!\!\|^{\blacktriangle_{\mathtt{skp}}}$, since rules ($\mathtt{skp}\|$) and ($\|\mathtt{skp}$) apply only when the parallel composition of client and server is stuck with the reduction of Definition 2.6.

With skips we did not manage to find a suitable notion of duality. In fact duality requires involution, and therefore we need to extend the syntax in order to distinguish inputs which are duals of barred outputs from inputs which are duals of non barred outputs. In absence of duality we cannot characterise subtyping as done in Sect. 4 for the calculus without skips.

In order to give a formal system characterising $\blacktriangle_{\mathtt{skp}}$-compliance it is handy to define a function $\mathtt{s} : ((\mathsf{SB}^{\blacktriangle}_{\mathtt{skp}} \times \mathsf{TR}) \cup \{\circ\}) \times \mathsf{SB} \times (\mathcal{N} \cup \overline{\mathcal{N}}) \to (\mathsf{SB}^{\blacktriangle}_{\mathtt{skp}} \times \mathsf{TR}) \cup \{\circ\}$ which returns:

- the pair of the second and the third argument, when the second argument is checkpointed,
- the first argument modified using the third, when the second argument is not checkpointed, and the first argument is pair,
- $\circ$ otherwise.

More precisely:

$$\mathtt{s}(\chi, {}_{\triangle}\sigma, \alpha) = \begin{cases} ({}_{\blacktriangle}\sigma, \alpha) & \text{if } \triangle = \blacktriangle \\ (\rho, \mu\,\alpha) & \text{if } \triangle \neq \blacktriangle \text{ and } \chi = (\rho, \mu) \\ \circ & \text{otherwise.} \end{cases}$$

The first six reduction rules in Definition 5.1 can be shortly written in terms of the function s:

Lemma 6.3 *Let $k \in I$.*

$$\chi \prec {}_{\triangle}(\textstyle\sum_{i\in I} a_i.\sigma_i) \xrightarrow{a_k} \mathtt{s}(\chi, {}_{\triangle}(\textstyle\sum_{i\in I} a_i.\sigma_i), a_k) \prec \sigma_k$$
$$\chi \prec {}_{\triangle}(\textstyle\bigoplus_{i\in I} \overline{a}_i.\sigma_i) \xrightarrow{\overline{a}_k} \mathtt{s}(\chi, {}_{\triangle}(\textstyle\bigoplus_{i\in I} \overline{a}_i.\sigma_i), \overline{a}_k) \prec \sigma_k$$

We now axiomatically characterise the $\blacktriangle_{\mathtt{skp}}$-compliance relation by means of a formal system, whose judgments are of the form $\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$, where Γ is an environment, i.e. a finite set $\Gamma = \{\zeta_i \prec \rho_i \dashv^{\blacktriangle}_{\mathtt{skp}} \chi_i \prec \sigma_i\}_{i\in I}$. The rules of the formal system are given in Figure 4, where in writing $\mathtt{b}\zeta$ we assume that $\zeta \in \mathsf{SB}^{\blacktriangle}_{\mathtt{skp}} \times \mathsf{TR}$. We denote by $\dashv^{\blacktriangle}_{\mathtt{skp}}$ the formal counterpart of $\dashv\!\!\|^{\blacktriangle_{\mathtt{skp}}}$.

The following example shows $\blacktriangle_{\mathtt{skp}}$-compliance of the client and the server defined by (3) in the Introduction.

Example **6.4** Let $\rho = {}_{\blacktriangle}\overline{\mathsf{lg}}.(\overline{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr})$ and $\sigma = {}_{\blacktriangle}\mathsf{lg}.{}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}})$ and
$\Gamma = \{\circ \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma\}$,
$\Gamma_1 = \Gamma,\ (\rho, \overline{\mathsf{lg}}) \prec \overline{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr} \dashv^{\blacktriangle}_{\mathtt{skp}} (\sigma, \mathsf{lg}) \prec {}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}})$,
$\Gamma_2 = \Gamma_1,\ (\rho, \overline{\mathsf{lg}}\,\overline{\mathsf{bg}}) \prec \mathsf{pr} \dashv^{\blacktriangle}_{\mathtt{skp}} ({}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bg}) \prec \overline{\mathsf{pr}}$,
$\Gamma_2' = \Gamma_1,\ (\rho, \overline{\mathsf{lg}}\,\overline{\mathsf{bt}}) \prec \mathsf{pr} \dashv^{\blacktriangle}_{\mathtt{skp}} ({}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bt}) \prec \overline{\mathsf{pr}}$,

$$\frac{\text{either } \zeta = \chi = \circ \text{ or } \Gamma \rhd \circ \prec \mathsf{b}\zeta \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec \mathsf{b}\chi}{\Gamma \rhd \zeta \prec \mathbf{1} \dashv^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma}\,(\textsc{Ax}) \qquad \frac{}{\Gamma, \zeta \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma \rhd \zeta \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma}\,(\textsc{Hyp})$$

