Branching Bisimilarity for Processes with Time-Outs

Gaspard Reghem
ENS Paris-Saclay, Université Paris-Saclay, France

Rob J. van Glabbeek
School of Informatics, University of Edinburgh, UK
School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

Abstract

This paper provides an adaptation of branching bisimilarity to reactive systems with time-outs. Multiple equivalent definitions are procured, along with a modal characterisation and a proof of its congruence property for a standard process algebra with recursion. The last section presents a complete axiomatisation for guarded processes without infinite sequences of unobservable actions.

2012 ACM Subject Classification Theory of computation → Concurrency

Keywords and phrases Reactive Systems, Time-outs, Branching Bisimilarity, Modal Characterisation, Congruence, Axiomatisation

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2024.36

Related Version *Full Version*: `https://arxiv.org/abs/2408.10117` [20]

Funding Supported by Royal Society Wolfson Fellowship RSWF\R1\221008.

1 Introduction

Strong bisimilarity [17] is the default semantic equivalence on labelled transition systems (LTSs), modelling systems that move from state to state by performing discrete, uninterpreted actions. In [11], it has been generalised, under the name *strong reactive bisimilarity*, to LTSs that feature, besides the hidden action τ [17], an unobservable *time-out* action t [9], modelling the end of a time-consuming activity from which we abstract. This addition significantly increases the expressiveness of the model [10, 11].

Applied to the verification of realistic distributed systems, strong bisimilarity is too fine an equivalence, especially because it does not cater to abstraction from internal activity. *Branching bisimilarity* [13] is a variant that does abstract from internal activity, and lies at the basis of many verification toolsets [3, 6]. The present paper generalises branching bisimilarity to LTSs with time-outs, thereby combining the virtues of [11] and [13]. It supports the resulting notion of *branching reactive bisimilarity* through a modal characterisation, congruence results for a standard process algebra with recursion, and a complete axiomatisation.

The addition of the time-out action t aims at modelling the passage of time while staying in the realm of *untimed* process algebra. Here, "untimed" means that our framework does not facilitate measuring time, even though it models whether a system can pause in some state or not. We assume that the execution of any action is instantaneous; thus, time elapses in states only. The amount of time spent in a state is dictated by the interaction of the system with an external entity called its *environment*.

We call a system *reactive* if it interacts with an environment able to allow or disallow visible actions. The environment represents a user or other systems, running in parallel, which has no control over τ or t actions. If X is the set of visible actions currently allowed by the environment and the system can perform any transition labelled by an element of $X \cup \{\tau\}$ then it will perform one of those transitions immediately. When a visible action

35th International Conference on Concurrency Theory (CONCUR 2024).
Editors: Rupak Majumdar and Alexandra Silva; Article No. 36; pp. 36:1–36:22
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is performed, it triggers the environment to choose a new set of allowed actions. If the environment is allowing X and the system cannot perform any action from $X \cup \{\tau\}$, then the system is said to be *idling*. When the system idles, time-outs become executable, but the environment can also get impatient and choose a new X before any time-out occurs.

We have supposed that the environment cannot synchronise with the execution of a time-out, thus implying that, right after executing a time-out, the environment is still allowing the same set of allowed actions as before this execution. For example, the process $a.P + \mathrm{t}.(a.Q + \tau.R)$ will never reach Q because, for the time-out to happen, the environment has to block a and so $a.Q + \tau.R$ can only be reached when the environment blocks a. In this case, the τ-transition is always executed before the environment can allow a again.

Similarly, strong and branching reactive bisimilarity satisfy the process algebraic law $\tau.P + \mathrm{t}.Q = \tau.P$, essentially giving τ priority over t. Whereas this could have been formalised through an operational semantics in which the process $\tau.P + \mathrm{t}.Q$ lacks an outgoing t-transition, here, and in [11], we derive an LTS for a standard process algebra with time-outs in a way that treats t just like any other action. Instead, the priority of τ over t is implemented in the reactive bisimilarity: its says that even though the transition $\tau.P + \mathrm{t}.Q \xrightarrow{\mathrm{t}} Q$ is present in our LTS, it will never be taken. This approach is not only simpler, it also generalises better to choices like $b.P + \mathrm{t}.Q$, where the priority of b over t is conditional on the environment in which the system is placed, namely on whether or not this environment allows the b-action to occur.

From the system's perspective, the environment can be in two kinds of states: either allowing a specific set of actions, or being triggered to change. Our model does not stipulate how much time the environment takes to choose a new set of allowed actions once triggered, or even if it will ever make such a choice. Thus, the system could perform some transitions while the environment is triggered, especially those labelled τ. In our view, the most natural way to see the environment is as another system executed in parallel, while enforcing synchronisation on all visible actions. This implies that the environment allows a set X of actions when it idles in a state whose set of initial actions is X, and the environment is triggered when it is not idling, especially when it can perform a τ-transition. In this paradigm, while the environment is triggered, any action can be allowed for a brief amount of time. However, there is no reason to believe that it will necessarily settle down on a specific set. For instance, this can happen if the environment reaches a *divergence*: an infinite sequence of τ-transitions.

In [7], seven (or nine) forms of branching bisimilarity are classified; they differ only in the treatment of divergence. In the present paper we are chiefly interested in divergence-free processes, on grounds that in the intuition of [11] any sequence of τ-transitions could be executed in time zero; yet we do wish to allow infinite sequences of t-transitions. For divergence-free process all these forms of branching bisimilarity coincide. Nevertheless, we do not formally exclude divergences, and in their presence our branching reactive bisimilarity generalises the *stability respecting branching bisimilarity* of [7], which differs from the default version from [13] through the presence of Clause 2.e of Definition 1. There does not exist a plausible reactive generalisation of the default version.

Section 2 supplies the formal definition of branching reactive bisimilarity as well as its rooted version, which will be shown to be its congruence closure. It also provides equivalent definitions that reduce our bisimilarity to a non-reactive one and illustrate that branching reactive bisimilarity coincides with stability respecting branching bisimilarity in the absence of time-outs.

Section 3 gives a modal characterisation of branching reactive bisimilarity and its rooted version on an extension of the Hennessy-Milner logic. Section 4 introduces the process algebra CCSP_t^θ along with an alternative characterisation of branching reactive bisimilarity that will be used to prove that rooted branching reactive bisimilarity is a full congruence for CCSP_t^θ.

Section 5 displays a complete axiomatisation of our bisimilarity on different fragments of CCSP_t^θ. Most completeness proofs rely on standard techniques like equation merging, but the very last one uses a relatively new method called "canonical representatives".

2 Branching Reactive Bisimilarity

A *labelled transition system* (LTS) is a triple $(\mathbb{P}, Act, \rightarrow)$ with $\mathbb{P}$ a set (of *states* or *processes*), Act a set (of *actions*) and $\rightarrow \in \mathbb{P} \times Act \times \mathbb{P}$. In this paper we consider LTSs with $Act := A \uplus \{\tau, \mathrm{t}\}$, where A is a set of *visible actions*, τ is the *hidden or invisible action*, and t the *time-out action*. Let $A_\tau := A \cup \{\tau\}$. $P \xrightarrow{\alpha} P'$ stands for $(P, \alpha, P') \in \rightarrow$ and these triplets are called *transitions*. Moreover, $P \xrightarrow{(\alpha)} P'$ denotes that either $\alpha = \tau$ and $P = P'$, or $P \xrightarrow{\alpha} P'$. Furthermore, *paths* are sequences of connected transitions and $\Longrightarrow$ is the reflexive-transitive closure of $\xrightarrow{\tau}$. The set of *initial* actions of a process $P \in \mathbb{P}$ is $\mathcal{I}(P) := \{\alpha \in A_\tau \mid P \xrightarrow{\alpha}\}$. Here $P \xrightarrow{\alpha}$ means that there is a Q with $P \xrightarrow{\alpha} Q$.

▶ **Definition 1.** A *branching reactive bisimulation* is a symmetric[1] relation $\mathscr{R} \subseteq (\mathbb{P} \times \mathbb{P}) \cup (\mathbb{P} \times \mathscr{P}(A) \times \mathbb{P})$ such that, for all $P, Q \in \mathbb{P}$ and $X \subseteq A$,

1. if $\mathscr{R}(P, Q)$ then
   - **a.** if $P \xrightarrow{\alpha} P'$ with $\alpha \in A_\tau$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{(\alpha)} Q_2$ with $\mathscr{R}(P, Q_1)$ and $\mathscr{R}(P', Q_2)$,
   - **b.** for all $Y \subseteq A$, $\mathscr{R}(P, Y, Q)$;
2. if $\mathscr{R}(P, X, Q)$ then
   - **a.** if $P \xrightarrow{\tau} P'$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{(\tau)} Q_2$ with $\mathscr{R}(P, X, Q_1)$ and $\mathscr{R}(P', X, Q_2)$,
   - **b.** if $P \xrightarrow{a} P'$ with $a \in X$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{a} Q_2$ with $\mathscr{R}(P, X, Q_1)$ and $\mathscr{R}(P', Q_2)$,
   - **c.** if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ then there is a path $Q \Longrightarrow Q_0$ with $\mathscr{R}(P, Q_0)$,
   - **d.** if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there is a path $Q =: Q_0 \Longrightarrow Q_1 \xrightarrow{\mathrm{t}} Q_2 \Longrightarrow Q_3 \xrightarrow{\mathrm{t}} \dots \Longrightarrow Q_{2r-1} \xrightarrow{(\mathrm{t})} Q_{2r}$ with $r > 0$, such that $\forall i \in [0, r-1], \mathscr{R}(P, X, Q_{2i}) \wedge \mathcal{I}(Q_{2i+1}) \cap (X \cup \{\tau\}) = \emptyset$ and $\mathscr{R}(P', X, Q_{2r})$,
   - **e.** if $P \not\xrightarrow{\tau}$ then there is a path $Q \Longrightarrow Q_0 \not\xrightarrow{\tau}$.

For $P, Q \in \mathbb{P}$, if there exists a branching reactive bisimulation $\mathscr{R}$ with $\mathscr{R}(P, Q)$ (resp. $\mathscr{R}(P, X, Q)$) then P and Q are said to be *branching reactive bisimilar* (resp. *branching X-bisimilar*), which is denoted $P \leftrightarroweq_{br} Q$ (resp. $P \leftrightarroweq_{br}^X Q$).

To build the above definition, the definition of a strong reactive bisimulation [11] was modified in a branching manner [13]. Intuitively, a triplet $\mathscr{R}(P, X, Q)$ affirms that P and Q behave similarly when the environment allows (only) the set of actions in X to occur, whereas a couple $\mathscr{R}(P, Q)$ says that P and Q behave in the same way when the environment has been triggered to change. As said before, the environment can be seen as a system executed in parallel while enforcing the synchronisation of all visible actions.

Clause 1 captures the scenario of a triggered environment: if P can perform a visible or invisible action then Q has to be able to match it; and the environment can settle on a set Y of allowed actions at any moment. Time-outs are not considered because these can occur only when the system idles, and idling can happen only when the environment has stabilised on a set of allowed actions. One might notice that, in [11], the first clause was only required for invisible actions. However, there the case $\alpha \neq \tau$ is actually implied by the other clauses. If in our definition Clause 1.a were restricted to invisible actions then $\leftrightarroweq_{br}$ would not be a congruence for the parallel operator, as shown in Appendix A.

