

# Weak Bisimulation up to Elaboration

## (long version, with full proofs)

Damien Pous

ENS Lyon

**Abstract** We study the use of the elaboration preorder (due to Arun-Kumar and Natarajan) in the framework of up-to techniques for weak bisimulation. We show that elaboration yields a correct technique that encompasses the commonly used up to expansion technique. We also define a theory of up-to techniques for elaboration that in particular validates an elaboration up to elaboration technique, while it is known that weak bisimulation up to weak bisimilarity is unsound. In this sense, the resulting setting improves over previous works in terms of modularity. Our results are obtained using nontrivial proofs that exploit termination arguments. In particular, we need the termination of internal computations for the up-to techniques to be correct. We show how this condition can be relaxed to some extent in order to handle processes exhibiting infinite internal behaviour.

## Introduction

Weak bisimilarity ( $\approx$ ) is a commonly used behavioural equivalence for the analysis of concurrent systems. *Weak* here means distinguishing between visible actions of a system, that express interactions with its environment, and  $\tau$  *transitions*, that are treated as internal moves, and hence unobservable. To prove a weak bisimilarity result, one usually exhibits a relation  $\mathcal{R}$  between states of the systems being compared, and shows that  $\mathcal{R}$  is a weak bisimulation relation (we shall often simply use ‘bisimilarity’ and ‘bisimulation’ in the sequel, and refer explicitly to the strong version of these relations when necessary).

The crux of a bisimulation proof is often the study of silent transitions, as this part of the proof expresses the fact that internal calculations do not introduce unexpected behaviours. Typically, this is where it is shown that an optimisation is valid, that an encoding is fully abstract, or that some invariant about a data structure is preserved. Because one has to take into account all possible silent transitions, this makes bisimulation relations grow a lot, although, intuitively, many of the  $\tau$  transitions being examined are irrelevant from the point of view of the overall behaviour of the system.

Several *up-to techniques* have been proposed to alleviate the task of bisimulation proofs. The idea of up-to techniques is to manipulate functions from relations to relations, that compute a form of closure. These functions are used in the bisimulation game as shown on the diagram on the left below:

$$\begin{array}{ccc}
P & \mathcal{R} & Q \\
\alpha \downarrow & & \downarrow \hat{\alpha} \\
P' & \mathcal{F}(\mathcal{R}) & Q'
\end{array}
\qquad
\begin{array}{l}
\mathcal{U} : \mathcal{R} \mapsto \mathcal{R} \cup \approx \\
\mathcal{W} : \mathcal{R} \mapsto \approx \mathcal{R} \approx
\end{array}
\qquad
\begin{array}{l}
\mathcal{X} : \mathcal{R} \mapsto \succsim \mathcal{R} \approx \\
\mathcal{E} : \mathcal{R} \mapsto \approx \mathcal{R} \approx
\end{array}$$

When the symmetric candidate relation  $\mathcal{R}$  contains a pair  $\langle P, Q \rangle$ , and  $P$  does a transition to  $P'$  along an action  $\alpha$ ,  $Q$  has to perform the same action, modulo some internal computation ( $\tau$  transitions), to yield a process  $Q'$ . The point is that unlike in the standard bisimulation game, the resulting pair  $\langle P', Q' \rangle$  has to belong to  $\mathcal{F}(\mathcal{R})$  instead of  $\mathcal{R}$  (bisimulation is obtained by taking the identity function for  $\mathcal{F}$ ).

For example, if we take for  $\mathcal{F}$  the function  $\mathcal{U}$  above, we can use known facts about  $\approx$  when examining the transitions of processes related by  $\mathcal{R}$ . More interestingly, function  $\mathcal{W}$  allows one to apply known bisimilarity laws to transform  $P'$  and  $Q'$  in order to obtain a pair belonging to  $\mathcal{R}$ . Unfortunately, the technique given by  $\mathcal{W}$  is unsound, as shown by the following standard counterexample (written in CCS): consider a process  $P$  which is not bisimilar to 0, and define  $\mathcal{R} \triangleq \{\langle \tau.P, 0 \rangle\}$ . Since  $\tau.P \approx P$ , we can use  $\mathcal{W}$  to repeatedly undo the silent transition  $\tau.P \xrightarrow{\tau} P$ , so that in the game of weak bisimulation up to weak bisimilarity, we never explore the actual behaviour of  $P$ .

$$\begin{array}{ccccc}
\tau.P & \mathcal{R} & 0 & & \\
\tau \downarrow & & & & \\
P & \approx & \tau.P & \mathcal{R} & 0 \\
& & \tau \downarrow & & \\
& & P & \approx & \tau.P & \mathcal{R} & 0 \\
& & & & \tau \downarrow & & \\
& & & & & & \dots
\end{array}$$

To address this difficulty, [9] introduces *expansion* ( $\succsim$ ), a behavioural preorder included in weak bisimilarity, that leads to the up-to technique given by function  $\mathcal{X}$  defined above. Unlike  $\mathcal{W}$ ,  $\mathcal{X}$  yields a correct proof technique, because expansion expresses a kind of efficiency constraint: intuitively, if  $P \succsim Q$ , then  $Q$  is ‘faster’ than  $P$ , in the sense that  $P$  and  $Q$  exhibit the same behaviour, but  $Q$  has to require less silent transitions to do so (we define  $\succsim$  formally below). Since  $P \succsim \tau.P$  does not hold,  $\mathcal{X}$  rules out the above counterexample.

However, as experience shows [5,7], there are cases where reasoning up to expansion does not suffice, because the silent moves one would like to factor out in a bisimulation proof are not contained in expansion. Typically, this occurs when the ‘faster process’ has to spend some time at certain points to do some internal bookkeeping, for instance to update a data structure. To go beyond expansion, we have proposed in [7] a general and, at least to some extent, modular theory of up-to techniques for weak bisimulation. [7] introduces a notion of *controlled relation*, which guarantees that a given relation can be used in place of expansion. Several sufficient conditions for a relation to be controlled are given, among which, most notably, a criterion based on a termination property that prevents the existence of what we call ‘infinite ladders’ like depicted on the diagram above (which shows an infinite  $\xrightarrow{\tau} \approx$  ladder).

Nevertheless, the resulting setting lacks flexibility, essentially because the property of being a controlled relation is not stable by union. This prevents the incremental construction of bisimulation proofs, and thus represents a drawback in terms of modularity: in this setting, extending a proof requires an involved knowledge of the up-to techniques at work, in order to check that relations remain controlled along the extension (we discuss this in more details at the end of Sect. 3).

In this paper, we focus on the *elaboration* preorder, which has been introduced in [2]. Elaboration, written  $\approx$ , is somehow the dual of expansion: informally,  $P \approx Q$  means that  $P$  performs *at least as many silent transitions* as  $Q$ , while exhibiting the same behaviour. Elaboration strictly contains expansion, and is in some sense very close to  $\approx$ . The focus in [2] is on congruence properties of  $\approx$  in the setting of CCS, and on the axiomatisation of this relation.

The first result we establish is that  $\approx$  yields a correct up-to technique for bisimulation when the system is *terminating*, that is, when it does not exhibit infinite sequences of silent transitions. Rather remarkably, this result cannot be derived by a simple diagram chasing (as is the case for the up to expansion technique). The proof relies instead on the approach of [7], the termination hypothesis being used to derive the absence of infinite ‘ladders’.

Our second contribution is to show that  $\approx$  supports the development of a modular theory of up-to techniques, along the lines of the treatment of up-to techniques for strong bisimulation presented in [8]. This represents a significant step forward w.r.t. [7] in terms of modularity, notably because the *up to transitivity* proof technique, given by  $\mathcal{T} : \mathcal{R} \mapsto \mathcal{R}^*$ , is shown to be correct for elaboration (under the previous termination hypothesis). We devote particular attention to this important result: when applicable to reason about a coinductively defined relation  $\simeq$ ,  $\mathcal{T}$  provides the powerful techniques given by  $\mathcal{R} \mapsto (\mathcal{R} \cup \simeq)^*$ , or the more restrictive (but more commonly used)  $\mathcal{R} \mapsto \simeq \mathcal{R} \simeq$ . As we show in the paper, an application of the resulting framework is the study of an *up to polyadic contexts* proof technique (a polyadic context is a context with possibly many holes in it). Establishing directly the correctness of this technique can be really tedious, while correctness of  $\mathcal{T}$  allows one to derive a modular proof that boils down to correctness in the – simpler – monadic case.