$$\frac{\forall j \in J.\ \Gamma' \rhd \mathbf{s}(\zeta, \rho, a_j) \prec \rho_j \dashv^{\blacktriangle}_{\mathtt{skp}} \mathbf{s}(\chi, \sigma, \overline{a}_j) \prec \sigma_j \qquad \text{either } \zeta = \chi = \circ \text{ or } \Gamma' \rhd \circ \prec \mathsf{b}\zeta \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec \mathsf{b}\chi}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma}\,(+ \cdot \oplus)$$
where $\Gamma' = \Gamma,\ \zeta \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$ and $\rho = {}_{\triangle 1}(\sum_{i \in I \cup J} a_i.\rho_i)$ and $\sigma = {}_{\triangle 2}(\bigoplus_{j \in J} \overline{a}_j.\sigma_j)$

$$\frac{\forall i \in I.\ \Gamma' \rhd \mathbf{s}(\zeta, \rho, \overline{a}_i) \prec \rho_i \dashv^{\blacktriangle}_{\mathtt{skp}} \mathbf{s}(\chi, \sigma, a_i) \prec \sigma_i \qquad \text{either } \zeta = \chi = \circ \text{ or } \Gamma' \rhd \circ \prec \mathsf{b}\zeta \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec \mathsf{b}\chi}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma}\,(\oplus \cdot +)$$
where $\Gamma' = \Gamma,\ \zeta \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$ and $\rho = {}_{\triangle 1}(\bigoplus_{i \in I} \overline{a}_i.\rho_i)$ and $\sigma = {}_{\triangle 2}(\sum_{j \in I \cup J} a_j.\sigma_j)$

$$\frac{\forall j \in J.\ \Gamma' \rhd \circ \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec {}_{\blacktriangle ?}\sigma_j \qquad \mathcal{A}^{\oplus}(\sigma) \cap \mathcal{A}^{+}(\rho) = \emptyset}{\Gamma \rhd \circ \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma}\,(\cdot \mathtt{skp})$$
where $\Gamma' = \Gamma,\ \circ \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma$ and $\rho = {}_{\blacktriangle}(\sum_{i \in I} a_i.\rho_i)$ and $\sigma = {}_{\blacktriangle}(\bigoplus_{j \in J} \not{a}_j.\sigma_j \oplus \bigoplus_{l \in L} \overline{a}_l.\sigma_l)$

$$\frac{\forall i \in I.\ \Gamma' \rhd \circ \prec {}_{\blacktriangle ?}\rho_i \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma \qquad \mathcal{A}^{\oplus}(\rho) \cap \mathcal{A}^{+}(\sigma) = \emptyset}{\Gamma \rhd \circ \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma}\,(\mathtt{skp} \cdot)$$
where $\Gamma' = \Gamma,\ \circ \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma$ and $\rho = {}_{\blacktriangle}(\bigoplus_{i \in I} \not{a}_i.\rho_i \oplus \bigoplus_{l \in L} \overline{a}_l.\rho_l)$ and $\sigma = {}_{\blacktriangle}(\sum_{j \in J} a_j.\sigma_j)$

Fig. 4. The formal system for $\dashv^{\blacktriangle}_{\mathtt{skp}}$-compliance

$\Gamma_3 = \Gamma_2,\ \circ \prec \not{\mathsf{lg}}.(\not{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}) \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec {}_{\blacktriangle}(\mathsf{bg}.\not{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}})$,
$\Gamma_4 = \Gamma_3,\ \circ \prec {}_{\blacktriangle}(\not{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}) \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec {}_{\blacktriangle}(\mathsf{bg}.\not{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}})$,
$\Gamma_5 = \Gamma_4,\ ({}_{\blacktriangle}(\not{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}),\ \overline{\mathsf{bg}}) \prec \mathsf{pr} \dashv^{\blacktriangle}_{\mathtt{skp}} ({}_{\blacktriangle}(\mathsf{bg}.\not{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bg}) \prec \not{\mathsf{pr}}$,
$\Gamma_6 = \Gamma_4,\ ({}_{\blacktriangle}(\not{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}),\ \overline{\mathsf{bt}}) \prec \mathsf{pr} \dashv^{\blacktriangle}_{\mathtt{skp}} ({}_{\blacktriangle}(\mathsf{bg}.\not{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bt}) \prec \not{\mathsf{pr}}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{D}}{\Gamma_2 \ \rhd\ (\rho, \overline{\mathsf{lg}}\,\overline{\mathsf{bg}}\,\mathsf{pr}) \prec \mathbf{1} \dashv^{\blacktriangle}_{\mathtt{skp}} ({}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bg}\,\overline{\mathsf{pr}}) \prec \mathbf{1}}\,(\textsc{Ax}) \quad \mathcal{D}'}{\Gamma_1 \ \rhd\ (\rho, \overline{\mathsf{lg}}\,\overline{\mathsf{bg}}) \prec \mathsf{pr} \dashv^{\blacktriangle}_{\mathtt{skp}} ({}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bg}) \prec \overline{\mathsf{pr}}}\,(+ \cdot \oplus) \qquad \mathcal{D}_1 \quad \mathcal{D}_2}{\Gamma \rhd (\rho, \overline{\mathsf{lg}}) \prec \overline{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr} \dashv^{\blacktriangle}_{\mathtt{skp}} (\sigma, \mathsf{lg}) \prec {}_{\blacktriangle}(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}})}\,(\oplus \cdot +)}{\rhd \circ \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma}\,(\oplus \cdot +)$$