[1] meaning that $(P, Q) \in \mathscr{R} \Leftrightarrow (Q, P) \in \mathscr{R}$ and $(P, X, Q) \in \mathscr{R} \Leftrightarrow (Q, X, P) \in \mathscr{R}$

Clause 2 depicts the scenario of an environment allowing X. τ-transitions have to be matched since the environment cannot disallow them, and their execution does not trigger the environment to change. Visible actions have to be matched only if they are allowed, and their execution triggers the environment. Triggering the environment or not explains why Clause 2a matches Q_2 in a triplet and Clause 2b in a couple. If P idles (i.e. $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$) then the environment can be triggered, thus, Q has to be able to instantaneously reach a state Q_0 related to P in a triggered environment.[2] If P idles and has an outgoing time-out transition then Q has to be able to match it in a branching manner. This involves Q performing any sequence of τ and t-transitions, such that all states encountered prior to the last optional t are related to P.[3] Lastly, a stability respecting clause [7] was added for practical reasons. In Appendix A, an example shows that without it $\leftrightarrow_{br}$ would not even be an equivalence. For the important class of *divergence-free* systems, without infinite sequences $Q_0 \xrightarrow{\tau} Q_1 \xrightarrow{\tau} \dots$, Clause 2.e is easily seen to be redundant.

▶ **Lemma 2.** *Let $\mathscr{R}$ be a branching reactive bisimulation.*

1. *If $\mathscr{R}(P, X, Q)$, $P \not\xrightarrow{\tau}$ and $Q \Longrightarrow Q'$ then also $\mathscr{R}(P, X, Q')$.*
2. *If $\mathscr{R}(P, Q)$ or $\mathscr{R}(P, X, Q)$, $P \not\xrightarrow{\tau}$ and $Q \not\xrightarrow{\tau}$ then $\mathcal{I}(Q) = \mathcal{I}(P)$.*
3. *If $\mathscr{R}(P, X, Q)$, $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $Q \not\xrightarrow{\tau}$ then $\mathscr{R}(P, Q)$.*
4. *If $\mathscr{R}(P, X, Q)$ and $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ then there is a path $Q \Longrightarrow Q_0$ with $\mathscr{R}(P, Q_0)$, $Q_0 \not\xrightarrow{\tau}$ and $\mathcal{I}(Q_0) = \mathcal{I}(P)$.*

Proof.

1. This is an immediate consequence of the symmetric counterpart of Clause 2.a (where Q takes a τ-step). When that clause yields $P \Longrightarrow P_1 \xrightarrow{(\tau)} P_2$ we have $P_2 = P$.
2. This is a direct consequence of Clause 1.a or 2.b and its symmetric counterpart.
3. By Clause 2.e there is path $Q \Longrightarrow Q_0$ with $Q_0 \not\xrightarrow{\tau}$. By Claim 1 of this lemma, $\mathscr{R}(P, X, Q_0)$. Thus, by Clause 2.c there is a path $Q_0 \Longrightarrow Q_1$ with $\mathscr{R}(P, Q_1)$, but $Q_1 = Q_0 = Q$ since $Q \not\xrightarrow{\tau}$.
4. By Clause 2.e there is path $Q \Longrightarrow Q_0$ with $Q_0 \not\xrightarrow{\tau}$. By Claim 1 of this lemma, $\mathscr{R}(P, X, Q_0)$. That $\mathcal{I}(Q_0) = \mathcal{I}(P)$ and $\mathscr{R}(P, Q_0)$ follows by Claims 2 and 3 of this lemma. ◀

Definition 1 enables us to elide some time-outs. Using the process algebra notation to be formally introduced in Section 4, the processes $a.\mathrm{t}.b.0$ and $a.\mathrm{t}.\mathrm{t}.b.0$ (as well as $a.\mathrm{t}.\tau.\mathrm{t}.b.0$) are branching reactive bisimilar. Both require an unquantified positive but finite amount of rest between the actions a and b. To support this example, Clause 2.d of Definition 1 must allow a single time-out transition of one process to be matched by either zero or multiple time-outs of the other. An alternative definition, treating time-outs more like visible transitions, is obtained by replacing Clause 2.d by

2. d. if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{\mathrm{t}} Q_2$ with $\mathscr{R}(P', X, Q_2)$.

Requiring that the matching time-out is executable (i.e. $\mathcal{I}(Q_1) \cap (X \cup \{\tau\}) = \emptyset$) is not necessary here, as it is implied by the other clauses. Indeed, Lemma 2.3, which is not affected by changing Clause 2.d, implies the existence of a path $Q \Longrightarrow Q_1 \not\xrightarrow{\tau}$ such that $\mathscr{R}(P, Q_1)$ and $\mathcal{I}(Q_1) \cap (X \cup \{\tau\}) = \emptyset$. Since $Q_1 \not\xrightarrow{\tau}$, $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$, Clause 2d yields $Q_1 \xrightarrow{\mathrm{t}} Q_2$ with $\mathscr{R}(P', X, Q_2)$. This version of the definition has been studied [21] and has properties similar to $\leftrightarrow_{br}$, which are recapped in Appendix B.

---

[2] By Lemma 2.4 we can even choose Q_0 such that $Q_0 \not\xrightarrow{\tau}$, so that $\mathcal{I}(Q_0) = \mathcal{I}(P)$.

[3] Clause 2.d requires this only for states of the form Q_{2i} with $i \in [0, r{-}1]$, but by Lemma 2.1 it holds for all of them. Clause 2.c further implies that in Clause 2.d we have $\mathscr{R}(P, Q_{2i+1})$ for all $i \in [0, r{-}1]$.

In [13], branching bisimilarity is expressed in multiple equivalent ways. For practical purposes, our definition uses the semi-branching format, which is equivalent to the branching format thanks to the following lemma.

▶ **Lemma 3** (Stuttering Lemma). *Let $P, P^\dagger, P^\ddagger, Q \in \mathbb{P}$, if $P \leftrightarrow_{br} Q$, $P^\ddagger \leftrightarrow_{br} Q$ (resp. $P \leftrightarrow_{br}^{X} Q$, $P^\ddagger \leftrightarrow_{br}^{X} Q$) and $P \xrightarrow{\tau} P^\dagger \xrightarrow{\tau} P^\ddagger$ then $P^\dagger \leftrightarrow_{br} Q$ (resp. $P^\dagger \leftrightarrow_{br}^{X} Q$).*

Proof. Let $\mathscr{R}$ be a branching reactive bisimulation. Let's define $\mathscr{R}' := \mathscr{R} \cup \{(P^\dagger, Q), (Q, P^\dagger) \mid \exists P, P^\ddagger \in \mathbb{P}, P \Longrightarrow P^\dagger \Longrightarrow P^\ddagger \wedge \mathscr{R}(P,Q) \wedge \mathscr{R}(P^\ddagger, Q)\} \cup \{(P^\dagger, X, Q), (Q, X, P^\dagger) \mid \exists P, P^\ddagger \in \mathbb{P}, P \Longrightarrow P^\dagger \Longrightarrow P^\ddagger \wedge \mathscr{R}(P, X, Q) \wedge \mathscr{R}(P^\ddagger, X, Q)\}$. $\mathscr{R}'$ is symmetric by definition and $\mathscr{R}'$ is a branching reactive bisimulation, as proven in [20, Appendix E]. ◀

▶ **Proposition 4.** *$\leftrightarrow_{br}$ and $(\leftrightarrow_{br}^{X})_{X \subseteq A}$ are equivalence relations.*

Proof. Reflexivity and symmetry are trivial following the definition. For transitivity, consider two branching reactive bisimulations $\mathscr{R}_1$ and $\mathscr{R}_2$. Let's define $\mathscr{R} := (\mathscr{R}_1 \circ \mathscr{R}_2) \cup (\mathscr{R}_2 \circ \mathscr{R}_1)$. Here $\mathscr{R}_1 \circ \mathscr{R}_2 := \{(P,Q) \mid \exists R.\ \mathscr{R}(P,R) \wedge \mathscr{R}(R,Q)\} \cup \{(P,X,Q) \mid \exists R.\ \mathscr{R}(P,X,R) \wedge \mathscr{R}(R,X,Q)\}$. $\mathscr{R}$ is symmetric by definition and $\mathscr{R}$ is a branching reactive bisimulation, as proven in [20, Appendix E]. ◀

2.1 Rooted Version

A well-known limitation of branching bisimilarity $\leftrightarrow_b$ is that it fails to be a congruence for the choice operator $+$. For example, $a \leftrightarrow_b \tau.a$ but $a + b \not\leftrightarrow_b \tau.a + b$. Since the objective is to define a congruence, instead of $\leftrightarrow_{br}$ we use the *congruence closure* of $\leftrightarrow_{br}$, which is the coarsest congruence included in $\leftrightarrow_{br}$.

▶ **Definition 5.** A *rooted branching reactive bisimulation* is a symmetric relation $\mathscr{R} \subseteq (\mathbb{P} \times \mathbb{P}) \cup (\mathbb{P} \times \mathscr{P}(A) \times \mathbb{P})$ such that, for all $P, Q \in \mathbb{P}$ and $X \subseteq A$,

1. if $\mathscr{R}(P,Q)$
   a. if $P \xrightarrow{\alpha} P'$ with $\alpha \in A_\tau$ then there is a transition $Q \xrightarrow{\alpha} Q'$ with $P' \leftrightarrow_{br} Q'$,
   b. for all $Y \subseteq A$, $\mathscr{R}(P, Y, Q)$;
2. if $\mathscr{R}(P, X, Q)$
   a. if $P \xrightarrow{\tau} P'$ then there is a transition $Q \xrightarrow{\tau} Q'$ with $P' \leftrightarrow_{br}^{X} Q'$,
   b. if $P \xrightarrow{a} P'$ with $a \in X$ then there is a transition $Q \xrightarrow{a} Q'$ with $P' \leftrightarrow_{br} Q'$,
   c. if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ then $\mathscr{R}(P,Q)$,
   d. if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there is a transition $Q \xrightarrow{\mathrm{t}} Q'$ with $P' \leftrightarrow_{br}^{X} Q'$.

For $P, Q \in \mathbb{P}$, if there exists a rooted branching reactive bisimulation $\mathscr{R}$ with $\mathscr{R}(P,Q)$ (resp. $\mathscr{R}(P,X,Q)$) then P and Q are said to be *rooted branching reactive bisimilar* (resp. rooted branching X-bisimilar), which is denoted $P \leftrightarrow_{br}^{r} Q$ (resp. $P \leftrightarrow_{br}^{rX} Q$).

A rooted version of a bisimulation consists in enforcing a stricter matching on the first transition of a system. In the branching case, the first transition is matched in the strong manner. The stability respecting clause can be removed, as it is now implied by the other clauses. Rooting the bisimilarity is the standard technique to obtain its congruence closure; later $\leftrightarrow_{br}^{r}$ will be proven to be a congruence. As any branching reactive bisimulation relating $P + b$ and $Q + b$, for a fresh action b, induces a rooted branching reactive bisimulation relating P and Q, it then follows that $\leftrightarrow_{br}^{r}$ is the coarsest included in $\leftrightarrow_{br}$. Since $\leftrightarrow_{br}$ is an equivalence, the proof of Proposition 4 can be adapted to $\leftrightarrow_{br}^{r}$ in a straightforward way.