Although it can be argued that the termination of silent transitions is realistic in many systems (typically, when silent moves are used to update the internal representation of a data structure), some programming techniques may be the source of deliberate infinite internal behaviours, such as busy waiting loops. In order to be able to handle some of these situations, we move to a setting where silent transitions are decomposed into two kinds of internal moves, respectively called the *progressive* and *non-progressive* silent transitions (as in [4]). Only progressive silent transitions are supposed to be terminating. We show that under this relaxed hypothesis, the previous results can be adapted, by validating an ‘up to progressive elaboration’ technique for bisimulation, and showing the correctness of progressive elaboration up to transitivity. While being similar to the proofs of the results above, establishing the properties for non-terminating

systems involves rather intricate usages of well-founded induction. Beyond this technical aspect, we believe that the general proof pattern adopted in this paper exposes an interesting application of rewriting techniques to concurrency.

*Outline of the paper.* In Sect. 1, we introduce our notations and briefly recall the results of [7] that will be used in the sequel. In Sect. 2 we define the elaboration preorder, and establish correctness of the up to elaboration proof technique when silent transitions of the system are terminating. We develop in Sect. 3 a theory of up-to techniques for elaboration, and draw a comparison with existing techniques. We extend these results to non-terminating systems in Sect. 4, and give final remarks in Sect. 5.

## 1 Preliminaries

### 1.1 Labelled Transition Systems, Definitions

We consider labelled transition systems (LTS)  $\langle \mathcal{P}, \mathcal{L}, \rightarrow \rangle$ , with domain  $\mathcal{P}$ , labels or actions in  $\mathcal{L}$  and transition relation  $\rightarrow \subseteq \mathcal{P} \times \mathcal{L} \times \mathcal{P}$ . The elements of  $\mathcal{P}$  are called *processes* and are denoted by  $P, Q$ . Except in Sect. 4,  $\mathcal{L}$  will always implicitly contain a distinguished *silent action*, noted  $\tau$ . We let  $\alpha, \beta$  (resp.  $a, b$ ) range over actions, in  $\mathcal{L}$  (resp. *visible actions*, in  $\mathcal{L} \setminus \{\tau\}$ ). Some examples will be given using the syntax of CCS, which we suppose is known to the reader.

We let  $\mathcal{R}, \mathcal{S}, \mathcal{B}$  range over binary relations (simply called *relations* in the sequel) between processes. We denote respectively by  $\mathcal{R}^+, \mathcal{R}^=, \mathcal{R}^*$  the transitive, reflexive, transitive and reflexive closures of a relation  $\mathcal{R}$ .  $PRQ$  means  $(P, Q) \in \mathcal{R}$ . The composition of two relations  $\mathcal{R}$  and  $\mathcal{S}$ , written  $\mathcal{R}\mathcal{S}$ , is defined by  $\mathcal{R}\mathcal{S} \triangleq \{(P, Q) / \exists T (P, T) \in \mathcal{R} \text{ and } (T, Q) \in \mathcal{S}\}$ . We also define the inverse of a relation:  $\mathcal{R}^{-1} \triangleq \{(P, Q) / (Q, P) \in \mathcal{R}\}$ .  $\mathcal{I}$  is the identity relation, defined by  $\mathcal{I} \triangleq \{(P, P) / P \in \mathcal{P}\}$ . We say that  $\mathcal{R}$  *contains*  $\mathcal{S}$  (alternatively, that  $\mathcal{S}$  is contained in  $\mathcal{R}$ ), written  $\mathcal{S} \subseteq \mathcal{R}$ , if  $PSQ$  implies  $PRQ$ . Given an action  $\alpha$ , the set of transitions along  $\alpha$  induces a relation denoted by  $\xrightarrow{\alpha}$ :  $\xrightarrow{\alpha} \triangleq \{(P, Q) / (P, \alpha, Q) \in \rightarrow\}$ . Its inverse is written using a reversed arrow:  $\xleftarrow{\alpha} = (\xrightarrow{\alpha})^{-1}$ , and similarly for other forms of arrows, defined below. Finally, we call *function* a mapping from relations to relations.

**Definition 1.1 (Termination).** *A relation  $\mathcal{R}$  terminates if there is no infinite sequence  $(P_i)_{i \in \mathbb{N}}$  such that  $\forall i, P_i \mathcal{R} P_{i+1}$ .*

Such terminating relations are also called *Noetherian* in the literature. They lead to the powerful technique of proof by *well-founded induction* on which we heavily rely in the sequel. We will also make use of *lexicographic inductions*, that is, inductions based on lexicographic orders. In our case, such orders will always consist of the product of a terminating relation  $\mathcal{R}$  with the standard ordering of natural numbers:  $\langle P, n \rangle \succ \langle Q, m \rangle$  iff  $PRQ$  or  $(P = Q \text{ and } n > m)$ .

The definitions of behavioural equivalences and preorders will make use of the following *weak transition* relations.

**Definition 1.2 (Weak transitions).**

$$\hat{\alpha} \triangleq \begin{cases} \tau^= & \text{if } \alpha = \tau \\ a & \text{if } \alpha = a \in \mathcal{L} \setminus \{\tau\} \end{cases} \quad \begin{matrix} \hat{\alpha} \triangleq \tau^* \alpha \tau^* \\ \hat{\alpha} \triangleq \tau^* \hat{\alpha} \tau^* \end{matrix}$$

We can remark the following properties:  $\hat{\tau} = \tau^*$ ,  $\tau = \tau^+$ ,  $\hat{\alpha} = \alpha$  (note in particular the difference between  $\hat{\tau}$  and  $\tau$ ).

**Definition 1.3 (Evolution of relations).** Let  $\alpha$  be an action and  $\mathcal{R}, \mathcal{S}$  two relations. We say that  $\mathcal{R}$   $\alpha$ -evolves to  $\mathcal{S}$  if whenever  $PRQ$ ,  $P \xrightarrow{\alpha} P'$  implies  $Q \xrightarrow{\hat{\alpha}} Q'$  and  $P'SQ'$  for some  $Q'$ . Given two relations  $\mathcal{R}$  and  $\mathcal{S}$ , we say that:

- $\mathcal{R}$  evolves to  $\mathcal{S}$  if  $\mathcal{R}$   $\alpha$ -evolves to  $\mathcal{S}$  for all  $\alpha \in \mathcal{L}$ ,
- $\mathcal{R}$  evolves silently to  $\mathcal{S}$  if  $\mathcal{R}$   $\tau$ -evolves to  $\mathcal{S}$ ,
- $\mathcal{R}$  evolves visibly to  $\mathcal{S}$  if  $\mathcal{R}$   $a$ -evolves to  $\mathcal{S}$  for all  $a \in \mathcal{L} \setminus \{\tau\}$ .

**Definition 1.4 (Bisimulation, Bisimilarity).** Let  $\mathcal{R}$  be a relation.  $\mathcal{R}$  is a bisimulation if it is symmetric and evolves to itself. Bisimilarity, denoted by  $\approx$ , is defined as the union of all bisimulations.

The following technical lemma will be often used implicitly in the sequel.

**Lemma 1.5.** Let  $(\mathcal{R}_i)_{i \in I}$  and  $(\mathcal{S}_j)_{j \in J}$  be two families of relations, and let  $\alpha \in \mathcal{L}$  be an action. If for any  $i \in I$ , there exists  $j \in J$  such that  $\mathcal{R}_i$   $\alpha$ -evolves to  $\mathcal{S}_j$ , then  $\bigcup_{i \in I} \mathcal{R}_i$   $\alpha$ -evolves to  $\bigcup_{j \in J} \mathcal{S}_j$ .