where:
$\mathcal{D}$ is the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_5 \ \rhd\ \circ \prec {}_{\blacktriangle}(\not{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}) \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec {}_{\blacktriangle}(\mathsf{bg}.\not{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}})}\,(\textsc{Hyp})}{\Gamma_5 \ \rhd\ ({}_{\blacktriangle}(\not{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}),\ \overline{\mathsf{bg}}\,\mathsf{pr}) \prec \mathbf{1} \dashv^{\blacktriangle}_{\mathtt{skp}} ({}_{\blacktriangle}(\mathsf{bg}.\not{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bg}\,\overline{\mathsf{pr}}) \prec \mathbf{1}}\,(\textsc{Ax}) \quad \mathcal{D}_4}{\Gamma_4 \ \rhd\ ({}_{\blacktriangle}(\not{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}),\ \overline{\mathsf{bg}}) \prec \mathsf{pr} \dashv^{\blacktriangle}_{\mathtt{skp}} ({}_{\blacktriangle}(\mathsf{bg}.\not{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bg}) \prec \not{\mathsf{pr}}}\,(+ \cdot \oplus) \qquad \mathcal{D}_3}{\Gamma_3 \ \rhd\ \circ \prec {}_{\blacktriangle}(\not{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}) \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec {}_{\blacktriangle}(\mathsf{bg}.\not{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}})}\,(\oplus \cdot +)}{\Gamma_2 \ \rhd\ \circ \prec \not{\mathsf{lg}}.(\not{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}) \dashv^{\blacktriangle}_{\mathtt{skp}} \circ \prec {}_{\blacktriangle}(\mathsf{bg}.\not{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}})}\,(\mathtt{skp} \cdot)$$

$\mathcal{D}_1$ is the derivation

$$\dfrac{\dfrac{\mathcal{D}_0}{\Gamma_2' \ \rhd \ (\rho, \overline{\mathsf{lg}}\,\overline{\mathsf{bt}}\,\mathsf{pr}) \prec \mathbf{1} \dashv\!\vert^{\blacktriangle}_{\mathsf{skp}} (\blacktriangle(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bt}\,\overline{\mathsf{pr}}) \prec \mathbf{1}}\,(\text{Ax}) \quad \mathcal{D}_0'}{\Gamma_1 \ \rhd \ (\rho, \overline{\mathsf{lg}}\,\overline{\mathsf{bt}}) \prec \mathsf{pr} \dashv\!\vert^{\blacktriangle}_{\mathsf{skp}} (\blacktriangle(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bt}) \prec \overline{\mathsf{pr}}}\,(+\cdot\oplus)$$

$\mathcal{D}_2$ is the derivation

$$\Gamma_1 \ \rhd \ \circ \prec \blacktriangle\!\!\!\!/\overline{\mathsf{lg}}.(\overline{\mathsf{bg}}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}) \dashv\!\vert^{\blacktriangle}_{\mathsf{skp}} \circ \prec \blacktriangle\mathsf{lg}.\blacktriangle(\mathsf{bg}.\overline{\mathsf{pr}} + \mathsf{bt}.\overline{\mathsf{pr}})$$

$\mathcal{D}_4$ is the derivation

$$\dfrac{}{\Gamma_5 \ \rhd \ \circ \prec \blacktriangle(\overline{\mathsf{bg}\!\!\!/}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}) \dashv\!\vert^{\blacktriangle}_{\mathsf{skp}} \circ \prec \blacktriangle(\mathsf{bg}.\overline{\mathsf{pr}\!\!\!/} + \mathsf{bt}.\overline{\mathsf{pr}})}\,(\text{Hyp})$$