▶ **Proposition 6.** *$\leftrightarrow_{br}^{r}$ and $(\leftrightarrow_{br}^{rX})_{X \subseteq A}$ are equivalence relations.*

2.2 Alternative Forms of Definition 1

Definition 1 can be rephrased in various ways. First of all, using Requirements 1.b and 2.c, one can move Requirement 2.d from Clause 2 (dealing with triples (P, X, Q)) to Clause 1 (dealing with pairs (P, Q)), now adding a universal quantifier over X to the requirement. Next, Requirement 2.e can be copied under Clause 1. This makes Clause 1.b unnecessary, thereby obtaining a definition in which the triples (P, X, Q) are encountered only after taking a t-transition. In this form it is obvious that branching reactive bisimilarity reduces to the classical stability respecting branching bisimilarity for systems without t-transitions. We have chosen the form of Definition 1 over the above alternatives, because we believe it comes with more natural intuitions for its plausibility.

In Appendix C a further modification of Definitions 1 and 5 is proposed, called *generalised [rooted] branching reactive bisimulation*. We show that each [rooted] branching reactive bisimulation is a generalised [rooted] branching reactive bisimulation, and two systems are [rooted] branching reactive bisimilar iff they are related by a generalised [rooted] branching reactive bisimulation. This characterisation of $\leftrightarrow_{br}$ and $\leftrightarrow_{br}^{r}$ will be used in the proofs of Theorem 11 and Proposition 15.

In [19], Pohlmann introduces an encoding which maps strong reactive bisimilarity to strong bisimilarity where time-outs are considered as any visible action. This encoding in essence places a given process in a most general environment, one that features environment time-out actions t_ε, as well as actions ε_X for settling in a state that allows exactly the actions in X. This proves that reactive equivalences can be expressed as non-reactive ones at the cost of increasing the processes' size. Thus, any tool set able to work on strong bisimulation could theoretically deal with its reactive counterpart.

In Appendix D, this encoding is slightly modified to yield a similar result for branching reactive bisimulation and its rooted version, for the latter result also employing actions t_X. It appears that these modifications do not impact its effect on strong reactive bisimilarity. Since our bisimilarity has some time-out eliding properties, it is not mapped to stability respecting branching bisimilarity, but to a new bisimilarity, defined below.

▶ **Definition 7.** A t-*branching bisimulation* is a symmetric relation $\mathscr{R} \subseteq \mathbb{P} \times \mathbb{P}$ such that, for all $P, Q \in \mathbb{P}$, if $\mathscr{R}(P, Q)$ then

1. if $P \xrightarrow{\alpha} P'$ with $\alpha \in A_\tau \cup \{\mathrm{t}_\varepsilon, \varepsilon_X \mid X \subseteq A\}$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{(\alpha)} Q_2$ with $\mathscr{R}(P, Q_1)$ and $\mathscr{R}(P', Q_2)$,
2. if $P \xrightarrow{\mathrm{t}} P'$ then there is a path $Q = Q_0 \Longrightarrow Q_1 \xrightarrow{\mathrm{t}} Q_2 \Longrightarrow Q_3 \xrightarrow{\mathrm{t}} \dots \Longrightarrow Q_{2r-1} \xrightarrow{(\mathrm{t})} Q_{2r}$ with $r > 0$, such that $\forall i \in [0, 2r{-}1]$, $\mathscr{R}(P, Q_i)$ and $\mathscr{R}(P', Q_{2r})$,
3. if $P \stackrel{\tau}{\nrightarrow}$ then there is a path $Q \Longrightarrow Q_0 \stackrel{\tau}{\nrightarrow}$.

For $P, Q \in \mathbb{P}$, if there exists a t-branching bisimulation $\mathscr{R}$ with $\mathscr{R}(P, Q)$ then P and Q are said to be t-*branching bisimilar*, which is denoted $P \leftrightarrow_{\mathrm{t}b} Q$.

The encoding also sends $\leftrightarrow_{br}^{r}$ to the rooted version of $\leftrightarrow_{\mathrm{t}b}$.

▶ **Definition 8.** A *rooted* t-*branching bisimulation* is a symmetric relation $\mathscr{R} \subseteq \mathbb{P} \times \mathbb{P}$ such that, for all $P, Q \in \mathbb{P}$, if $\mathscr{R}(P, Q)$ then

1. if $P \xrightarrow{\alpha} P'$ with $\alpha \in Act \cup \{\mathrm{t}_\varepsilon, \mathrm{t}_X, \varepsilon_X \mid X \subseteq A\}$ then there is a transition $Q \xrightarrow{\alpha} Q'$ with $P' \leftrightarrow_{\mathrm{t}b} Q'$.

For $P, Q \in \mathbb{P}$, if there exists a rooted t-branching bisimulation $\mathscr{R}$ with $\mathscr{R}(P, Q)$ then P and Q are said to be *rooted* t-*branching bisimilar*, which is denoted $P \leftrightarrow_{\mathrm{t}b}^{r} Q$.

Providing a complete axiomatisation of rooted t-branching bisimilarity will be useful in the proof of completeness of the axiomatisation of rooted branching reactive bisimilarity.

3 Modal Characterisation

The Hennessy-Milner logic [15] expresses properties of the behaviour of processes in an LTS. In [11], the modality $\langle X\rangle\varphi$ was added to obtain a modal characterisation of strong reactive bisimilarity ($\leftrightarrow_r$). In order to capture branching reactive bisimilarity we add another modality $X\varphi$. To avoid confusion, $\langle X\rangle\varphi$ is renamed $\langle \mathrm{t}_X\rangle\varphi$.

▶ **Definition 9.** The class $\mathbb{L}$ of *reactive Hennessy-Milner formulas* is defined as follows, where I is an index set, $\alpha \in Act$, $a \in A$ and $X \subseteq A$,

$$\varphi ::= \top \mid \bigwedge_{i\in I} \varphi_i \mid \neg\varphi \mid \langle\alpha\rangle\varphi \mid X\varphi$$

Table 1 Semantics of $\models$ and $(\models_Y)_{Y\subseteq A}$.

$P \models \top$			$P \models_Y \top$		
$P \models \bigwedge_{i\in I}\varphi_i$	iff	$\forall i \in I, P \models \varphi_i$	$P \models_Y \bigwedge_{i\in I}\varphi_i$	iff	$\forall i \in I, P \models_Y \varphi_i$
$P \models \neg\varphi$	iff	$P \not\models \varphi$	$P \models_Y \neg\varphi$	iff	$P \not\models_Y \varphi$
$P \models \langle\alpha\rangle\varphi$	iff	$\exists P \xrightarrow{\alpha} P', P' \models \varphi$	$P \models_Y \langle\tau\rangle\varphi$	iff	$\exists P \xrightarrow{\tau} P', P' \models_Y \varphi$
			$P \models_Y \langle \mathrm{t}\rangle\varphi$	iff	$\exists P \xrightarrow{\mathrm{t}} P', P' \models_Y \varphi$
$P \models_Y \langle a\rangle\varphi$	iff	$(a \in Y \vee \mathcal{I}(P) \cap (Y \cup \{\tau\}) = \emptyset) \wedge \exists P \xrightarrow{a} P', P' \models \varphi$			
$P \models X\varphi$	iff	$\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset \wedge P \models_X \varphi$			
$P \models_Y X\varphi$	iff	$\mathcal{I}(P) \cap (X \cup Y \cup \{\tau\}) = \emptyset \wedge P \models_X \varphi$			

The satisfaction rules of $\mathbb{L}$ are given in Table 1. $P \models \varphi$ means that P satisfies φ when the environment is triggered, and $P \models_Y \varphi$ indicates that P satisfies φ when the environment allows Y. The modality $X\varphi$ expresses that a process can idle in its current state during a period in which the environment allows the actions in X, after which it behaves according to φ.

The modality $\langle \mathrm{t}_X\rangle\varphi$ from [11] can now be defined as $\langle \mathrm{t}_X\rangle\varphi := X\langle \mathrm{t}\rangle\varphi$. Write $\mathbb{L}_s$ for the fragment of $\mathbb{L}$ from [11], which includes $\langle \mathrm{t}_X\rangle\varphi$ but does not feature $X\varphi$ or $\langle \mathrm{t}\rangle\varphi$. Then the modal characterisation theorem of [11] says $\quad P \leftrightarrow_r Q \quad\Leftrightarrow\quad \forall\varphi \in \mathbb{L}_s.\ (P \models \varphi \Leftrightarrow Q \models \varphi)$.

Here we restrict attention to the fragment of $\mathbb{L}$ that includes $\langle \mathrm{t}_X\rangle\varphi$ and $X\varphi$, but not $\langle \mathrm{t}\rangle\varphi$. On this fragment $\models_Y$ is defined such that whenever $\mathcal{I}(P) \cap (Y \cup \{\tau\}) = \emptyset$ then $P \models_Y \varphi$ iff $P \models \varphi$. This is because the environment may choose to change during a period of idling.

To obtain a modal characterisation of [rooted] branching relative bisimilarity, we need a few other derived modalities. First of all, $\langle\varepsilon\rangle\varphi := \bigvee_{i\in\mathbb{N}}\langle\tau\rangle^i\varphi$. To lessen the notations, for all $\alpha \in A_\tau$, $\langle\hat{\alpha}\rangle\varphi$ denotes $\varphi \vee \langle\tau\rangle\varphi$ if $\alpha = \tau$, $\langle\alpha\rangle\varphi$ otherwise, and the modality $\langle\hat{\mathrm{t}}_X\rangle\varphi$ denotes $\langle \mathrm{t}_X\rangle\varphi \vee X\varphi$ or $X\langle\hat{\mathrm{t}}\rangle\varphi$. Moreover, $\varphi \wedge \langle\hat{\alpha}\rangle\varphi'$ is shortened to $\varphi\langle\hat{\alpha}\rangle\varphi'$. Furthermore, we define $\varphi\langle\varepsilon_X\rangle\varphi' := \bigvee_{i\in\mathbb{N}}\varphi\langle\varepsilon_X\rangle^{(i)}\varphi'$, where $\varphi\langle\varepsilon_X\rangle^{(0)}\varphi' := \langle\varepsilon\rangle(\varphi \wedge \langle\hat{\mathrm{t}}_X\rangle\varphi')$ and, for all $i > 0$, $\varphi\langle\varepsilon_X\rangle^{(i)}\varphi' := \langle\varepsilon\rangle(\varphi \wedge \langle \mathrm{t}_X\rangle(\varphi \wedge (\varphi\langle\varepsilon_X\rangle^{(i-1)}\varphi')))$. The satisfaction rules of these new modalities can be derived from the basic ones: see Table 2.