*Proof.* Follows immediately from the definition of  $\alpha$ -evolution.  $\square$

## 1.2 Existing Up-to Techniques for Bisimulation

The following lemma will be useful in the sequel. It states correctness of reasoning up to transitivity on visible actions.

**Lemma 1.6.** Let  $\mathcal{R}$  be a relation. If  $\mathcal{R}$  evolves silently to itself, and visibly to  $\mathcal{R}^*$ , then  $\mathcal{R}^*$  evolves to itself.

*Proof.* By two successive inductions, we show that for any  $n$ ,  $\mathcal{R}^n$  evolves silently to itself, and  $\mathcal{R}^n$  evolves visibly to  $\mathcal{R}^*$  ( $\mathcal{R}^n$  is the composition of  $\mathcal{R}$  with itself,  $n$  times).  $\square$

Some important up-to techniques for bisimulation are given by the two following results which are simple reformulations of [7, Theorems 2.6 and 3.6].

**Theorem 1.7.** Let  $\mathcal{R}$  be a symmetric relation. If  $\mathcal{R}$  evolves silently to  $\approx \mathcal{R} \approx$  and visibly to  $\mathcal{R}^*$ , then  $\mathcal{R}$  is contained in bisimilarity.

**Theorem 1.8.** Let  $\mathcal{B}$  be a relation contained in bisimilarity, evolving to  $\mathcal{B}^*$ , and such that  $\mathcal{B}^+ \xrightarrow{\tau}$  terminates. If  $\mathcal{R}$  is a symmetric relation that evolves silently to  $\mathcal{B}^* \mathcal{R} \approx$  and visibly to  $\mathcal{R}^*$ , then  $\mathcal{R}$  is contained in bisimilarity.

In both cases, visible and silent transitions are treated differently, and up to transitivity is allowed on visible actions only. The difference between these two results is in the up-to technique that is allowed after a silent action: in the first case, one uses the compression preorder, written  $\succsim$  ( $\succsim$  will be defined in Sect. 2.1). This result is essentially already present in [9,10], without the transitivity on visible actions. In the second case, the up-to technique is given by a relation  $\mathcal{B}$ , which has to satisfy a termination property. In [7], the actual requirement for  $\mathcal{B}$  is to be a *controlled relation* [7, Definition 3.1], and it is shown that the conditions in the above theorem are sufficient for  $\mathcal{B}$  to be controlled.

The compression, used in Theorem 1.7, is not as involved as the sufficient condition expressed by Theorem 1.8. On the other hand, as will be discussed in Sect. 3, the technique given by the former theorem is more amenable to the incremental development of proofs than the setting of the latter.

## 2 Elaboration

### 2.1 Definition and Basic Properties

We now define elaboration, that has been introduced in the setting of CCS in [2].

**Definition 2.1 (Elaboration relation, Elaboration).** *A relation  $\mathcal{R}$  is an elaboration relation (in short, an elaboration) if whenever  $PRQ$ :*

- (i) *if  $P \xrightarrow{\alpha} P'$ , then  $Q \xrightarrow{\hat{\alpha}} Q'$  with  $P'\mathcal{R}Q'$ ,*
- (ii) *if  $Q \xrightarrow{\alpha} Q'$ , then  $P \xrightarrow{\hat{\alpha}} P'$  with  $P'\mathcal{R}Q'$ .*

Elaboration, denoted by  $\overset{\sim}{\succsim}$ , is the union of all elaboration relations.

**Proposition 2.2.** *For any elaboration relation  $\mathcal{R}$  and action  $\alpha \in \mathcal{L}$ , we have:*

$$\overset{\alpha}{\simeq} \mathcal{R} \subseteq \mathcal{R} \overset{\hat{\alpha}}{\simeq} \quad \text{and} \quad \mathcal{R} \overset{\alpha}{\simeq} \subseteq \overset{\alpha}{\simeq} \mathcal{R} .$$

*Elaboration relations are stable by union and composition.*

*Elaboration is both a preorder and an elaboration relation.*

*Proof.* The inclusions come with simple inductions, and yield almost directly to the other points.  $\square$

Note that [2] uses a reversed version of the symbol for elaboration – we adopted this choice to follow the convention in other papers about up-to techniques and behavioural preorders, notably [9].

The intuition behind elaboration is that if  $P \overset{\sim}{\succsim} Q$ , then  $P$  is able to always be *at least as slow* as  $Q$ , as expressed by clause (ii). In relation to this, we may remark that divergences blur the difference between elaboration and bisimilarity: if  $P \approx Q$ , then  $P! \tau \overset{\sim}{\succsim} Q$ . This observation suggests that elaboration is a coarse relation, rather close to  $\approx$  (see also Prop. 2.4 below).

In the case of CCS, elaboration is almost a precongruence: as for observational equivalence, we need to handle carefully the preemptive power of silent actions.

[2] defines a sound and complete axiomatisation of the largest precongruence contained in elaboration, called *conformance*, for finite processes.

We can also relate the elaboration preorder with the equivalence relations defined in [6]: *dynamic congruence* and *progressing bisimulation*, which are shown to coincide for CCS agents. The former appears to be the coarser relation that is both a bisimulation and a congruence. The latter is obtained by playing bisimulation games where both processes have to answer to a silent transition by playing at least one silent transition (clause (ii) on both sides). Dynamic congruence is thus less demanding than the elaboration preorder (it is contained in its kernel, and this inclusion is strict in CCS: consider processes  $\tau.a + \tau.a.\tau + \tau.a.\tau.\tau$  and  $\tau.a + \tau.a.\tau.\tau + \tau.a.\tau.\tau$ ).

To draw a comparison between  $\approx$  and other behavioural preorders, we recall the definition of *expansion* [9,10] (called *efficiency preorder* in [1]). A slightly coarser definition of expansion appears in [3,7], here we call it *compression* in order to avoid confusions. The difference has consequences as far as up-to techniques are concerned, as will be explained in Sect. 3.

**Definition 2.3 (Expansion, Compression).**

- Expansion, denoted by  $\succsim$ , is the largest relation such that whenever  $P \succsim Q$ ,
- if  $P \xrightarrow{\alpha} P'$ , then  $Q \xrightarrow{\hat{\alpha}} Q'$  with  $P' \succsim Q'$ ,
  - if  $Q \xrightarrow{\alpha} Q'$ , then  $P \xrightarrow{\hat{\alpha}} P'$  with  $P' \succsim Q'$ .
- Compression, denoted by  $\succcurlyeq$ , is the largest relation such that whenever  $P \succcurlyeq Q$ ,
- if  $P \xrightarrow{\alpha} P'$ , then  $Q \xrightarrow{\hat{\alpha}} Q'$  with  $P' \succcurlyeq Q'$ ,
  - if  $Q \xrightarrow{\alpha} Q'$ , then  $P \xrightarrow{\hat{\alpha}} P'$  with  $P' \succcurlyeq Q'$ .

In contrast with  $\approx$ ,  $P \succsim Q$  intuitively captures the fact that  $Q$  is able to be always faster than  $P$  (and similarly for  $P \succcurlyeq Q$ ).

**Proposition 2.4.** *In any LTS, we have  $\sim \subset \succ \subset \succsim \subset \approx$  and  $\succ \subset \succcurlyeq \subset \approx$ . Moreover, in CCS,  $a|\tau \not\stackrel{\succ}{\sim} \tau.a$  and  $a \not\stackrel{\succcurlyeq}{\sim} a|\tau$ .*

As shown by the examples above, elaboration and compression are not comparable in general. These examples can be used to make the same observation with *almost weak bisimulation* [9] or *relaxed expansion* [7] instead of compression.

**2.2 Bisimulation up to Elaboration**

In order for elaboration to yield a correct up-to technique, we need a termination hypothesis, for which we introduce the following terminology.