$\mathcal{D}_3$ is the derivation

$$\dfrac{\dfrac{\Gamma_6 \ \rhd \ \circ \prec \blacktriangle(\overline{\mathsf{bg}\!\!\!/}.\mathsf{pr} \oplus \overline{\mathsf{bt}\!\!\!/}.\mathsf{pr}) \dashv\!\vert^{\blacktriangle}_{\mathsf{skp}} \circ \prec \blacktriangle(\mathsf{bg}.\overline{\mathsf{pr}\!\!\!/} + \mathsf{bt}.\overline{\mathsf{pr}\!\!\!/})}{\Gamma_6 \ \rhd \ (\blacktriangle(\overline{\mathsf{bg}\!\!\!/}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}), \ \overline{\mathsf{bt}}\,\mathsf{pr}) \prec \mathbf{1} \dashv\!\vert^{\blacktriangle}_{\mathsf{skp}} (\blacktriangle(\mathsf{bg}.\overline{\mathsf{pr}\!\!\!/} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bt}\,\overline{\mathsf{pr}}) \prec \mathbf{1}}\,(\text{Ax}) \quad \mathcal{D}_5}{\Gamma_4 \ \rhd \ (\blacktriangle(\overline{\mathsf{bg}\!\!\!/}.\mathsf{pr} \oplus \overline{\mathsf{bt}}.\mathsf{pr}), \ \overline{\mathsf{bt}}) \prec \mathsf{pr} \dashv\!\vert^{\blacktriangle}_{\mathsf{skp}} (\blacktriangle(\mathsf{bg}.\overline{\mathsf{pr}\!\!\!/} + \mathsf{bt}.\overline{\mathsf{pr}}), \mathsf{bt}) \prec \overline{\mathsf{pr}}}\,(+\cdot\oplus)$$

$\mathcal{D}_5$ is the derivation

$$\Gamma_6 \ \rhd \ \circ \prec \blacktriangle(\overline{\mathsf{bg}\!\!\!/}.\mathsf{pr} \oplus \overline{\mathsf{bt}\!\!\!/}.\mathsf{pr}) \dashv\!\vert^{\blacktriangle}_{\mathsf{skp}} \circ \prec \blacktriangle(\mathsf{bg}.\overline{\mathsf{pr}\!\!\!/} + \mathsf{bt}.\overline{\mathsf{pr}})$$

We omit writing $\mathcal{D}'$, $\mathcal{D}_0$ and $\mathcal{D}_0'$. Derivation $\mathcal{D}_0$ is obtained from derivation $\mathcal{D}$ by swapping the bt and bg choices. Derivations $\mathcal{D}'$ and $\mathcal{D}_0'$ are obtained from derivation $\mathcal{D}$ and $\mathcal{D}_0$, respectively, by erasing the barring of $\overline{\mathsf{pr}}$.
The derivations with vertical dots are shown incomplete for the sake of readability. Actually one could wonder why the judgments right below the vertical dots are not derived by means of rule (Hyp). As a matter of fact, the subjects of such judgments are present in the respective environments only modulo some bars. In the omitted derivations these judgments go into the environments and become the subjects of instances of rule (Hyp).

The remaining of this section is devoted to the proof of the soundness and completeness of the formal system in Fig. 4.

6.1. Soundness and completeness

As for the calculus without skips we start by providing a coinductive definition of $\dashv\!\vert^{\blacktriangle}_{\mathsf{skp}}$-compliance.

Definition 6.5 Let $\{ \dashv\!\vert^{\blacktriangle\mathsf{skp}}_k \mid k \in \mathbb{N}\}$ be the family of relations over $\mathsf{SB}^{\prec}_{\mathsf{skp}}$ such that $\dashv\!\vert^{\blacktriangle\mathsf{skp}}_0 = \mathsf{SB}_{\prec} \times \mathsf{SB}_{\prec}$ and $\zeta \prec \rho \ \dashv\!\vert^{\blacktriangle\mathsf{skp}}_{k+1} \chi \prec \sigma$ if either conditions (1), (2) and (3) of Definition 3.5 (with ζ, χ and $\dashv\!\vert^{\blacktriangle}$ in place of δ, γ and $\dashv\!\vert^{\blacktriangle\mathsf{skp}}$, respectively) hold, or if

$\rho \neq \mathbf{1}$ and $\zeta \prec \rho \parallel \chi \prec \sigma \xrightarrow{\mathsf{skp}}$ and
$\zeta \prec \rho \parallel \chi \prec \sigma \longrightarrow \zeta' \prec \rho' \parallel \chi' \prec \sigma'$ implies $\zeta' \prec \rho' \ \dashv\!\vert^{\blacktriangle}_k \chi' \prec \sigma'$ for all $\zeta' \prec \rho'$, $\chi' \prec \sigma'$.

Then we define $\dashv\!\vert^{\blacktriangle\mathsf{skp}}_{\mathsf{co}} = \bigcap_{k\in\mathbb{N}} \dashv\!\vert^{\blacktriangle\mathsf{skp}}_k$.

Lemma 6.6 *The relations* $\dashv\!\vert^{\blacktriangle\mathsf{skp}}$ *and* $\dashv\!\vert^{\blacktriangle\mathsf{skp}}_{co}$ *coincide.*

Toward the axiomatic characterisation of the $\dashv\!\vert^{\blacktriangle}_{\mathsf{skp}}$-compliance we define the semantic counterparts of the judgments in the formal system. The following definitions are the analogous of Definitions 3.10 and 3.11, respectively.