▶ **Definition 10.** The sub-classes $\mathbb{L}_b$ and $\mathbb{L}_b^r$ are defined as follows, where I is an index set, $\alpha \in A_\tau$, $X \subseteq A$, $\varphi, \varphi' \in \mathbb{L}_b$ and $\psi \in \mathbb{L}_b^r$,

$$\varphi ::= \top \mid \bigwedge_{i\in I}\varphi_i \mid \neg\varphi \mid \langle\varepsilon\rangle(\varphi\langle\hat{\alpha}\rangle\varphi') \mid \varphi\langle\varepsilon_X\rangle\varphi' \mid \langle\varepsilon\rangle\neg\langle\tau\rangle\top \qquad (\mathbb{L}_b)$$

$$\psi ::= \top \mid \bigwedge_{i\in I}\psi_i \mid \neg\psi \mid \langle\alpha\rangle\varphi \mid \langle \mathrm{t}_X\rangle\varphi \qquad (\mathbb{L}_b^r)$$

Table 2 Semantics of $\models$ and $(\models_Y)_{Y\subseteq A}$ for the derived modalities.

$P \models \langle\hat{\alpha}\rangle\varphi$	iff	$\exists P \xrightarrow{(\alpha)} P', P' \models \varphi$	$P \models_Y \langle\hat{\tau}\rangle\varphi$	iff	$\exists P \xrightarrow{(\tau)} P', P' \models_Y \varphi$
$P \models \langle\varepsilon\rangle\varphi$	iff	$\exists P \Longrightarrow P', P' \models \varphi$	$P \models_Y \langle\varepsilon\rangle\varphi$	iff	$\exists P \Longrightarrow P', P' \models_Y \varphi$
$P \models \varphi\langle\varepsilon_X\rangle\varphi'$	iff	$\exists P \Longrightarrow P_1 \xrightarrow{\mathrm{t}} P_2 \Longrightarrow P_3 \xrightarrow{\mathrm{t}} \dots \Longrightarrow P_{2r-1} \xrightarrow{(\mathrm{t})} P_{2r}$ with $r>0$, such that $\forall i \in [1,2r-1]\ P_i \models_X \varphi \wedge P_{2r} \models_X \varphi'$ and $\forall i \in [0,r-1]\ \mathcal{I}(P_{2i+1}) \cap (X \cup \{\tau\}) = \emptyset$			
$P \models_Y \varphi\langle\varepsilon_X\rangle\varphi'$	iff	$\exists P \Longrightarrow P_1 \xrightarrow{\mathrm{t}} P_2 \Longrightarrow P_3 \xrightarrow{\mathrm{t}} \dots \Longrightarrow P_{2r-1} \xrightarrow{(\mathrm{t})} P_{2r}$ with $r>0$, such that $\forall i \in [1,2r-1]\ P_i \models_X \varphi \wedge P_{2r} \models_X \varphi'$ and $\mathcal{I}(P_1) \cap (Y \cup \{\tau\}) = \emptyset \wedge \forall i \in [0,r-1]\ \mathcal{I}(P_{2i+1}) \cap (X \cup \{\tau\}) = \emptyset$			
$P \models \langle\mathrm{t}_X\rangle\varphi$	iff	$\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset \wedge \exists P \xrightarrow{\mathrm{t}} P', P' \models_X \varphi$			

The last option for $\mathbb{L}_b$, inspired by [5], is used to encompass the stability respecting Clause 2.e of Definition 1.

▶ **Theorem 11.** *Let $P,Q \in \mathbb{P}$. For all $X \subseteq A$,*

- $P \leftrightarroweq_{br} Q$ *iff* $\forall\varphi \in \mathbb{L}_b, P \models \varphi \Leftrightarrow Q \models \varphi$,
- $P \leftrightarroweq_{br}^X Q$ *iff* $\forall\varphi \in \mathbb{L}_b, P \models_X \varphi \Leftrightarrow Q \models_X \varphi$,
- $P \leftrightarroweq_{br}^r Q$ *iff* $\forall\psi \in \mathbb{L}_b^r, P \models \psi \Leftrightarrow Q \models \psi$,
- $P \leftrightarroweq_{br}^{rX} Q$ *iff* $\forall\psi \in \mathbb{L}_b^r, P \models_X \psi \Leftrightarrow Q \models_X \psi$.

Proof. ($\Rightarrow$) The four propositions are proven simultaneously by structural induction on $\mathbb{L}_b$ and $\mathbb{L}_b^r$ in [20, Appendix F].

($\Leftarrow$) Let $\equiv\ := \{(P,Q) \mid \forall\varphi \in \mathbb{L}_b, P \models \varphi \Leftrightarrow Q \models \varphi\} \cup \{(P,X,Q) \mid \forall\varphi \in \mathbb{L}_b, P \models_X \varphi \Leftrightarrow Q \models_X \varphi\}$, and $\equiv^r\ := \{(P,Q) \mid \forall\psi \in \mathbb{L}_b^r, P \models \psi \Leftrightarrow Q \models \psi\} \cup \{(P,X,Q) \mid \forall\psi \in \mathbb{L}_b^r, P \models_X \psi \Leftrightarrow Q \models_X \psi\}$. It suffices to check that $\equiv$ [resp. $\equiv^r$] is a generalised [rooted] branching reactive bisimulation. This is done in [20, Appendix F]. ◀

4 Process Algebra and Congruence

The process algebra CCSP_t^θ is composed of classical operators from the well-known process algebras CCS [17], CSP [2, 18] and ACP [1, 4], as well as the time-out action t and two *environment operators* from [11], that were added in order to enable a complete axiomatisation.

▶ **Definition 12.** Let V be a countable set of variables, the *expressions* of CCSP_t^θ are recursively defined as follows:

$$E ::= 0 \mid x \mid \alpha.E \mid E+F \mid E \parallel_S F \mid \tau_I(E) \mid \mathcal{R}(E) \mid \theta_L^U(E) \mid \psi_X(E) \mid \langle y|\mathcal{S}\rangle$$

where $x \in V$, $\alpha \in Act$, $S,I,L,U,X \subseteq A$, $L \subseteq U$, $\mathcal{R} \subseteq A \times A$, $\mathcal{S}$ is a *recursive specification*: a set of equations $\{x = \mathcal{S}_x \mid x \in V_\mathcal{S}\}$ with $V_\mathcal{S} \subseteq V$ and each $\mathcal{S}_x$ a CCSP_t^θ expression, and $y \in V_\mathcal{S}$. We require that all sets $\{b \mid (a,b) \in \mathcal{R}\}$ for $a \in A$ are finite.

0 stands for a system which cannot perform any action. The expression $\alpha.E$ represents a system that first performs α and then E. The expression $E+F$ represents a choice to behave like E or F. The parallel composition $E \parallel_S F$ synchronises the execution of E and F, but only when performing actions in S. $\tau_I(E)$ represents the system E where all actions $a \in I$ are transformed into τ. The operator $\mathcal{R}$ renames a given action $a \in A$ into a choice between all actions b with $(a,b) \in \mathcal{R}$. $\langle y|\mathcal{S}\rangle$ is the y-component of a solution of $\mathcal{S}$.

CCSP_t^θ also has two environment operators that help to develop a complete axiomatisation (like the left merge for ACP). $\theta_L^U(E)$ is the expression E plunged into an environment X such that $L \subseteq X \subseteq U$. $\theta_X^X(E)$ is denoted $\theta_X(E)$. $\psi_X(E)$ plunges E into the environment X if a time-out occurs, but, has no effect if any other action is performed. The operational semantics of CCSP$_t^\theta$ is given in Figure 1. All operators except the environment ones follow the semantics of CCS, CSP or ACP. As $\theta_L^U(E)$ simulates the expression E plunged in an environment $L \subseteq X \subseteq U$, it has no effect on τ-transitions, which do not trigger the environment. Moreover, θ_L^U restricts the ability to perform visible actions to those allowed by the environment (i.e. included in U) and performing these actions triggers the environment. However, if the expression idles (i.e. $\mathcal{I}(E) \cap (L \cup \{\tau\}) = \emptyset$) then it might trigger the environment and $\theta_L^U(E)$ acts like E. $\psi_X(E)$ supposes that time-outs are performed while the environment allows X, thus, it has no effect on actions that are not t. However, if E can perform a time-out while the environment allows X (i.e. $\mathcal{I}(E) \cap (X \cup \{\tau\}) = \emptyset$) then $\psi_X(E)$ can perform the time-out while plunging the expression in the environment X.

$$\frac{}{\alpha.x \xrightarrow{\alpha} x} \qquad \frac{x \xrightarrow{\alpha} x'}{x + y \xrightarrow{\alpha} x'} \qquad \frac{y \xrightarrow{\alpha} y'}{x + y \xrightarrow{\alpha} y'}$$

$$\frac{x \xrightarrow{a} x' \wedge \mathcal{R}(a,b)}{\mathcal{R}(x) \xrightarrow{b} \mathcal{R}(x')} \qquad \frac{x \xrightarrow{\tau} x'}{\mathcal{R}(x) \xrightarrow{\tau} \mathcal{R}(x')} \qquad \frac{x \xrightarrow{\mathrm{t}} x'}{\mathcal{R}(x) \xrightarrow{\mathrm{t}} \mathcal{R}(x')}$$

$$\frac{x \xrightarrow{\alpha} x' \wedge \alpha \notin S}{x \parallel_S y \xrightarrow{\alpha} x' \parallel_S y} \qquad \frac{y \xrightarrow{\alpha} y' \wedge \alpha \notin S}{x \parallel_S y \xrightarrow{\alpha} x \parallel_S y'} \qquad \frac{x \xrightarrow{a} x' \wedge y \xrightarrow{a} y' \wedge a \in S}{x \parallel_S y \xrightarrow{a} x' \parallel_S y'}$$

$$\frac{x \xrightarrow{\alpha} x' \wedge \alpha \notin I}{\tau_I(x) \xrightarrow{\alpha} \tau_I(x')} \qquad \frac{x \xrightarrow{a} x' \wedge a \in I}{\tau_I(x) \xrightarrow{\tau} \tau_I(x')} \qquad \frac{\langle \mathcal{S}_x | \mathcal{S} \rangle \xrightarrow{\alpha} x'}{\langle x | \mathcal{S} \rangle \xrightarrow{\alpha} x'}$$

$$\frac{x \xrightarrow{\tau} x'}{\theta_L^U(x) \xrightarrow{\tau} \theta_L^U(x')} \qquad \frac{x \xrightarrow{a} x' \wedge a \in U}{\theta_L^U(x) \xrightarrow{a} x'} \qquad \frac{x \xrightarrow{\alpha} x' \wedge \alpha \neq t}{\psi_X(x) \xrightarrow{\alpha} x'}$$

$$\frac{x \xrightarrow{\alpha} x' \wedge \mathcal{I}(x) \cap (L \cup \{\tau\}) = \emptyset}{\theta_L^U(x) \xrightarrow{\alpha} x'} \qquad \frac{x \xrightarrow{\mathrm{t}} x' \wedge \mathcal{I}(x) \cap (X \cup \{\tau\}) = \emptyset}{\psi_X(x) \xrightarrow{\mathrm{t}} \theta_X(x')}$$

Figure 1 Operational semantics of CCSP$_t^\theta$.