**Definition 2.5 ( $\alpha$ -terminating LTS).** *Let  $\mathbb{S} = \langle \mathcal{P}, \mathcal{L}, \rightarrow \rangle$  be an LTS, and  $\alpha \in \mathcal{L}$  a label of  $\mathbb{S}$ . We say that  $\mathbb{S}$  is  $\alpha$ -terminating if  $\xrightarrow{\alpha}$  terminates.*

**Lemma 2.6.** *Let  $\alpha$  be an action and  $\mathcal{R}$  a relation such that  $\mathcal{R} \xrightarrow{\alpha} \subseteq \xrightarrow{\alpha} \mathcal{R}$ . If the LTS is  $\alpha$ -terminating then  $\mathcal{R} \xrightarrow{\alpha}$  terminates.*

*Proof.* First we prove that  $\varphi(n) : \mathcal{R}^n \xrightarrow{\alpha} \subseteq \xrightarrow{\alpha} \mathcal{R}^n$  holds for any  $n$ . Then, suppose that  $\mathcal{R} \xrightarrow{\alpha}$  does not terminate: there exists an infinite sequence  $(Q_i)_{i \geq 0}$  such that  $Q_i \mathcal{R} \xrightarrow{\alpha} Q_{i+1}$ . Using  $\varphi(i)$ , we can define an infinite sequence  $(P_i)_{i \geq 0}$  such that  $P_i \xrightarrow{\alpha} P_{i+1}$  and  $P_i \mathcal{R}^i Q_i$ , which is contradictory with the termination of  $\xrightarrow{\alpha}$ .  $\square$

**Hypothesis 2.7.** In the remainder of this section, we assume that we are given a  $\tau$ -terminating LTS.

Since  $\xrightarrow{\tau} = \xrightarrow{\tau}^+$ , this hypothesis is equivalent to the termination of  $\xrightarrow{\tau}$  (a property called *convergence* in [4]). We will show in Sect. 4 how to circumvent this restriction.

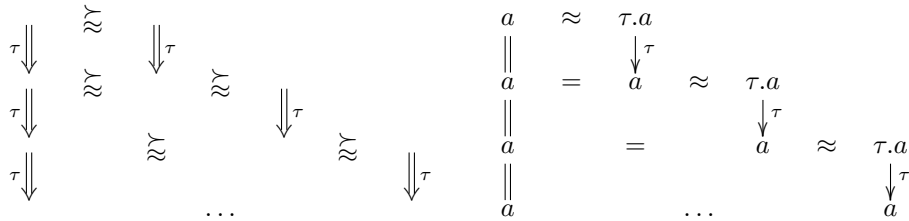
**Theorem 2.8 (Bisimilarity up to Elaboration).** *Any symmetric relation  $\mathcal{R}$  that evolves silently to  $\xrightarrow{\tau} \mathcal{R} \approx$  and visibly to  $\mathcal{R}^*$  is contained in bisimilarity.*

*Proof.* By Lemma 2.6,  $\xrightarrow{\tau} \xrightarrow{\tau}$  terminates, hence we can apply Theorem 1.8.  $\square$

We make some comments about this result and its proof. Lemma 2.6 entails that  $\xrightarrow{\tau}$  can be used in the setting proposed in [7] (it is a *controlled relation* – cf. [7]). In particular, in systems where elaboration is a precongruence, up to elaboration can be combined with the ‘up to context’ proof technique, yielding a powerful tool for bisimulation proofs.

Remarkably, and in contrast with the results in [7], the proof of Theorem 2.8 uses the fact that we work with *bisimulations* and not just simulations. Indeed, if we consider only the left-to-right part of diagrams, we are back to the standard difficulty related to weak simulation up to weak simulation (this actually can be rephrased as a problem of confluence and local confluence). Here, the central termination property, which we need to apply Theorem 1.8, is obtained using the right-to-left part of the elaboration game (when  $P$  answers a challenge offered by  $Q$ ).

To establish this termination property, we use the fact that when  $P \xrightarrow{\tau} Q$  and  $Q$  does a silent transition,  $P$  answers with at least one silent transition. As depicted on the left diagram below, any *ladder*  $(\xrightarrow{\tau} \xrightarrow{\tau})^+$  can be transformed into a *single step ladder*  $\xrightarrow{\tau} \xrightarrow{\tau}$  which is at least as high as the former. Hence termination of  $\xrightarrow{\tau}$  allows us to control the height of the ladder. By contrast, as shown on the right diagram, which recasts the counterexample seen in the introduction, in the bisimulation game the left hand side process is allowed not to move, and hence the same argument does not hold.





*Up to deterministic transitions.* Theorem 3.2 allows us to revisit a technique which has been introduced in [3, Chap. 4] in the setting of barbed bisimilarity:

**Corollary 3.3.** *If for any  $\alpha \in \mathcal{L}$ ,  $\alpha \xrightarrow{\tau} \subseteq \xrightarrow{\hat{\tau}} \hat{\alpha}$ , then  $\xrightarrow{\tau} \subseteq \xrightarrow{\hat{\tau}}$ .*

*Proof.* Relation  $\xrightarrow{\tau}$  satisfies the requirements of Theorem 3.2, and hence is an elaboration up to transitivity.  $\square$

This result gives the possibility, when  $\xrightarrow{\tau}$  is deterministic, to normalise processes w.r.t.  $\xrightarrow{\tau}$  along a bisimulation proof. [3] does not suppose  $\tau$ -termination, but requires the stronger hypothesis  $\alpha \xrightarrow{\tau} \subseteq \xrightarrow{\hat{\tau}} \hat{\alpha}$ .

We now define a class of functions corresponding to correct up-to techniques, that, as will be shown, enjoys nice compositional properties.

**Definition 3.4 (Safe function).** *A function  $\mathcal{F}$  is safe if for any relations  $\mathcal{R}$  and  $\mathcal{S}$ ,*

$$\text{if } \begin{cases} \mathcal{R} \subseteq \mathcal{S} \\ \mathcal{R} \rightsquigarrow \mathcal{S}^* \end{cases} \text{ then } \begin{cases} \mathcal{F}(\mathcal{R}) \subseteq \mathcal{F}(\mathcal{S}) \\ \mathcal{F}(\mathcal{R}) \rightsquigarrow \mathcal{F}(\mathcal{S})^* \end{cases}$$

This definition corresponds to [8, Definition 2.5]. The main difference is that we consider progressions to the reflexive transitive closures of relations. As shown in the following theorem, this makes it possible to use safe functions ‘up to transitivity’.

**Theorem 3.5 (Correctness of safe functions).** *Let  $\mathcal{F}$  be a safe function.*

*If a relation  $\mathcal{R}$  progresses to  $\mathcal{F}(\mathcal{R})^*$ , then it is contained in elaboration.*

*Proof.* Let  $\mathcal{R}_0 = \mathcal{R}$ ,  $\mathcal{R}_{n+1} = \mathcal{R}_n \cup \mathcal{F}(\mathcal{R}_n)$ ,  $\mathcal{R}_\omega = \bigcup_n \mathcal{R}_n$ . We show by induction that for any  $n$ ,  $\mathcal{R}_n \rightsquigarrow \mathcal{R}_{n+1}^*$ :

- by hypothesis,  $\mathcal{R}_0 \rightsquigarrow \mathcal{F}(\mathcal{R})^* \subseteq \mathcal{R}_1^*$ ,
- suppose that  $\mathcal{R}_n \rightsquigarrow \mathcal{R}_{n+1}^*$ , since  $\mathcal{R} \subseteq \mathcal{R}_{n+1}$ , we can use the safety of  $\mathcal{F}$ :  $\mathcal{F}(\mathcal{R}_n) \rightsquigarrow \mathcal{F}(\mathcal{R}_{n+1})^* \subseteq \mathcal{R}_{n+2}^*$ .

Hence  $\mathcal{R}_\omega \rightsquigarrow \mathcal{R}_\omega^*$ , and finally  $\mathcal{R}_\omega \subseteq \xrightarrow{\tau} \mathcal{R}_\omega^*$  using Theorem 3.2.  $\square$

The main difference with the proof of [8, Theorem 2.11] is the final application of Theorem 3.2, in order to handle the use of reflexive transitive closure.