$\mathbf{Prove}_{\mathtt{skp}}(\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma)$

if $\zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma \in \Gamma$ **then** $\dfrac{}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma}\,(\textsc{Hyp})$

else if $\rho = \mathbf{1}$ **then**

  if $\zeta = \circ$ **or** $\chi = \circ$ **then** $\dfrac{}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma}\,(\textsc{Ax})$

  else let $\mathcal{D} = \mathbf{Prove}_{\mathtt{skp}}(\Gamma \rhd \circ \prec \mathsf{b}\zeta \dashv^{\blacktriangle} \circ \prec \mathsf{b}\chi) \neq \mathbf{fail}$ **in** $\dfrac{\mathcal{D}}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma}\,(\textsc{Ax})$

else if $\rho = \triangle 1(\sum_{i \in I \cup J} a_i.\rho_i)$ **and** $\sigma = \triangle 2(\bigoplus_{j \in J} \overline{a}_j.\sigma_j)$ **then**

  let $\Gamma' = \Gamma, \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma$ **in**

    foreach $j \in J$ **let** $\mathcal{D}_j = \mathbf{Prove}_{\mathtt{skp}}(\Gamma' \rhd \mathbf{s}(\zeta, \rho, a_j) \prec \rho_j \dashv^{\blacktriangle} \mathbf{s}(\chi, \sigma, \overline{a}_j) \prec \sigma_j) \neq \mathbf{fail}$ **in**

      if $\zeta = \circ$ **or** $\chi = \circ$ **then** $\dfrac{\mathcal{D}_j \quad (\forall j \in J)}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma}\,(+ \cdot \oplus)$

      else let $\mathcal{D} = \mathbf{Prove}_{\mathtt{skp}}(\Gamma' \rhd \circ \prec \mathsf{b}\zeta \dashv^{\blacktriangle} \circ \prec \mathsf{b}\chi) \neq \mathbf{fail}$ **in** $\dfrac{\mathcal{D}_j \quad (\forall j \in J) \qquad \mathcal{D}}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma}\,(+ \cdot \oplus)$

else if $\rho = \triangle 1(\bigoplus_{i \in I} \overline{a}_i.\rho_i)$ **and** $\sigma = \triangle 2(\sum_{j \in I \cup J} a_j.\sigma_j)$ **then**

  let $\Gamma' = \Gamma, \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma$ **in**

    foreach $i \in I$ **let** $\mathcal{D}_i = \mathbf{Prove}_{\mathtt{skp}}(\Gamma' \rhd \mathbf{s}(\zeta, \rho, \overline{a}_i) \prec \rho_i \dashv^{\blacktriangle} \mathbf{s}(\chi, \sigma, a_i) \prec \sigma_i) \neq \mathbf{fail}$ **in**

      if $\zeta = \circ$ **or** $\chi = \circ$ **then** $\dfrac{\mathcal{D}_i \quad (\forall i \in I)}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma}\,(\oplus \cdot +)$

      else let $\mathcal{D} = \mathbf{Prove}_{\mathtt{skp}}(\Gamma' \rhd \circ \prec \mathsf{b}\zeta \dashv^{\blacktriangle} \circ \prec \mathsf{b}\chi) \neq \mathbf{fail}$ **in** $\dfrac{\mathcal{D}_i \quad (\forall i \in I) \qquad \mathcal{D}}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma}\,(\oplus \cdot +)$

else if $\rho =_{\blacktriangle} (\sum_{i \in I} a_i.\rho_i)$ **and** $\sigma =_{\blacktriangle} (\bigoplus_{j \in J} \overline{a}_j.\sigma_j \oplus \bigoplus_{l \in L} \overline{a}_l.\sigma_l)$ **and** $\mathcal{A}^{\oplus}(\sigma) \cap \mathcal{A}^{+}(\rho) = \emptyset$ **then**

  let $\Gamma' = \Gamma, \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma$ **in**

    foreach $j \in J$ **let** $\mathcal{D}_j = \mathbf{Prove}_{\mathtt{skp}}(\Gamma' \rhd \circ \prec \rho \dashv^{\blacktriangle} \circ \prec {}_{\blacktriangle?}\sigma_j) \neq \mathbf{fail}$ **in** $\dfrac{\mathcal{D}_j \quad (\forall j \in J)}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma}\,(\cdot\mathtt{skp})$

else if $\rho =_{\blacktriangle} (\bigoplus_{i \in I} \overline{a}_i.\rho_i \oplus \bigoplus_{l \in L} \overline{a}_l.\rho_l)$ **and** $\sigma =_{\blacktriangle} (\sum_{j \in J} a_j.\sigma_j)$ **and** $\mathcal{A}^{\oplus}(\rho) \cap \mathcal{A}^{+}(\sigma) = \emptyset$ **then**