All $\mathcal{S}_x$ are considered to be sub-expressions of $\langle y|\mathcal{S}\rangle$. An occurrence of a variable x is *bound* in $E \in \text{CCSP}_t^\theta$ iff it occurs in a sub-expression $\langle y|\mathcal{S}\rangle$ of E such that $x \in V_\mathcal{S}$; otherwise it is *free*. An expression E is *invalid* if it has a sub-expression $\theta_L^U(F)$ or $\psi_X(F)$ such that a variable occurrence is free in F, but bound in E. An example justifying this condition can be found in [11]. The set of valid expressions of CCSP$_t^\theta$ is denoted $\mathbb{E}$. If an expression is valid and all of its variable occurrences are bound then it is *closed* and we call it a *process*; the set of processes is denoted $\mathbb{P}$.

A *substitution* is a partial function $\rho : V \rightharpoonup E$. The application $E[\rho]$ of a substitution ρ to an expression $E \in \mathbb{E}$ is the result of the simultaneous replacement, for all $x \in \text{dom}(\rho)$, of each free occurrence of x by the expression $\rho(x)$, while renaming bound variables to avoid name clashes. We write $\langle E|\mathcal{S}\rangle$ for the expression E where any $y \in V_\mathcal{S}$ is substituted by $\langle y|\mathcal{S}\rangle$.

4.1 Time-out Bisimulation

Thanks to the environment operator θ_L^U, it is possible to express our bisimilarity in a much more succinct way. Indeed, θ_X was defined so that $P \leftrightarrow_{br}^X Q$ if and only if $\theta_X(P) \leftrightarrow_{br} \theta_X(Q)$.

▶ **Definition 13.** A *branching time-out bisimulation* is a symmetric relation $\mathscr{B} \subseteq \mathbb{P} \times \mathbb{P}$ such that, for all $P, Q \in \mathbb{P}$, if $P \mathscr{B} Q$ then

1. if $P \xrightarrow{\alpha} P'$ with $\alpha \in A_\tau$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{(\alpha)} Q_2$ with $P \mathscr{B} Q_1$ and $P' \mathscr{B} Q_2$
2. if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{\mathrm{t}} Q_2 \Longrightarrow Q_3 \xrightarrow{\mathrm{t}} \ldots \Longrightarrow Q_{2r-1} \xrightarrow{(\mathrm{t})} Q_{2r}$ with $r > 0$, such that $Q_1 \stackrel{\tau}{\nrightarrow}$, $\forall i \in [1, r{-}1]$, $\theta_X(P) \mathscr{B} \theta_X(Q_{2i}) \wedge \mathcal{I}(Q_{2i+1}) \cap (X \cup \{\tau\}) = \emptyset$ and $\theta_X(P') \mathscr{B} \theta_X(Q_{2r})$
3. if $P \stackrel{\tau}{\nrightarrow}$ then there is a path $Q \Longrightarrow Q_0 \stackrel{\tau}{\nrightarrow}$.

Note that in Condition 2 above one also has $P \mathscr{B} Q_1$ and consequently $\mathcal{I}(Q_1) \cap (X \cup \{\tau\}) = \emptyset$. A rooted version of branching time-out bisimulation can be defined in the same vein.

▶ **Definition 14.** A *rooted branching time-out bisimulation* is a symmetric relation $\mathscr{B} \subseteq \mathbb{P} \times \mathbb{P}$ such that, for all $P, Q \in \mathbb{P}$ such that $P \mathscr{B} Q$,

1. if $P \xrightarrow{\alpha} P'$ with $\alpha \in A_\tau$ then there is a step $Q \xrightarrow{\alpha} Q'$ such that $P' \leftrightarrow_{br} Q'$
2. if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there is a step $Q \xrightarrow{\mathrm{t}} Q'$ such that $\theta_X(P') \leftrightarrow_{br} \theta_X(Q')$.

▶ **Proposition 15.** *Let $P, Q \in \mathbb{P}$,*

1. *$P \leftrightarrow_{br} Q$ (resp. $P \leftrightarrow_{br}^X Q$) iff there exists a branching time-out bisimulation $\mathscr{B}$ with $P \mathscr{B} Q$ (resp. $(\theta_X(P) \mathscr{B} \theta_X(Q))$),*
2. *$P \leftrightarrow_{br}^X Q$ if and only if $\theta_X(P) \leftrightarrow_{br} \theta_X(Q)$,*
3. *$P \leftrightarrow_{br}^r Q$ (resp. $P \leftrightarrow_{br}^{rX} Q$) iff there exists a rooted branching time-out bisimulation $\mathscr{B}$ with $P \mathscr{B} Q$ (resp. $(\theta_X(P) \mathscr{B} \theta_X(Q))$).*

Proof. Note that Proposition 15.2 is a trivial corollary of 15.1.

Let $\mathscr{R}$ be a [generalised rooted] branching reactive bisimulation, let's define $\mathscr{B} := \{(P,Q) \mid \mathscr{R}(P,Q)\} \cup \{(\theta_X(P), \theta_X(Q)) \mid \mathscr{R}(P,X,Q)\}$. $\mathscr{B}$ is a [rooted] branching time-out bisimulation, as proven in [20, Appendix G]. Let $\mathscr{B}$ be a [rooted] branching time-out bisimulation, let's define $\mathscr{R} = \{(P,Q) \mid P \mathscr{B} Q\} \cup \{(P,X,Q) \mid \theta_X(P) \mathscr{B} \theta_X(Q)\}$. $\mathscr{R}$ is a [rooted] generalised branching reactive bisimulation, as proven in [20, Appendix G]. ◀

Time-out bisimulations are very practical as there are no triplets to deal with anymore.

4.2 Congruence

Until now, bisimilarity was only defined between closed expressions, but any relation $\sim \subseteq \mathbb{P} \times \mathbb{P}$ can be extended to $\mathbb{E} \times \mathbb{E}$ in the following way: $E \sim F$ iff $\forall \rho : V \rightarrow \mathbb{P}$, $E[\rho] \sim F[\rho]$. It can be extended further to substitutions $\rho, \nu \in V \rightharpoonup \mathbb{E}$ by $\rho \sim \nu$ iff $\mathrm{dom}(\rho) = \mathrm{dom}(\nu)$ and $\forall x \in \mathrm{dom}(\rho)$, $\rho(x) \sim \nu(x)$.

▶ **Definition 16.** An equivalence $\sim \subseteq \mathbb{E} \times \mathbb{E}$ is a congruence for an n-ary operator f if $P_i \sim Q_i$ for all $i = 0, \ldots, n{-}1$ implies $f(P_0, ..., P_{n-1}) \sim f(Q_0, ..., Q_{n-1})$. It is a *lean congruence* if, for all $E \in \mathbb{E}$ and all $\rho, \nu \in V \rightharpoonup \mathbb{E}$ such that $\rho \sim \nu$, $E[\rho] \sim E[\nu]$. It is a *full congruence* if

1. it is a congruence for all operators in the language, and
2. for all recursive specifications $\mathcal{S}, \mathcal{S}'$ with $V_{\mathcal{S}} = V_{\mathcal{S}'}$ and $x \in V_{\mathcal{S}}$ such that $\langle x|\mathcal{S}\rangle, \langle x|\mathcal{S}'\rangle \in \mathbb{P}$, if $\forall y \in V_{\mathcal{S}}$, $\mathcal{S}_y \sim \mathcal{S}'_y$ then $\langle x|\mathcal{S}\rangle \sim \langle x|\mathcal{S}'\rangle$.

To show that $\sim$ is a lean congruence it suffices to restrict attention to closed substitutions $\rho, \nu \in V \rightarrow \mathbb{P}$, because the general property will then follow by composition of substitutions. A full congruence is a lean congruence, and a lean congruence is a congruence for all operators in the language, but both implications are strict, as shown in [8].

To show that $\leftrightarrow^r_{br}$ and $\leftrightarrow^r_{tb}$ are full congruences, it is first necessary to prove that $\leftrightarrow_{br}$ and $\leftrightarrow_{tb}$ are congruences for some of the operators of CCSP^θ_t.

▶ **Proposition 17.** *$\leftrightarrow_{br}$ and $\leftrightarrow_{tb}$ are congruences for action prefixing, parallel composition, abstraction, renaming and the environment operator θ^U_L, for all $L \subseteq U \subseteq A$.*

Proof. Let $\mathscr{B}$ be the smallest relation such that, for all $P, Q \in \mathbb{P}$,

- if $P \leftrightarrow_{br} Q$ then $P \mathscr{B} Q$;
- if $P \mathscr{B} Q$ then, for all $\alpha \in Act$, $I \subseteq A$, $\mathcal{R} \in A \times A$ and $L \subseteq U \subseteq A$, $\alpha.P \mathscr{B} \alpha.Q$, $\tau_I(P) \mathscr{B} \tau_I(Q)$, $\mathcal{R}(P) \mathscr{B} \mathcal{R}(Q)$ and $\theta^U_L(P) \mathscr{B} \theta^U_L(Q)$;
- if $P_1 \mathscr{B} Q_1$, $P_2 \mathscr{B} Q_2$ and $S \subseteq A$ then $P_1 \|_S P_2 \mathscr{B} Q_1 \|_S Q_2$.

It suffices to show that $\mathscr{B}$ is a branching time-out bisimulation up to $\leftrightarrow$, which implies $\mathscr{B} \subseteq \leftrightarrow_{br}$. A bisimulation "up to" is a notion introduced by Milner in [17]; it is commonly used when proving congruence properties. The proof uses some lemmas which were obtained in [11]. Details can be found in [20, Appendix H]. A similar proof yields the result for $\leftrightarrow_{tb}$. ◀

▶ **Theorem 18.** *$\leftrightarrow^r_{br}$ and $\leftrightarrow^r_{tb}$ are full congruences.*

Proof. Let $\mathscr{B} \subseteq \mathbb{P} \times \mathbb{P}$ be the smallest relation such that

- if $P \leftrightarrow^r_{br} Q$ then $P \mathscr{B} Q$;
- if $P_1 \mathscr{B} Q_1$ and $P_2 \mathscr{B} Q_2$ then $P_1 + P_2 \mathscr{B} Q_1 + Q_2$ and $\forall S \subseteq A$, $P_1 \|_S P_2 \mathscr{B} Q_1 \|_S Q_2$;
- if $P \mathscr{B} Q$ then $\forall \alpha \in Act$, $\alpha.P \mathscr{B} \alpha.Q$, $\forall I \subseteq A$, $\tau_I(P) \mathscr{B} \tau_I(Q)$, $\forall \mathcal{R} \subseteq A \times A$, $\mathcal{R}(P) \mathscr{B} \mathcal{R}(Q)$, $\forall L \subseteq U \subseteq A$, $\theta^U_L(P) \mathscr{B} \theta^U_L(Q)$ and $\forall X \subseteq A$, $\psi_X(P) \mathscr{B} \psi_X(Q)$;
- if $\mathcal{S}$ is a recursive specification with $z \in V_\mathcal{S}$ and $\rho, \nu \in V \setminus V_\mathcal{S} \rightarrow \mathbb{P}$ are substitutions such that $\forall x \in V \setminus V_\mathcal{S}$, $\rho(x) \mathscr{B} \nu(x)$, then $\langle z|\mathcal{S}\rangle[\rho] \mathscr{B} \langle z|\mathcal{S}\rangle[\nu]$;
- if $\mathcal{S}$ and $\mathcal{S}'$ are recursive specifications and $x \in V_\mathcal{S} = V_{\mathcal{S}'}$ with $\langle x|\mathcal{S}\rangle, \langle x|\mathcal{S}'\rangle \in \mathbb{P}$ such that $\forall y \in V_\mathcal{S}$, $\mathcal{S}_y \leftrightarrow^r_{br} \mathcal{S}'_y$, then $\langle x|\mathcal{S}\rangle \mathscr{B} \langle x|\mathcal{S}'\rangle$.