We now show how safe functions can be combined in a modular way.

**Lemma 3.6.** *Let  $\mathcal{F}, \mathcal{G}$  be two safe functions, the following functions are safe:*

$$\begin{array}{ll} \text{Id} : \mathcal{R} \mapsto \mathcal{R} & \mathcal{F} \cup \mathcal{G} : \mathcal{R} \mapsto \mathcal{F}(\mathcal{R}) \cup \mathcal{G}(\mathcal{R}) \\ \mathcal{U} : \mathcal{R} \mapsto \xrightarrow{\tau} & \mathcal{F} \circ \mathcal{G} : \mathcal{R} \mapsto \mathcal{F}(\mathcal{G}(\mathcal{R})) \end{array}$$

By contrast with [8], composing functions using the *chaining operator*  $\mathcal{F} \frown \mathcal{G} : \mathcal{R} \mapsto \mathcal{F}(\mathcal{R})\mathcal{G}(\mathcal{R})$  does not preserve safety, essentially for the same reasons as in the weak bisimilarity case [10] (in particular,  $\tau$ -termination does not help). However, chaining can be ‘emulated’ using safe functions up to transitivity: instead of  $\mathcal{F} \frown \mathcal{G}$ , we can work with  $(\mathcal{F} \cup \mathcal{G})^*$ , which we believe provides enough flexibility for actual elaboration proofs.

*Elaboration up to context.* We now further enrich the set of up-to techniques for elaboration with an up to context technique. We call (*monadic*) *context* a mapping from processes to processes (like in [7], we adopt an approach that allows us to abstract over the details of the underlying syntax). We denote by  $C[P]$  the application of a context  $C$  to a process  $P$ . In the following technical definition, both  $\xrightarrow{\epsilon}$  and  $\xRightarrow{\epsilon}$  are synonyms for the identity relation  $\mathcal{I}$  (we suppose  $\epsilon \notin \mathcal{L}$ ).

**Definition 3.7 (Faithfulness).** *Let  $\mathcal{C}$  be a family of contexts. We say that  $\mathcal{C}$  is faithful if for all  $C \in \mathcal{C}$ , whenever  $C[P] \xrightarrow{\alpha} R$ , there are  $C' \in \mathcal{C}$ ,  $P' \in \mathcal{P}$  and  $\delta \in \mathcal{L} \cup \{\epsilon\}$  such that  $R = C'[P']$  and  $P \xrightarrow{\delta} P'$ , and for any  $Q, Q'$  such that  $Q \xrightarrow{\delta} Q'$ ,  $C[Q] \xrightarrow{\alpha} C'[Q']$ .*

*A context  $C$  is faithful if it belongs to a faithful family of contexts.*

This is the direct adaptation to the weak case of the notion of faithfulness found in [8]. In CCS every *non-degenerate* context [10] is faithful; in the  $\pi$ -calculus, every *non-input guarded* context is faithful. The following proposition shows that these families of contexts yield correct up-to techniques for elaboration. The proof is very similar to the proof of the corresponding result in [10].

**Proposition 3.8 (Safety of faithful contexts).** *Let  $\mathcal{C}$  be a safe family of contexts; the following closure up to  $\mathcal{C}$  function is safe:*

$$\tilde{\mathcal{C}} : \mathcal{R} \mapsto \{(C[P], C[Q]) \mid C \in \mathcal{C} \text{ and } PRQ\} .$$

*Proof.* Let  $\mathcal{C}$  be a faithful family of contexts and suppose that  $\mathcal{R} \rightsquigarrow \mathcal{S}^*$  (1) and  $\mathcal{R} \subseteq \mathcal{S}$  (2). We have immediately  $\tilde{\mathcal{C}}(\mathcal{R}) \subseteq \tilde{\mathcal{C}}(\mathcal{S})$ , and we show  $\tilde{\mathcal{C}}(\mathcal{R}) \rightsquigarrow \tilde{\mathcal{C}}(\mathcal{S})^*$ . Suppose that  $T \tilde{\mathcal{C}}(\mathcal{R}) U$  so that by definition,  $T = C[P]$  and  $U = C[Q]$  for some context  $C \in \mathcal{C}$ , with  $PRQ$ .

- if  $U \xrightarrow{\alpha} U'$ , since  $\mathcal{C}$  is faithful, there exists  $C' \in \mathcal{C}$ , such that  $U' = C'[Q']$  with  $Q \xrightarrow{\delta} Q'$ . Using (1), there exists  $P \xrightarrow{\delta} P'$  with  $P'S^*Q'$ . Faithfulness gives  $T = C[P] \xrightarrow{\alpha} C'[P']$  and we check that  $C'[P']\tilde{\mathcal{C}}(\mathcal{R})^*C'[Q']$ .
- the case  $T \xrightarrow{\alpha} T'$  is treated similarly. □

Although we consider only monadic contexts in Prop. 3.8, Theorem 3.5 allows us to use  $\tilde{\mathcal{C}}$  transitively, thus validating the up to *polyadic* contexts technique. The modularity of this proof contrasts with the weak bisimulation setting, where correctness of this technique requires a tedious case analysis [10].

**On Modularity Properties of Up-to Techniques.** Introducing the up to elaboration proof technique enriches the existing landscape of up-to techniques for bisimulation. We have seen that this technique enjoys nice properties, allowing one to develop elaboration proofs in an incremental and modular fashion. We now study other up-to techniques from this point of view.



divergences are of course expressible, but the processes used for the modelling do not exhibit  $\tau$ -divergences.

If, on the contrary, the system we would like to reason about does contain divergences, a first approach could be to ‘tag’ non terminating silent moves and treat these as visible. However, such visible transitions must be mapped to some visible actions on the other side of the elaboration game, in order to play these in one-to-one correspondence. This of course might be too demanding in some cases, typically when divergences arise because implementing a given behaviour introduces some loops (that are not present in the original specification). In order to address such situations, we adopt an approach from [4], which consists in isolating a subset of the  $\tau$  transitions that are terminating, while still treating all  $\tau$  moves as silent.

In the following we consider a LTS where silent moves are split into two special actions:  $\{\tau_{>}, \tau_{=}\} \subseteq \mathcal{L}$ . Transitions  $\xrightarrow{\tau_{>}}$  and  $\xrightarrow{\tau_{=}}$  will respectively be called *progressive* and *non-progressive* silent transitions. *Silent transitions*, written  $\xrightarrow{\tau}$ , are defined by  $\xrightarrow{\tau} \triangleq \xrightarrow{\tau_{>}} \cup \xrightarrow{\tau_{=}}$ . Coherently,  $a, b$  will range over  $\mathcal{L} \setminus \{\tau_{>}, \tau_{=}\}$ . We recall our notations for weak transitions (Definition 1.2) below.

$$\begin{array}{lcl} \hat{a} = a & \hat{\tau} = \hat{\tau}_{>} = \hat{\tau}_{=} = \tau = & \\ \hat{\tau} = \tau^* & \hat{a} = a = \tau^* a \tau^* & \tau_{>} = \tau^* \tau_{>} \tau^* \end{array}$$

In this setting the notions of bisimulation and bisimilarity ignore the distinction between the two kinds of silent transitions (in particular, these relations do not coincide with what we would obtain by treating  $\tau_{=}$  as visible actions). The definition of elaboration is adapted so as to control progressive transitions only:

**Definition 4.1** ( $\tau_{>}$ -Expansion,  $\tau_{>}$ -Elaboration).

$\tau_{>}$ -Expansion, denoted by  $\succsim_{>}$ , is the largest relation such that when  $P \succsim_{>} Q$ ,

- (a) if  $P \xrightarrow{\alpha} P'$  then  $Q \xrightarrow{\hat{\alpha}} Q'$  with  $P' \succsim_{>} Q'$ , for any  $\alpha \in \mathcal{L}$ ,
- (i) if  $Q \xrightarrow{\alpha} Q'$  then  $P \xrightarrow{\hat{\alpha}} P'$  with  $P' \succsim_{>} Q'$ , for any  $\alpha \neq \tau_{>}$ ,
- (ii) if  $Q \xrightarrow{\tau_{>}} Q'$  then  $P \xrightarrow{\tau_{>}} P'$  with  $P' \succsim_{>} Q'$ .