  let $\Gamma' = \Gamma, \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma$ **in**

    foreach $i \in I$ **let** $\mathcal{D}_i = \mathbf{Prove}_{\mathtt{skp}}(\Gamma' \rhd \circ \prec {}_{\blacktriangle?}\rho_i \dashv^{\blacktriangle} \circ \prec \sigma) \neq \mathbf{fail}$ **in** $\dfrac{\mathcal{D}_i \quad (\forall i \in I)}{\Gamma \rhd \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma}\,(\mathtt{skp}\cdot)$

else fail

Fig. 5. The algorithm $\mathbf{Prove}_{\mathtt{skp}}$

Definition 6.7 We write

- $\models^{\mathtt{skp}} \Gamma$ if $\zeta' \prec \rho' \dashv\!\vdash^{\blacktriangle\mathtt{skp}} \chi' \prec \sigma'$ for all $\zeta' \prec \rho' \dashv^{\blacktriangle}_{\mathtt{skp}} \chi' \prec \sigma' \in \Gamma$;
- $\Gamma \models^{\mathtt{skp}} \zeta \prec \rho \dashv^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$ if $\models^{\mathtt{skp}} \Gamma$ implies $\zeta \prec \rho \dashv\!\vdash^{\blacktriangle\mathtt{skp}} \chi \prec \sigma$.

Definition 6.8 Let $k \geq 0$. We write

- $\models^{\mathtt{skp}}_{k} \Gamma$ if $\zeta' \prec \rho' \dashv\!\vdash^{\blacktriangle\mathtt{skp}}_{k} \chi' \prec \sigma'$ for all $\zeta' \prec \rho' \dashv^{\blacktriangle} \chi' \prec \sigma' \in \Gamma$,
- $\Gamma \models^{\mathtt{skp}}_{k} \zeta \prec \rho \dashv^{\blacktriangle} \chi \prec \sigma$ if $\models^{\mathtt{skp}}_{k} \Gamma$ implies $\zeta \prec \rho \dashv\!\vdash^{\blacktriangle\mathtt{skp}}_{k} \chi \prec \sigma$,

Theorem 6.9 (Soundness) *If* $\Gamma \rhd \zeta \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$, *then* $\Gamma \models \zeta \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$.

Proof Again we have $\Gamma \models^{\mathtt{skp}}_k \zeta \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$ for all k implies $\Gamma \models \zeta \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$, so that it suffices to prove that $\Gamma \rhd \zeta \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$ implies $\Gamma \models^{\mathtt{skp}}_k \zeta \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$ for all k, by simultaneous induction over the derivation $\mathcal{D}$ of $\Gamma \rhd \zeta \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$ and over k.

The argument is similar to the proof of Theorem 3.13 but in case the derivation ends by rules $(\cdot\mathtt{skp})$ or $(\mathtt{skp}\cdot)$. We show only the first case, since the second one is similar.

Now suppose that $\mathcal{D}$ ends by the inference:

$$\frac{\forall\, j \in J.\ \Gamma' \rhd \circ \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \circ \prec {}_{\blacktriangle}?\sigma_j \qquad \mathcal{A}^{\oplus}(\sigma) \cap \mathcal{A}^{+}(\rho) = \emptyset}{\Gamma \rhd \circ \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma}\ (\cdot\mathtt{skp})$$

where $\Gamma' = \Gamma,\ \circ \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma$ and $\rho = {}_{\blacktriangle}(\sum_{i\in I} a_i.\rho_i)$ and $\sigma = {}_{\blacktriangle}(\bigoplus_{j\in J} \overline{a}_j.\sigma_j \oplus \bigoplus_{l\in L} \overline{a}_l.\sigma_l)$. If $\models^{\mathtt{skp}}_k \Gamma$ for some $k > 0$, then $\models^{\mathtt{skp}}_{k-1} \Gamma$. By induction we have $\Gamma \models^{\mathtt{skp}}_{k-1} \circ \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma$, and hence $\circ \prec \rho \dashv\!\vert^{\blacktriangle\,\mathtt{skp}}_{k-1} \circ \prec \sigma$, that is $\models^{\mathtt{skp}}_{k-1} \Gamma'$. From this and the induction hypothesis over $\mathcal{D}$, i.e. $\Gamma' \models_{k-1} \circ \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \circ \prec {}_{\blacktriangle}?\sigma_j$ for all $j \in J$, we get $\circ \prec \rho \dashv\!\vert^{\blacktriangle\,\mathtt{skp}}_{k-1} \circ \prec {}_{\blacktriangle}?\sigma_j$. Notice that $\mathcal{A}^{\oplus}(\sigma) \cap \mathcal{A}^{+}(\rho) = \emptyset$, and $\circ \prec \rho \parallel \circ \prec {}_{\blacktriangle}?\sigma_j$ for all $j \in J$ are exactly the one step reducts of $\circ \prec \rho \parallel \circ \prec \sigma$. Then we conclude $\circ \prec \rho \dashv\!\vert^{\blacktriangle\,\mathtt{skp}}_{k} \circ \prec \sigma$, as required. □