Since $\leftrightarrow^r_{br} \subseteq \mathscr{B}$, it suffices to prove that $\mathscr{B}$ is a rooted branching time-out bisimulation up to $\leftrightarrow_{br}$, as done in [20, Appendix I]. This implies $\mathscr{B} = \leftrightarrow^r_{br}$ and the definition will then give us that $\leftrightarrow^r_{br}$ is a lean congruence. Moreover, the last condition of $\mathscr{B}$ adds that it is a full congruence. A similar proof yields the result for $\leftrightarrow^r_{tb}$. ◀

5 Axiomatisation

We will provide complete axiomatisations for $\leftrightarrow^r_{br}$ and $\leftrightarrow^r_{tb}$ on various fragments of CCSP^θ_t.

5.1 Recursive Principles

The expression $\langle x|\mathcal{S}\rangle$ is intuitively defined as the x-component of the solution of $\mathcal{S}$. However, $\mathcal{S}$ could perfectly well have multiple solutions that are not bisimilar to each other. For instance, take $\mathcal{S} = \{x = x\}$; any expression is an x-component of a solution of $\mathcal{S}$. For our complete axiomatisation, we need to restrict attention to recursive specifications which have a unique solution with respect to our notion of bisimilarity. This property can be decomposed into two principles [1, 4]: the *recursive definition principle* (RDP) states that a system of recursive equations has at least one solution and the *recursive specification principle* (RSP) that it has at most one solution. The latter holds under a condition traditionally called *guardedness*.

▶ **Definition 19.** Let $\mathcal{S}$ be a recursive specification and $\sim\subseteq \mathbb{P}\times\mathbb{P}$, a *solution up to* $\sim$ of $\mathcal{S}$ is a substitution $\rho\in\mathbb{E}^{V_\mathcal{S}}$ such that $\rho\sim\mathcal{S}[\rho]$. Here ρ and $\mathcal{S}\in\mathbb{E}^{V_\mathcal{S}}$ are seen as $V_\mathcal{S}$-tuples.

In [1, 4] RDP was proven for the classical notion of strong bisimilarity $\leftrightarrow$. Since $\leftrightarrow^r_{br}$ and $\leftrightarrow^r_{tb}$ are included in $\leftrightarrow$, it holds for both of these relations as well.

▶ **Proposition 20** (RDP). *Let $\mathcal{S}$ be a recursive specification. The substitution $\rho: x\mapsto\langle x|\mathcal{S}\rangle$ for all $x\in V_\mathcal{S}$ is a solution of $\mathcal{S}$ up to $\leftrightarrow$. It is called the* default solution *of $\mathcal{S}$.*

An occurrence of a variable x in an expression E is *well-guarded* if x occurs in a subexpression $a.F$ of E, with $a\in A$. Here we do not allow τ and t as guards. An expression E is *well-guarded* if no operator τ_I occurs in E and all free occurrences of variables in E are well-guarded. A recursive specification $\mathcal{S}$ is *manifestly well-guarded* if no operator τ_I occurs in $\mathcal{S}$ and for all $x,y\in V_\mathcal{S}$ all occurrences of x in the expression $\mathcal{S}_y$ are well-guarded; it is *well-guarded* if it can be made manifestly well-guarded by repeated substitution of $\mathcal{S}_y$ for y within terms $\mathcal{S}_x$. A CCSP$^\theta_t$ process $P\in\mathbb{P}$ is *guarded* if each recursive specification occurring in E is well-guarded. It is *strongly guarded* if moreover there is no infinite path of τ and t-transitions $P_0\xrightarrow{\alpha_1}P_1\xrightarrow{\alpha_2}P_2\xrightarrow{\alpha_1}\dots$ with $\alpha_i\in\{\tau,\mathrm{t}\}$ for all $i>0$, starting in a state P_0 reachable from P.

▶ **Proposition 21** (RSP). *Let $\mathcal{S}$ be a well-guarded recursive specification and $\rho,\nu\in\mathbb{E}^{V_\mathcal{S}}$. If ρ and ν are solutions of $\mathcal{S}$ up to $\leftrightarrow^r_{br}$ (or $\leftrightarrow^r_{tb}$) then $\rho\leftrightarrow^r_{br}\nu$ (resp. $\rho\leftrightarrow^r_{tb}\nu$).*

Proof. Modifying $\mathcal{S}$ by substituting $\mathcal{S}_y$ for y within terms $\mathcal{S}_x$ with $x,y\in V_\mathcal{S}$ does not affect the set of its solutions. Hence we can restrict attention to manifestly well-guarded $\mathcal{S}$.

Thanks to the composition of substitutions, it suffices to prove the proposition when $\rho,\nu\in\mathbb{P}^{V_\mathcal{S}}$ and only variables of $V_\mathcal{S}$ can occur in $\mathcal{S}_x$ for $x\in V_\mathcal{S}$. It suffices to show that the symmetric closure of $\mathscr{B}:=\{(H[\mathcal{S}[\rho]],H[\mathcal{S}[\nu]])\mid H\in\mathbb{E}$ is without τ_I operators and with free variables from $V_\mathcal{S}$ only$\}$ is a rooted branching time-out bisimulation up to $\leftrightarrow_{br}$. Here $\mathcal{S}[\rho]\in\mathbb{P}^{V_\mathcal{S}}$ is seen as a substitution. Details can be found in [20, Appendix J]. An almost identical strategy can be applied to get RSP for $\leftrightarrow^r_{tb}$. ◀

The following lemma, whose proof can be found in [20, Appendix K], states that, when considering strongly guarded processes, eliding a time-out is independent of the set of allowed actions.

▶ **Lemma 22.** *Let P be a strongly guarded CCSP$^\theta_t$ process and $X\subseteq A$. If $\mathcal{I}(P)\cap(X\cup\{\tau\})=\emptyset$, $P\xrightarrow{\mathrm{t}}P'$ and $\theta_X(P)\leftrightarrow_{br}\theta_X(P')$ then $\forall Y\subseteq A$, $\mathcal{I}(P)\cap(Y\cup\{\tau\})=\emptyset\Rightarrow\theta_Y(P)\leftrightarrow_{br}\theta_Y(P')$.*

Actually, this lemma holds because our restriction of strong guardedness is too strong. Indeed, the equation $x=t.(a+\tau.x)$ has a single solution, but it is not well-guarded. The process $P=\langle x\mid\{x=t.(a+\tau.x)\}\rangle$ is not guarded, yet satisfies $P\xrightarrow{\mathrm{t}}P':=a+\tau.P$ and $\theta_\emptyset(P)\leftrightarrow_{br}\theta_\emptyset(P')$, while $\theta_{\{a\}}(P)\not\leftrightarrow_{br}\theta_{\{a\}}(P')$. Even if we write P as $\tau_{\{b\}}(\langle x\mid\{x=t.(a+b.x)\}\rangle$ it fails to be strongly guarded. This restriction was kept because being more precise is very challenging. For instance, the equation $x=t.(a+\tau.x)+t.a$ has multiple solutions: the default one, $\langle x\mid\{x=t.(a+\tau.x)+t.a+t.(a+t.b)\}\rangle$ and others. Notice that adding a branch $t.a$ to an equation with one solution can lead to it having multiple ones. Intuitively, there are situations where time-out contraction enables to hide the existence of other time-outs. Characterising these situations requires the use of semantic conditions that are difficult to verify, thus, making them undesirable. Moreover, applying the Pohlmann encoding to the processes in order to, then, use the axiomatisation of $\leftrightarrow^r_{tb}$ leads to the similar complications. This limitation deserves to be studied properly because it will appear for all bisimilarities authorising time-out contraction.

5.2 Axioms and Soundness

The set of axioms provided is composed of the axiomatisation of $\leftrightarroweq_r$ [11], together with three branching axioms. The *branching axiom* is well-known since it is used in the axiomatisation of rooted branching bisimilarity [13]. The t-*branching axiom* and the τ/t-*branching axiom* are newly introduced; they are the adaptation of the branching axiom to time-out contraction.

Table 3 Axiomatisation of $\leftrightarroweq^r_{br}$ and $\leftrightarroweq^r_{tb}$.

$x+(y+z)=(x+y)+z$	$\tau_I(x+y)=\tau_I(x)+\tau_I(y)$	$\mathcal{R}(x+y)=\mathcal{R}(x)+\mathcal{R}(y)$
$x+y=y+x$	$\tau_I(\alpha.x)=\alpha.\tau_I(x)$ if $\alpha\notin I$	$\mathcal{R}(\tau.x)=\tau.\mathcal{R}(x)$
$x+x=0$	$\tau_I(\alpha.x)=\tau.\tau_I(x)$ if $\alpha\in I$	$\mathcal{R}(\mathrm{t}.x)=\mathrm{t}.\mathcal{R}(x)$
$x+0=x$		$\mathcal{R}(a.x)=\sum_{\{b\mid\mathcal{R}(a,b)\}}b.\mathcal{R}(x)$

Expansion Theorem: if $P=\sum\limits_{i\in I}\alpha_i.P_i$ and $Q=\sum\limits_{j\in J}\beta_j.Q_j$ then

$$P\parallel_S Q=\sum_{i\in I,\alpha_i\notin S}(\alpha_i.P_i\parallel_S Q)+\sum_{j\in J,\beta_j\notin S}(P\parallel_S\beta_j.Q_j)+\sum_{i\in I,j\in J,\alpha_i=\beta_j\in S}\alpha_i.(P_i\parallel_S Q_j)$$

$$\alpha.(\tau.(x+y)+x)=\alpha.(x+y) \quad \textbf{(Branching Axiom)}$$

$$\alpha.(\mathrm{t}.(x+\textstyle\sum_{i\in I}\mathrm{t}.y_i)+x)=\alpha.(x+\sum_{i\in I}\mathrm{t}.y_i) \quad \textbf{(t-Branching Axiom)}$$

$$\alpha.(\tau.(x+y)+\mathrm{t}.(x+y)+x)=\alpha.(x+y) \quad \textbf{(τ/t-Branching Axiom)}$$

$\langle x|\mathcal{S}\rangle=\langle\mathcal{S}_x|\mathcal{S}\rangle$ **(RDP)** $\qquad$ $\mathcal{S}\Rightarrow x=\langle x|\mathcal{S}\rangle$ with $\mathcal{S}$ well-guarded **(RSP)**