$\tau_{>}$ -Elaboration, denoted by  $\approx\approx_{>}$ , is the largest relation such that when  $P \approx\approx_{>} Q$ ,

- (b) if  $P \xrightarrow{\alpha} P'$  then  $Q \xrightarrow{\hat{\alpha}} Q'$  with  $P' \approx\approx_{>} Q'$ , for any  $\alpha \in \mathcal{L}$ ,
- (i) if  $Q \xrightarrow{\alpha} Q'$  then  $P \xrightarrow{\hat{\alpha}} P'$  with  $P' \approx\approx_{>} Q'$ , for any  $\alpha \neq \tau_{>}$ ,
- (ii) if  $Q \xrightarrow{\tau_{>}} Q'$  then  $P \xrightarrow{\tau_{>}} P'$  with  $P' \approx\approx_{>} Q'$ .

The clauses (a) and (b) of these definitions correspond respectively to the left-to-right parts of the standard definitions of expansion and bisimulation. The right-to-left part ((i) and (ii) – that are identical for both  $\tau_{>}$ -Expansion and  $\tau_{>}$ -Elaboration) ensures that progressive silent transitions are ‘preserved’.

**Proposition 4.2.**  $\succsim_{>}$  and  $\approx\approx_{>}$  are preorders, and the following inclusions hold:

$$\sim \subset \succsim_{>} \subset \approx\approx_{>} \subset \approx .$$

Moreover, for any  $\alpha \neq \tau_=>$ , we have:

$$\begin{array}{ccc} (\overleftarrow{\tau})^n \succ \subseteq \succ (\overleftarrow{\tau})^n & \succ \xrightarrow{\alpha} \subseteq \xrightarrow{\alpha} \succ & \succ \xrightarrow{\tau_=>}^* \subseteq \xrightarrow{\tau_=>}^* \succ \\ \overleftarrow{\tau} \succ \subseteq \succ \overleftarrow{\tau} & \succ \xrightarrow{\alpha} \subseteq \xrightarrow{\alpha} \succ & \succ \xrightarrow{\tau_=>}^* \subseteq \xrightarrow{\tau_=>}^* \succ \end{array} .$$

*Proof.* Standard.  $\square$

**Hypothesis 4.3.** We assume in the remainder that the LTS is  $\tau_>$ -terminating.

The previous hypothesis is the natural adaptation of Hypothesis 2.7. Notice however that the termination of  $\overrightarrow{\tau_>}$  is more demanding than that of  $\overrightarrow{\tau_>}$ : non-progressive silent actions shall not ‘disturb’ the termination of progressive ones. Using Lemma 2.6, it gives the termination of  $\overrightarrow{\tau_>} \overrightarrow{\tau_>}$ , that is central to the proof of the following result:

**Theorem 4.4 (Bisimulation up to  $\tau_>$ -Elaboration).** *Let  $\mathcal{R}$  be a symmetric relation. If the following conditions hold whenever  $PRQ$ :*

- (i) if  $P \xrightarrow{\alpha} P'$  then  $Q \xrightarrow{\alpha} Q'$  with  $P' \mathcal{R}^* Q'$ ,
- (ii) if  $P \xrightarrow{\tau_>} P'$  then  $Q \xrightarrow{\hat{\tau}} Q'$  with  $P' \succ \mathcal{R} \approx Q'$ , and
- (iii) if  $P \xrightarrow{\tau_=>} P'$  then  $Q \xrightarrow{\hat{\tau}} Q'$  with  $P' \succ \mathcal{R} \approx Q'$ ,

then  $\mathcal{R}$  is contained in bisimilarity.

As expected, up to transitivity is allowed on visible transitions (i), and up to  $\tau_>$ -elaboration is supported only on progressive silent transitions (ii). Non-progressive ones just allow the use of up to  $\tau_>$ -expansion (iii).

*Proof (of Theorem 4.4).* We show that the symmetric relation  $(\mathcal{R} \cup \approx)^*$  is a bisimulation. Let  $\mathcal{S} = \succ \mathcal{R} \approx$ ; we remark that  $(\mathcal{R} \cup \approx)^* = \approx \mathcal{S}^*$ , so that it is sufficient to show that  $\mathcal{S}^*$  evolves to itself. This is established by proving successively the following propositions.

$$P' \overleftarrow{\tau} PRQ \text{ entails } P' \succ \mathcal{R} \approx \overleftarrow{\tau} Q \text{ or } \begin{cases} P' \succ \mathcal{R} \approx P'' \mathcal{R} \approx \overleftarrow{\tau} Q \\ \text{with } P \xrightarrow{\tau_=>} \succ P'' \end{cases} \quad (1)$$

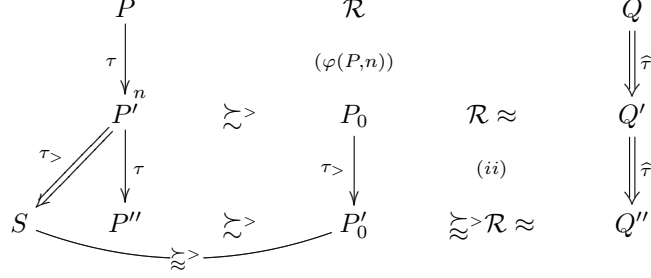
$$\overleftarrow{\tau} \mathcal{S}^* \subseteq \mathcal{S}^* \overleftarrow{\tau} \quad (2)$$

$$\overleftarrow{\tau} \mathcal{R} \subseteq \mathcal{S}^* \overleftarrow{\tau} \quad (3)$$

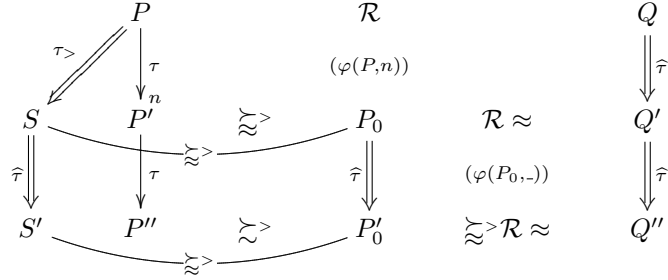
The proof of (1) goes by lexicographic induction, using the termination of  $\overrightarrow{\tau_>}$  and the predicate  $\varphi(P, n)$ : “Whenever  $P' \overleftarrow{\tau}^n PRQ$ , either (a)  $P' \succ \mathcal{R} \approx \overleftarrow{\tau} Q$  or (b)  $P' \succ \mathcal{R} \approx P_0 \mathcal{R} \approx \overleftarrow{\tau} Q$  and  $P \xrightarrow{\tau_=>} \succ P_0$ ”. The result is immediate when  $n = 0$ , for  $n + 1 > 0$ , we first use the induction hypothesis  $\varphi(P, n)$ , that yields two cases:

- (a)  $P' \succ \mathcal{R} \approx P_0 \mathcal{R} \approx \overleftarrow{\tau} Q$ . By definition of  $\succ$ , we have  $P_0 \xrightarrow{\hat{\tau}} P'_0$ . If  $P_0 \xrightarrow{\tau_>} P'_0$ , we prove (b): as represented on the diagram below, this progressive transition

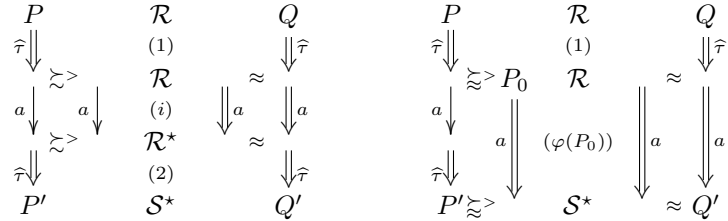
allows us to find  $S$  so that the second requirement holds. Otherwise, we easily prove (a), by using (iii) instead of (ii).