Theorem 6.10 (Completeness) *If* $\zeta \prec \rho \dashv\!\vert^{\blacktriangle\,\mathtt{skp}} \chi \prec \sigma$, *then* $\rhd \zeta \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \chi \prec \sigma$ *is derivable.*

Proof As in the case of the calculus without skip the completeness can be shown by using an algorithm $\mathbf{Prove}_{\mathtt{skp}}$ which builds a derivation in the formal system of Fig. 4 if possible, and it fails otherwise. Figure 5 shows this algorithm. The main difference between **Prove** (Fig. 3) and $\mathbf{Prove}_{\mathtt{skp}}$ are the last two cases, which correspond to the application of the rules $(\mathtt{skp}\cdot)$ and $(\cdot\mathtt{skp})$. Clearly these cases do not destroy termination and just build the derivations which can be obtained using these rules. □

The main result of this section is that the formal system provides a complete axiomatic characterisation of the ${}^{\blacktriangle}_{\mathtt{skp}}$-compliance, which leads to an decision procedure for ${}^{\blacktriangle}_{\mathtt{skp}}$-compliance:

Theorem 6.11 *The formal system of Fig. 5 characterises* ${}^{\blacktriangle}_{\mathtt{skp}}$*-compliance, i.e.*

$$\rho \dashv\!\vert^{\blacktriangle\,\mathtt{skp}} \sigma \quad \textit{if and only if} \quad \rhd \circ \prec \rho \dashv\!\vert^{\blacktriangle}_{\mathtt{skp}} \circ \prec \sigma.$$

7. Related work and conclusion

Since the pioneering work by Danos and Krivine [DK04], reversible computations in process algebras have been widely studied. The calculus of [DK04] adds a distributed monitoring system to CCS [Mil89] allowing to rewind computations. Phillips and Ulidowski [PU07] propose a method for reversing process operators that are definable by SOS rules in a general format, using keys to bind synchronised actions together. A reversible variant of the higher-order π-calculus is defined in [LMS10], using name tags for identifying threads and explicit memory processes. In [LMSS11] Lanese et al. enrich the calculus of [LMS10] with a fine-grained rollback primitive. To the best of our knowledge the first works dealing with rollback of communicating systems are [dVKH10a, dVKH10b, KSH14]. In these papers an extension of CCS models the combination of rollback recovery and coordinated checkpoints.

As pointed out in [PU07], reversibility in process calculi is challenging, since we cannot distinguish between the processes $a \| a$ and $a.a$ by simply recording the past actions. For this reason both histories and unique identifiers for threads have been used to track information. A key requirement, dubbed *causal consistency* in [DK04], is that of undoing only actions if no other action depending on them has been executed (and not undone). Session behaviours overcome all these problems: in fact both the client and the server reduce in a sequential way. This justifies the relative simplicity of our calculus.

A work close to ours has been carried on by Tiezzi and Yoshida in [TY14, TY15], where they study the interplay between reverse computations and session-based interactions (for a comparison between session types and contracts see, e.g., [LP08]). Their calculus uses tags and memories as previous proposals in the literature on reversibility. In particular, they define the semantics for reversible sessions by adapting the approach in [LMS10], but they do not consider compliance. Compliance in a setting with rollback has been first studied in [BDdL14].

The version with *skips* of our system has been inspired by [BdL14], where the notion of standard compliance on session behaviours has been loosened, by allowing a server to skip outputs not needed by its client. In the present context both the client and the server are allowed to skip an output, but here this can be done with a different motivation, since the outputs that can be skipped have actually already been received.

Notice that a process whose behaviour is described by a session-behaviour with checkpoints is assumed to have the possibility, after a rollback, of resuming the computation following the very same branch of the computation on which the rollback has been performed. In our formalism no assumption is in fact made about the point and the motivation for a rollback. This makes our session behaviours suitable as a basis of reversible *session-structured* computations where rollbacks depends intrinsically on the single interacting processes.

From a different point of view, instead, rollbacks could be used as a strategy to get compliance. For instance assuming the interacting processes to roll back whenever the current branch of the computation cannot proceed and a different branch could work instead. This approach has been investigated in [BDLdL16], where compliance does not enjoy CONSERVATIVITY OF ERASURE but the inverse property: if behaviours without checkpoints are compliant, then an arbitrary addition of checkpoints preserves the compliance between them. More precisely the calculus of [BDLdL16] does not have checkpoints, but external choices of conames. Two external choices can be viewed as agreement points to which processes can roll back. So the previous property can be rephrased as: if behaviours without checkpoints are compliant, then the behaviours obtained by replacing some internal choices by external ones are compliant too.