$\theta^U_L(\sum_{i\in I}\alpha_i.x_i)=\sum_{i\in I}\alpha_i.x_i$	$(\forall i\in I,\alpha_i\notin L\cup\{\tau\})$
$\theta^U_L(x+\alpha.y+\beta.z)=\theta^U_L(x+\alpha.y)$	$(\alpha\in L\cup\{\tau\}\wedge\beta\notin U\cup\{\tau\})$
$\theta^U_L(x+\alpha.y+\beta.z)=\theta^U_L(x+\alpha.y)+\theta^U_L(\beta.z)$	$(\alpha\in L\cup\{\tau\}\wedge\beta\in U\cup\{\tau\})$
$\theta^U_L(\alpha.x)=\alpha.x$	$(\alpha\neq\tau)$
$\theta^U_L(\tau.x)=\tau.\theta^U_L(x)$	
$\psi_X(x+\alpha.y)=\psi_X(x)+\alpha.y$	$(\alpha\notin X\cup\{\tau,t\})$
$\psi_X(x+\alpha.y+\mathrm{t}.z)=\psi_X(x+\alpha.y)$	$(\alpha\in X\cup\{\tau\})$
$\psi_X(x+\alpha.y+\beta.z)=\psi_X(x+\alpha.y)+\beta.z$	$(\alpha,\beta\in X\cup\{\tau\})$
$\psi_X(\alpha.x)=\alpha.x$	$(\alpha\neq\mathrm{t})$
$\psi_X(\sum_{i\in I}\mathrm{t}.y_i)=\sum_{i\in I}\mathrm{t}.\theta_X(y_i)$	

$$(\forall X\subseteq A,\ \psi_X(x)=\psi_X(y))\Rightarrow x=y \quad \textbf{(Reactive Approximation Axiom)}$$

Let Ax^∞ be the set of all axioms in the first two rectangles in Table 3 and $Ax:=Ax^\infty\setminus\{\mathrm{RDP},\mathrm{RSP}\}$. Let Ax^∞_r be the set of all axioms in Table 3 except the τ/t-branching one and $Ax_r:=Ax^\infty_r\setminus\{\mathrm{RDP},\mathrm{RSP}\}$. The τ/t-branching axiom is removed from Ax^∞_r because the law Lτ: $\tau.x+\mathrm{t}.y=\tau.x$ can be derived from the reactive approximation axiom [11], and applying Lτ to the branching axiom yields the τ/t-branching axiom, thus making it redundant.

▶ **Proposition 23.** *Let P,Q be two CCSP$^\theta_t$ processes.*
- *If $Ax^\infty\vdash P=Q$ then $P\leftrightarroweq^r_{tb}Q$.*
- *If $Ax^\infty_r\vdash P=Q$ then $P\leftrightarroweq^r_{br}Q$.*

Proof. Since $\leftrightarroweq^r_{br}$ and $\leftrightarroweq^r_{tb}$ are congruences, it suffices to prove that each axiom is sound, meaning that replacing, in each axiom, $=$ by the desired bisimilarity and each variable by a process produces a true statement. Most of these axioms were proven to be sound for the classical notion $\leftrightarroweq$ of strong bisimilarity [17] in [11]. Thus, since both $\leftrightarroweq^r_{br}$ and $\leftrightarroweq^r_{tb}$ are included in $\leftrightarroweq$, most of them are sound for $\leftrightarroweq^r_{br}$ and $\leftrightarroweq^r_{tb}$.

Only the branching axioms, RSP and the reactive approximation axiom remain to be proven sound. The soundness of the branching axioms is trivial and the soundness of RSP is exactly Proposition 21. For the reactive approximation axiom, it suffices to show that $\mathscr{B} := \;\leftrightarroweq_{br}^{r} \cup \{(P,Q),(Q,P) \mid \forall X \subseteq A, \psi_X(P) \leftrightarroweq_{br}^{r} \psi_X(Q)\}$ is a rooted branching time-out bisimulation, as done in [20, Appendix L]. ◀

5.3 Completeness

A well-known feature of most process algebras is that the standard collection of axioms allows one to bring any guarded process expression in the following normal form [1, 4].

▶ **Definition 24.** Let P be a guarded CCSP_t^θ process. The *head-normal form* of P is $\hat{P} := \sum_{\{(\alpha,Q) \mid P \xrightarrow{\alpha} Q\}} \alpha.Q$.

In [11], it is proven that the axiomatisation of $\leftrightarroweq_r$ enables one to equate any guarded process with its head-normal form (using a definition of guardedness that is more liberal than the one employed here, with τ and t allowed as guards). Since the axiomatisation of $\leftrightarroweq_r$ is included in Ax^∞ and Ax_r^∞, this yields the property for them as well.

▶ **Lemma 25.** *Let P be a guarded $CCSP_t^\theta$ process. Then $Ax^\infty \vdash P = \hat{P}$ and $Ax_r^\infty \vdash P = \hat{P}$. Moreover, Ax or Ax_r are sufficient if P is recursion-free.*

This lemma is used extensively in the proof of the following completeness results.

▶ **Proposition 26.** *Let P, Q be two recursion-free $CCSP_t^\theta$ processes. If $P \leftrightarroweq_{br} Q$ (resp. $P \leftrightarroweq_{\mathrm{t}b} Q$) then, for all $\alpha \in Act$, $Ax_r \vdash \alpha.\hat{P} = \alpha.\hat{Q}$ (resp. $Ax \vdash \alpha.\hat{P} = \alpha.\hat{Q}$).*

Proof. The *depth* $d(p)$ of a process P is the length of the longest path starting from P. Note that it is properly defined for recursion-free processes only. The proof proceeds by induction on $\max(d(P), d(Q))$. The technique is fairly standard and the details can be found in [20, Appendix M]. ◀

▶ **Theorem 27.** *Let P, Q be two recursion-free $CCSP_t^\theta$ processes. If $P \leftrightarroweq_{br}^{r} Q$ (resp. $P \leftrightarroweq_{\mathrm{t}b}^{r} Q$) then $Ax_r \vdash P = Q$ (resp. $Ax \vdash P = Q$).*

Proof. It suffices to express both processes in their head-normal form and then to equate each pair of matching branches using Proposition 26. Details are in [20, Appendix M]. ◀

The following theorem lifts this result for $\leftrightarroweq_{\mathrm{t}b}^{r}$ from finite (recursion-free) processes to arbitrary (infinite) ones, subject to the restriction of strong guardedness.

▶ **Theorem 28.** *Let P, Q be strongly guarded $CCSP_t^\theta$ processes.
If $P \leftrightarroweq_{\mathrm{t}b}^{r} Q$ then $Ax^\infty \vdash P = Q$.*

Proof. A well-known technique called *equation merging* can be applied. Details can be found in [20, Appendix N]. ◀

5.4 Canonical Representative

Unfortunately, equation merging does not work on reactive bisimulations [11]. Thus, another technique is used [14, 16], called *canonical representatives*. The idea is to build the simplest process for each equivalence class of $\leftrightarrow_{br}^{r}$ and use them as intermediary to equate processes.

Let us denote with $\mathbb{P}^g$ the strongly guarded fragment of $\mathbb{P}$. For all $P \in \mathbb{P}^g$, $[P] := \{Q \in \mathbb{P}^g \mid P \leftrightarrow_{br} Q\}$ is the $\leftrightarrow_{br}$-equivalence class of P. $[\mathbb{P}^g]$ denotes the set of all $\leftrightarrow_{br}$-equivalence classes. Using the axiom of choice, a choice function $\chi : [\mathbb{P}^g] \to \mathbb{P}^g$ can be defined such that $\forall R \in [\mathbb{P}^g], \chi(R) \in R$. A transition relation can be defined between $\leftrightarrow_{br}$-equivalence classes:

$$\forall \alpha \in A_\tau, (R \xrightarrow{\alpha} R' \Leftrightarrow \chi(R) \Longrightarrow P_1 \xrightarrow{\alpha} P_2 \land P_1 \in R \land P_2 \in R' \land (\alpha \in A \lor R \neq R'))$$

$$R \xrightarrow{\mathrm{t}} R' \Leftrightarrow \exists X \subseteq A, r > 0, \chi(R) \Longrightarrow P_1 \xrightarrow{\mathrm{t}} P_2 \Longrightarrow P_3 \xrightarrow{\mathrm{t}} \ldots \Longrightarrow P_{2r-1} \xrightarrow{\mathrm{t}} P_{2r}$$
$$\land \forall i \in [0, r{-}1], \theta_X(P_{2i}) \in [\theta_X(\chi(R))] \land \mathcal{I}(P_{2i+1}) \cap (X \cup \{\tau\}) = \emptyset$$
$$\land P_1 \in R \land P_{2r} \in R' \land [\theta_X(\chi(R))] \neq [\theta_X(\chi(R'))]$$

All bisimulations can be extended to $\leftrightarrow_{br}$-equivalence classes. It suffices to consider the set of states $\mathbb{P}^g \uplus [\mathbb{P}^g] \uplus \{\theta_X([P]) \mid X \subseteq A \land P \in \mathbb{P}^g\}$.

▶ **Proposition 29.** *Let* $P \in \mathbb{P}^g$*,* $P \leftrightarrow_{br} [P]$*.*

Proof. It suffices to prove that $\mathscr{B} := \{(P, [P]), ([P], P) \mid P \in \mathbb{P}^g\}$ is a branching time-out bisimulation up to $\leftrightarrow_{br}$. Details can be found in [20, Appendix O]. ◀

▶ **Definition 30.** Let $P, Q \in \mathbb{P}^g$, the *canonical representative* of P and Q is a recursive specification $\mathcal{S}$ such that $V_\mathcal{S} := \{x_P, x_Q\} \cup \{x_R \mid R \in \bigcup_{P' \in \mathrm{Reach}(P) \cup \mathrm{Reach}(Q)} \mathrm{Reach}([P'])\}$, and $\forall R \in \bigcup_{P' \in \mathrm{Reach}(P) \cup \mathrm{Reach}(Q)} \mathrm{Reach}([P'])$,

$$\mathcal{S}_{x_P} := \sum_{\{(\alpha, P') \mid P \xrightarrow{\alpha} P'\}} \alpha.x_{[P']} \;;\; \mathcal{S}_{x_Q} := \sum_{\{(\alpha, Q') \mid Q \xrightarrow{\alpha} Q'\}} \alpha.x_{[Q']} \text{ and } \mathcal{S}_{x_R} := \sum_{\{(\alpha, R') \mid R \xrightarrow{\alpha} R'\}} \alpha.x_{R'}$$

The canonical representative is well-defined since P, Q, as well as processes $[P'] \in [\mathbb{P}^g]$ are finitely branching [11]. Additionally, $\bigcup_{P' \in \mathrm{Reach}(P) \cup \mathrm{Reach}(Q)} \mathrm{Reach}([P'])$ is countable. Moreover, $\mathcal{S}$ is strongly guarded. Furthermore, by construction $\langle x_R | \mathcal{S} \rangle \leftrightarrow R$ for all $R \in \bigcup_{P' \in \mathrm{Reach}(P) \cup \mathrm{Reach}(Q)} \mathrm{Reach}([P'])$.