(b)  $P' \xrightarrow{\zeta} P_0 \mathcal{R} \approx \hat{\tau} Q$ , with  $P \xrightarrow{\tau} S \xrightarrow{\zeta} P_0$ . We prove (b), first we use the induction hypothesis at  $P_0$ , then the point is to transfer the second clause from  $P_0$  to  $P'_0$ , as shown on the diagram below.



We obtain (2) by standard diagram chasing arguments, and we prove (3) by well founded induction over the termination of  $\xrightarrow{\tau} \xrightarrow{\zeta}$ , using the predicate  $\psi(P)$ : “if  $P' \xrightarrow{\tau} PRQ$ , then  $P'S^*Q' \xrightarrow{\tau} Q$ ”. As shown on the diagrams below, using (1), either we can directly use (i), or we can apply the induction hypothesis, using the fact that  $P \xrightarrow{\tau} \xrightarrow{\zeta} P_0$ .



□

We now prove that both  $\tau_{>}$ -expansion and  $\tau_{>}$ -elaboration support the nice ‘up to transitivity’ technique, on visible and progressive silent steps. Since the right-to-left parts of these preorders coincide, we can factorise these proofs, using the following lemma:

**Lemma 4.5.** *Let  $\mathcal{R}$  be a relation.*

$$\text{If } \begin{cases} \mathcal{R} \xrightarrow{\alpha} \subseteq \xrightarrow{\alpha} \mathcal{R}^* \text{ for any } \alpha \neq \tau_{=} \\ \mathcal{R} \xrightarrow{\tau_{=}} \subseteq \xrightarrow{\hat{\tau}_{=}} \mathcal{R} \end{cases} \text{ then } \begin{cases} \mathcal{R}^* \xrightarrow{\alpha} \subseteq \xrightarrow{\hat{\alpha}} \mathcal{R}^* \text{ for any } \alpha \neq \tau_{>} \\ \mathcal{R}^* \xrightarrow{\tau_{>}} \subseteq \xrightarrow{\hat{\tau}_{>}} \mathcal{R}^* \end{cases} .$$

*Proof.* We prove successively the following inclusions.

$$\mathcal{R} \xrightarrow{\tau_{=}}^* \subseteq \xrightarrow{\hat{\tau}_{=}} \mathcal{R} \quad (1)$$

$$\mathcal{R}^* \xrightarrow{\tau_{=}}^* \subseteq \xrightarrow{\hat{\tau}_{=}} \mathcal{R}^* \quad (2)$$

$$\mathcal{R}^* \xrightarrow{\tau_{>}} \subseteq \xrightarrow{\hat{\tau}_{>}} \mathcal{R}^* \quad (3)$$

$$\mathcal{R}^* \xrightarrow{a} \subseteq \xrightarrow{\hat{a}} \mathcal{R}^* \quad (4)$$

(1) comes from a simple induction, we prove (2) and (3) simultaneously, with a lexicographic induction, using the termination of  $\xrightarrow{\tau_{>}}$  and the predicate  $\varphi(P, n)$ : “whenever  $P\mathcal{R}^n Q$ , if  $Q \xrightarrow{\tau_{=}}^* Q'$  then  $P \xrightarrow{\hat{\tau}_{=}} P'\mathcal{R}^* Q'$ , and if  $Q \xrightarrow{\tau_{>}} Q'$  then  $P \xrightarrow{\hat{\tau}_{>}} P'\mathcal{R}^* Q'$ ”.

$$\begin{array}{ccccc} P & \mathcal{R}^n & & \mathcal{R} & Q \\ \tau_{>} \Downarrow & (\varphi(P, n)) & \hat{\tau}_{=} \Downarrow & (1) & \downarrow \tau_{=} \\ P_0 & \mathcal{R}^* & \tau_{>} \Downarrow & \mathcal{R} & \downarrow \tau_{=}^* \\ \hat{\tau}_{=} \Downarrow & (\varphi(P_0, -)) & \downarrow \hat{\tau}_{=} & \mathcal{R}^* & \downarrow \tau_{>} \\ P' & \mathcal{R}^* & & \mathcal{R}^* & \downarrow \hat{\tau}_{=} \\ & & & & Q' \end{array} \quad \begin{array}{ccccc} P & \mathcal{R}^n & & \mathcal{R} & Q \\ \hat{\tau}_{=} \Downarrow & (\varphi(P, n)) & \downarrow \hat{\tau}_{=} & (1) & \downarrow \tau_{=} \\ P' & \mathcal{R}^* & & \mathcal{R} & \downarrow \tau_{=}^* \\ & & & & Q' \end{array}$$

Finally, we obtain (4) with another lexicographic induction, using the termination of  $\xrightarrow{\tau_{>}}$  and the predicate  $\psi(P, n)$ : “if  $P\mathcal{R}^n Q \xrightarrow{a} Q'$ , then  $P \xrightarrow{\hat{a}} P'\mathcal{R}^* Q'$ ”. The interesting cases are depicted below.

$$\begin{array}{ccccc} P & \mathcal{R} & & \mathcal{R}^n & Q \\ \tau_{>} \Downarrow & (3) & \downarrow \tau_{>} & & \downarrow a \\ P_0 & \mathcal{R}^* & \downarrow a (\psi(P, n)) & & \downarrow a \\ a \Downarrow & (\psi(P_0, -)) & \downarrow \hat{\tau}_{=} & & \downarrow a \\ \hat{\tau}_{=} \Downarrow & (2,3) & \downarrow \hat{\tau}_{=} & & \downarrow a \\ P' & \mathcal{R}^* & & \mathcal{R}^* & Q' \end{array} \quad \begin{array}{ccccc} P & \mathcal{R} & & \mathcal{R}^n & Q \\ \hat{\tau}_{=} \Downarrow & (1) & \downarrow \tau_{=} & & \downarrow a \\ P_0 & \mathcal{R} & \downarrow \tau_{=}^* & & \downarrow a (\psi(P, n)) \\ a \Downarrow & (H) & \downarrow a (\psi(P, n)) & & \downarrow a \\ \hat{\tau}_{=} \Downarrow & (2,3) & \downarrow \hat{\tau}_{=} & & \downarrow a \\ P' & \mathcal{R}^* & & \mathcal{R}^* & Q' \end{array}$$

□

**Theorem 4.6 ( $\tau_{>}$ -Elaboration up to Transitivity).** *Let  $\mathcal{R}$  be a relation. If the following conditions hold whenever  $PRQ$ :*

- (i) *if  $P \xrightarrow{\alpha} P'$  then  $Q \xrightarrow{\hat{\alpha}} Q'$  with  $P'\mathcal{R}^* Q'$ , for any  $\alpha \neq \tau_{=}$ ,*
- (ii) *if  $P \xrightarrow{\tau_{=}} P'$  then  $Q \xrightarrow{\hat{\tau}_{=}} Q'$  with  $P'\mathcal{R} Q'$ ,*
- (iii) *if  $Q \xrightarrow{\alpha} Q'$  then  $P \xrightarrow{\hat{\alpha}} P'$  with  $P'\mathcal{R}^* Q'$ , for any  $\alpha \neq \tau_{=}$ , and*

(iv) if  $Q \xrightarrow{\tau=} Q'$  then  $P \xrightarrow{\hat{\tau}} P'$  with  $P'RQ'$ ,

then  $\mathcal{R}$  is contained in  $\tau_>$ -elaboration.