Natural extensions of the present work consist in allowing to possibly perform several consecutive rollbacks and having sequences of checkpointed behaviours as pasts. It is easy to see that both extensions, even if they would not lead to great difficulties from a technical point of view, will lead instead to notions of compliance which are more demanding and less applicable in a general setting than the current one.

We plan to investigate whether our approach can be extended to multi-party sessions [HYC08], the rational being that the parallelism is limited since the interactions must follow the communication protocols prescribed by global types.

Acknowledgements

We are grateful to the anonymous reviewers of BEAT'14 and of the present submission for their useful remarks.

References

[BDdL14] Barbanera F, Dezani-Ciancaglini M, de' Liguoro U (2014) Compliance for reversible client/server interactions. In *BEAT* 162 of *EPTCS*, Open Publishing Association, pp 35–42

[BdL14] Barbanera F, de' Liguoro U (2014) Loosening the notions of compliance and sub-behaviour in client/server systems. In *ICE* 166 of *EPTCS*, Open Publishing Association, pp 94–110

[BdL15] Barbanera F, de' Liguoro U (2015) Sub-behaviour relations for session-based client/server systems. Math Struct Comp Sci 25(9):1339–1381

[BDLdL16] Barbanera F, Dezani-Ciancaglini M, Lanese I, de' Liguoro U (2016) Retractable contracts. In *PLACES*, 203 of EPTCS, Open Publishing Association, pp 61–72

[BH98] Brandt M, Henglein F (1998) Coinductive axiomatization of recursive type equality and subtyping. Fundamenta Informaticae 33(4):309–338

[BH15] Bernardi G, Hennessy M (2015) Modelling session types using contracts. Math Struct Comp Sci FirstView (9):1–51

[CCLP06] Carpineti S, Castagna G, Laneve C, Padovani L (2006) A formal account of contracts for Web Services. In *WS-FM*, number 4184 in LNCS. Springer, pp 148–162

[CGP09] Castagna G, Gesbert N, Padovani L (2009) A theory of contracts for web services. ACM Trans Program Lang Systems 31(5):19:1–19:61

[DK04] Danos V, Krivine J (2004) Reversible communicating systems. In *CONCUR* volume 3170 of *LNCS*. Springer, pp 292–307

[dVKH10a] de Vries E, Koutavas V, Hennessy M (2010) Communicating transactions - (extended abstract). In *CONCUR*, volume 6269 of *LNCS*. Springer, pp 569–583

[dVKH10b] de Vries E, Koutavas V, Hennessy M (2010) Liveness of communicating transactions—(extended abstract). In *APLAS*, volume 6461 of *LNCS*. Springer, pp 392–407

[HVK98] Honda K, Vasconcelos VT, Kubo M (1998) Language primitives and type disciplines for structured communication-based programming. In *ESOP*, volume 1381 of *LNCS*. Springer, pp 22–138

[HYC08] Honda K, Yoshida N, Carbone M (2008) Multiparty asynchronous session types. In *POPL*. ACM Press, pp 273–284

[KSH14] Koutavas V, Spaccasassi C, Hennessy M (2014) Bisimulations for communicating transactions—(extended abstract). In *FOSSACS*, volume 8412 of *LNCS*. Springer, pp 320–334

[LMS10] Lanese I, Mezzina CA, Stefani J-B (2010) Reversing higher-order pi. In *CONCUR*, volume 6269 of *LNCS*. Springer, pp 478–493

[LMSS11] Lanese I, Mezzina CA, Schmitt A, Stefani J-B (2011) Controlling reversibility in higher-order pi. In *CONCUR*, volume 6901 of *LNCS*. Springer, pp 297–311
[LP08] Laneve C, Padovani L (2008) The pairing of contracts and session types. In *Concurrency, Graphs and Models* 5065 of *LNCS*. Springer, pp 681–700
[Mil89] Milner R (1989) Communication and concurrency. PHI Series in computer science, Prentice Hall
[Pad10] Padovani L (2010) Contract-based discovery of web services modulo simple orchestrators. Theor Comp Sci 411:3328–3347
[PS96] Pierce BC, Sangiorgi D (1996) Typing and subtyping for mobile processes. Math Struct Comp Sci 6(5):409–453
[PU07] Iain CC (2007) Phillips and Irek Ulidowski. Reversing algebraic process calculi. J Logic Algeb Program 73(1-2):70–96
[TY14] Tiezzi F, Yoshida N (2014) Towards reversible sessions. In *PLACES*, volume 155 of *EPTCS*. Open Publishing Association, pp 17–24
[TY15] Tiezzi F, Yoshida N (2015) Reversible session-based pi-calculus. J Logic Algeb Methods Program 84(5):684–707

Received 1 March 2015
Accepted in revised form 22 January 2016 by Thomas Hildebrandt, Joachim Parrow, Matthias Weidlich, and Marco Carbone
Published online 24 February 2016