*Proof.* We show that  $\mathcal{R}^*$  is a  $\tau_>$ -elaboration relation. Lemma 4.5 gives the right-to-left part, and together with Lemma 2.6, the termination of  $\mathcal{R}^* \xrightarrow{\tau_>} \hat{\mathcal{R}}$ . We prove the left-to-right part by showing the following inclusions.

$$\xrightarrow{\tau=}^* \mathcal{R} \subseteq \mathcal{R} \xleftarrow{\hat{\tau}} \quad (1)$$

$$\xleftarrow{\hat{\tau}} \mathcal{R}^* \subseteq \mathcal{R}^* \xleftarrow{\hat{\tau}} \quad (2)$$

$$\xleftarrow{a} \mathcal{R}^* \subseteq \mathcal{R}^* \xleftarrow{a} \quad (3)$$

A simple induction over (ii) gives (1). We prove (2) with a lexicographic induction, using the termination of  $\mathcal{R}^* \xrightarrow{\tau_>} \hat{\mathcal{R}}$  and the predicate  $\varphi(P, n)$ : “if  $P' \xleftarrow{\hat{\tau}} PR^nQ$  then  $P'R^*Q' \xleftarrow{\hat{\tau}} Q'$ ”. The interesting cases are depicted below.

$$\begin{array}{ccccc} P & \mathcal{R}^n & & \mathcal{R} & Q \\ \hat{\tau} \downarrow & & \downarrow \tau_=(1) & & \downarrow \hat{\tau} \\ & & \downarrow \star & \mathcal{R} & \\ & (\varphi(P, n)) & \downarrow \tau_>(i) & & \downarrow \hat{\tau} \\ & & P_0 & \mathcal{R}^* & \\ & & \downarrow \hat{\tau}(\varphi(P_0, -)) & & \downarrow \hat{\tau} \\ P' & \mathcal{R}^* & & \mathcal{R}^* & Q' \end{array} \quad \begin{array}{ccccc} P & \mathcal{R}^n & & \mathcal{R} & Q \\ \hat{\tau} \downarrow & & \downarrow \tau_=(1) & & \downarrow \hat{\tau} \\ & (\varphi(P, n)) & \downarrow \star & \mathcal{R} & \\ & & P' & \mathcal{R}^* & \\ & & & & \downarrow \hat{\tau} \\ P' & \mathcal{R}^* & & \mathcal{R} & Q' \end{array}$$

The proof of (3) is quite similar: it goes by lexicographic induction, using the termination of  $\mathcal{R}^* \xrightarrow{\tau_>} \hat{\mathcal{R}}$  and the predicate  $\psi(P, n)$ : “if  $P' \xleftarrow{a} \mathcal{R}^nQ$  then  $P'R^* \xleftarrow{a} Q'$ ”. This leads to the diagrams below.

$$\begin{array}{ccccc} P & \mathcal{R}^n & & \mathcal{R} & Q \\ a \downarrow & & \downarrow \tau_=(1) & & \downarrow \hat{\tau} \\ & & \downarrow \star & \mathcal{R} & \\ & (\psi(P, n)) & \downarrow a(i) & & \downarrow a \\ & & & \mathcal{R}^* & \\ & & \downarrow \hat{\tau}(2) & & \downarrow \hat{\tau} \\ P' & \mathcal{R}^* & & \mathcal{R}^* & Q' \end{array} \quad \begin{array}{ccccc} P & \mathcal{R}^n & & \mathcal{R} & Q \\ a \downarrow & & \downarrow \tau_>(2) & & \downarrow \hat{\tau} \\ & & P_0 & \mathcal{R}^* & \\ & (\psi(P, n)) & \downarrow a(\psi(P_0, -)) & & \downarrow a \\ & & & \mathcal{R}^* & \\ & & \downarrow \hat{\tau}(2) & & \downarrow \hat{\tau} \\ P' & \mathcal{R}^* & & \mathcal{R}^* & Q' \end{array}$$

□

**Theorem 4.7 ( $\tau_>$ -Expansion up to Transitivity).** *Let  $\mathcal{R}$  be a relation. If the following conditions hold whenever  $PRQ$ :*

- (i) if  $P \xrightarrow{\alpha} P'$  then  $Q \xrightarrow{\hat{\alpha}} Q'$  with  $P'R^*Q'$ ,
- (ii) if  $Q \xrightarrow{\alpha} Q'$  then  $P \xrightarrow{\alpha} P'$  with  $P'R^*Q'$ , for any  $\alpha \neq \tau_=($ , and
- (iii) if  $Q \xrightarrow{\tau=} Q'$  then  $P \xrightarrow{\hat{\tau}} P'$  with  $P'RQ'$ ,

then  $\mathcal{R}$  is contained in  $\tau_>$ -expansion.

*Proof.* We prove that  $\mathcal{R}^*$  is a  $\tau_>$ -expansion relation. The proof of  $\xleftarrow{a} \mathcal{R}^* \subseteq \mathcal{R}^* \xleftarrow{\hat{a}}$  is standard, and we conclude with Lemma 4.5. □

## 5 Concluding Remarks

We have proposed the new up to elaboration proof technique for bisimulation as an alternative to existing approaches. The proofs in this paper demonstrate how nontrivial termination arguments can be used to validate sophisticated proof techniques for bisimulation.

We have argued that up to elaboration offers advantages with respect to existing up-to techniques, in terms of expressiveness, flexibility or modularity. Our hope is that this technique can help addressing more complex weak bisimulation proofs. That it could be the case is suggested by the mathematical elegance of the framework we obtain, which opens the way for modular and incremental construction of proofs. This should nevertheless be confirmed by actual experiments in the study of systems involving manipulation of large bisimulation relations.

Several results in this paper suggest directions for future investigations. To enhance further our framework, it would be interesting to study how to integrate different kinds of methods in order to guarantee  $\tau$ -termination, which is necessary for the results in Sect. 2. A possible approach would be to provide a measure together with the LTS, or to adopt syntactical criteria when the LTS is given by a calculus (a process algebra). Another interesting idea in this direction is given by type systems for termination. In Sect. 4, we proposed a way to handle the case of non terminating systems. We can however think of other approaches; in particular, we would like to study LTS where non-termination of  $\xrightarrow{\tau}$  comes from cycles only, or where any state has a finite number of derivatives.

Finally, we would like to have a better understanding of the main problem of the setting of [7] (to which this paper proposes an alternative solution), namely the fact that controlled relations are not stable by union. An interesting direction would be to look for connections with the question of termination of the union of terminating rewrite systems, that has been widely studied in rewriting theory.

*Acknowledgements.* We are very thankful to Daniel Hirschhoff for his numerous comments and suggestions, and his great help during the redaction process.

## References

1. S. Arun-Kumar and M. Hennessy. An Efficiency Preorder for Processes. *Acta Informatica*, 29(9):737–760, 1992.
2. S. Arun-Kumar and V. Natarajan. Conformance: A Precongruence Close to Bisimilarity. In *Proc. Struct. in Concurrency Theory*, pages 55–68. Springer Verlag, 1995.
3. C. Fournet. *The Join-Calculus: a Calculus for Distributed Mobile Programming*. PhD thesis, Ecole Polytechnique, 1998.
4. J. Groote and M. Reniers. Algebraic Process Verification. In *Handbook of Process Algebra*, pages 1151–1208. Elsevier, 2001.
5. D. Hirschhoff, D. Pous, and D. Sangiorgi. A Correct Abstract Machine for Safe Ambients. In *Proc. COORD '05*, volume 3454 of *LNCS*. Springer Verlag, 2005.
6. U. Montanari and V. Sassone. Dynamic Congruence vs. Progressing Bisimulation for CCS. *Fundamenta Informaticae*, 16(1):171–199, 1992.

7. D. Pous. Up-to Techniques for Weak Bisimulation. In *Proc. 32th ICALP*, volume 3580 of *LNCS*, pages 730–741. Springer Verlag, 2005.
8. D. Sangiorgi. On the Bisimulation Proof Method. *Journal of Mathematical Structures in Computer Science*, 8:447–479, 1998.
9. D. Sangiorgi and R. Milner. The problem of “Weak Bisimulation up to”. In *Proc. 3rd CONCUR*, volume 630 of *LNCS*, pages 32–46. Springer Verlag, 1992.
10. D. Sangiorgi and D. Walker. *The  $\pi$ -calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.