▶ **Proposition 31.** *Let* $P, Q \in \mathbb{P}^g$ *and* $\mathcal{S}$ *be the canonical representative of* P *and* Q*,* $Ax_r^\infty \vdash P = \langle x_P | \mathcal{S} \rangle$*.*

Proof. It suffices to show that P and $\langle x_P | \mathcal{S} \rangle$ are y_P-components of solutions of $\{y_{P^\dagger} = \sum_{\{(\alpha, P^\ddagger) \mid P^\dagger \xrightarrow{\alpha} P^\ddagger\}} \alpha.y_{P^\ddagger} \mid P^\dagger \in \mathrm{Reach}(P)\}$. Details can be found in [20, Appendix P]. ◀

▶ **Theorem 32.** *Let* $P, Q \in \mathbb{P}^g$*, if* $P \leftrightarrow_{br}^{r} Q$ *then* $Ax_r^\infty \vdash P = Q$*.*

Proof. It suffices to equate $\langle x_P | \mathcal{S} \rangle$ and $\langle x_Q | \mathcal{S} \rangle$ using RDP and the reactive approximation axiom. Details can be found in [20, Appendix P]. ◀

Conclusion

This paper defined a form of branching bisimilarity for processes with time-out transitions, and provided a modal characterisation, congruence results, and a complete axiomatisation for strongly guarded processes. The restriction to strongly guarded processes is rather severe; it rules out processes that may engage in an infinite sequence of time-out transitions, interspersed with τs. Relaxing this restriction is a suitable topic for further work. Another task is to combine this work with the ideas behind *justness* [12], a weaker form of fairness that allows the formulation and derivation of useful liveness properties. In a setting with time-outs, justness would demand that once a parallel component reaches a state in which a time-out transition is enabled, it cannot stay in that state forever after.

As an example of the use of branching reactive bisimulation, one could verify the correctness of a non-trivial system, such as Peterson's mutual exclusion protocol, as modelled in [10]. There it was argued that a similar model without time-out transitions is not possible. The model from [10] features eight visible actions of entering or leaving the critical or non-critical section of process A or B. Abstracting from all actions pertaining to process B yields a protocol that only deals with process A, and a correctness claim could be validated by showing it branching reactive bisimilar with a simple specification of the intended behaviour of A that would apply when B were not around. Although doing such a verification is entirely feasible, for now, it can not be achieved by algebraic means, using our complete axiomatisation. The reason is that abstraction from process B yields infinite sequences of unobservable actions, which are currently not covered by our work.

References

1. Jos C.M. Baeten and W. Peter Weijland. *Process Algebra.* Cambridge Tracts in Theoretical Computer Science 18. Cambridge University Press, 1990. doi:10.1017/CBO9780511624193.
2. Stephen D. Brookes, Tony (C.A.R.) Hoare, and Bill (A.W.) Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, 1984. doi:10.1145/828.833.
3. Olav Bunte, Jan Friso Groote, Jeroen J. A. Keiren, Maurice Laveaux, Thomas Neele, Erik P. de Vink, Wieger Wesselink, Anton Wijs, and Tim A. C. Willemse. The mCRL2 toolset for analysing concurrent systems—improvements in expressivity and usability. In Tomáš Vojnar and Lijun Zhang, editors, Proc. 25th International Conference on *Tools and Algorithms for the Construction and Analysis of Systems,* TACAS'19, held as part of the *European Joint Conferences on Theory and Practice of Software,* ETAPS'19, Prague, Czech Republic, volume 11428 of LNCS, pages 21–39. Springer, 2019. doi:10.1007/978-3-030-17465-1_2.
4. Wan J. Fokkink. *Introduction to Process Algebra.* Texts in Theoretical Computer Science, An EATCS Series. Springer, 2000. doi:10.1007/978-3-662-04293-9.
5. Wan J. Fokkink, Rob J. van Glabbeek, and Bas Luttik. Divide and congruence III: From decomposition of modal formulas to preservation of stability and divergence. *Information and Computation*, 268:104435, 2019. doi:10.1016/j.ic.2019.104435.
6. Hubert Garavel, Frédéric Lang, Radu Mateescu, and Wendelin Serwe. CADP 2010: A toolbox for the construction and analysis of distributed processes. In Parosh Aziz Abdulla and K. Rustan M. Leino, editors, Proceedings *Tools and Algorithms for the Construction and Analysis of Systems,* TACAS '11, volume 6605 of LNCS, pages 372–387. Springer, 2011. doi:10.1007/978-3-642-19835-9_33.
7. Rob J. van Glabbeek. The linear time – branching time spectrum II; the semantics of sequential systems with silent moves (extended abstract). In E. Best, editor, Proceedings *CONCUR'93,* 4^{th} International Conference on *Concurrency Theory,* Hildesheim, Germany, August 1993, volume 715 of LNCS, pages 66–81. Springer, 1993. doi:10.1007/3-540-57208-2_6.
8. Rob J. van Glabbeek. Lean and full congruence formats for recursion. In Proceedings 32^{nd} Annual ACM/IEEE Symposium on *Logic in Computer Science,* LICS'17, Reykjavik, Iceland, June 2017. IEEE Computer Society Press, 2017. doi:10.1109/LICS.2017.8005142.
9. Rob J. van Glabbeek. Failure trace semantics for a process algebra with time-outs. *Logical Methods in Computer Science*, 17(2), 2021. doi:10.23638/LMCS-17(2:11)2021.
10. Rob J. van Glabbeek. Modelling mutual exclusion in a process algebra with time-outs. *Information and Computation*, 294, 2023. doi:10.1016/j.ic.2023.105079.

11 Rob J. van Glabbeek. Reactive bisimulation semantics for a process algebra with timeouts. *Acta Informatica*, 60(1):11–57, 2023. doi:10.1007/s00236-022-00417-1.

12 Rob J. van Glabbeek and Peter Höfner. Progress, justness and fairness. *ACM Computing Surveys*, 52(4), August 2019. doi:10.1145/3329125.

13 Rob J. van Glabbeek and W. Peter Weijland. Branching time and abstraction in bisimulation semantics. *Journal of the ACM*, 43(3):555–600, 1996. doi:10.1145/233551.233556.

14 Clemens Grabmayer and Wan J. Fokkink. A complete proof system for 1-free regular expressions modulo bisimilarity. In H. Hermanns, L. Zhang, N. Kobayashi, and D. Miller, editors, Proc. 35th Annual ACM/IEEE Symposium on *Logic in Computer Science,* LICS'20, pages 465–478. ACM, 2020. doi:10.1145/3373718.3394744.

15 Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985. doi:10.1145/2455.2460.

16 Xinxin Liu and Tingting Yu. Canonical solutions to recursive equations and completeness of equational axiomatisations. In I. Konnov and L. Kovacs, editors, Proceedings 31st International Conference on *Concurrency Theory* (CONCUR 2020), volume 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CONCUR.2020.35.

17 Robin Milner. Operational and algebraic semantics of concurrent processes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, chapter 19, pages 1201–1242. Elsevier Science Publishers B.V. (North-Holland), 1990. Alternatively see *Communication and Concurrency*, Prentice-Hall, Englewood Cliffs, 1989, of which an earlier version appeared as *A Calculus of Communicating Systems*, LNCS 92, Springer, 1980, doi:10.1007/3-540-10235-3.

18 Ernst-Ruediger Olderog and Tony (C.A.R.) Hoare. Specification-oriented semantics for communicating processes. *Acta Informatica*, 23:9–66, 1986. doi:10.1007/BF00268075.

19 Maximilian Pohlmann. Reducing strong reactive bisimilarity to strong bisimilarity. Bachelor's thesis, Technische Universität Berlin, 2021. URL: https://maxpohlmann.github.io/Reducing-Reactive-to-Strong-Bisimilarity/thesis.pdf.

20 Gaspard Reghem and Rob J. van Glabbeek. Branching bisimilarity for processes with time-outs. technical report, full version of the present paper, 2024. arXiv:2408.10117.

21 Gaspard Reghem and Rob J. van Glabbeek. Concrete branching bisimilarity for processes with time-outs, 2024. URL: https://theory.stanford.edu/~rvg/abstracts.html#167.

A Examples

Scope of the First Clause of Definition 1

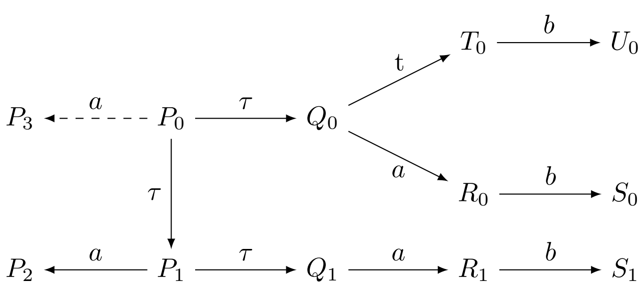

Figure 2 Counter-Example to a Naive Clause 1.a.

In Figure 2, the process $a.0 + \tau.(\text{t}.b.0 + a.b.0) + \tau.(\tau.a.b.0 + a.0)$ is represented as an LTS. Let $A := \{a, b\}$. Removing the dashed a-transition generates the process $\tau.(\text{t}.b.0 + a.b.0) + \tau.(\tau.a.b.0 + a.0)$.

First, we are going to show that these two processes are not branching reactive bisimilar. Let's try to build a branching reactive bisimulation between them. The only way to match the dashed a-transition of $a.0+\tau.(\mathrm{t}.0+a.b.0)+\tau.(\tau.a.b.0+a.0)$ is by the a-transition between P_1 and P_2, because all other a-transitions are followed by a b-transition. This requires to elide the τ-transition between P_0 and P_1, who must be branching reactive bisimilar. Since $P_0 \leftrightarrow_{br} P_1$, when considering the τ-transition between P_0 and Q_0, Q_0 has to be branching reactive bisimilar to P_1 or Q_1. Now, the a-transition between Q_0 and R_0 has to be matched by the a-transition between Q_1 and R_1 because of the following b-transition. This implies $Q_0 \leftrightarrow_{br} Q_1$, thus, $Q_0 \leftrightarrow_{br}^{\emptyset} Q_1$. One has $\mathcal{I}(Q_0) \cap (\emptyset \cup \{\tau\}) = \emptyset$ and $Q_0 \xrightarrow{\mathrm{t}} T_0$, i.e., when the environment temporary allows no visible actions, Q_0 can time-out into a state in which b is possible. This behaviour cannot be matched by Q_1 – a contradiction.

Now, consider the alternative to Definition 1 in which the first clause has been changed to

1. a. if $P \xrightarrow{\tau} P'$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{(\tau)} Q_2$ with $\mathscr{R}(P, Q_1)$ and $\mathscr{R}(P', Q_2)$.

In other words, the scope of the first clause is restricted to τ-transitions. This modification enables building a bisimulation between the two processes. Indeed, the dashed a-transition is only considered when the environment allows a. Thus, it is sufficient to get $P_0 \leftrightarrow_{br}^{A} P_1$ and $P_0 \leftrightarrow_{br}^{\{a\}} P_1$ and not $P_0 \leftrightarrow_{br} P_1$ anymore. Therefore, it is sufficient to match Q_0 and Q_1 in environments allowing a. As a result, the outgoing time-out transition of Q_0 is never considered when matching Q_0 with Q_1, solving our previous issue. Once this observation is made, building the bisimulation is trivial.

Finally, place both processes in the context $\_\_ \|_{\{a\}} (\tau.0 + a.0)$. It behaves like a one-way switch enabling to block all a-transitions forever as soon as the τ-transition is performed. Let's try to build a branching reactive bisimulation between the two processes. Following the same reasoning as before, it is necessary to get $P_0 \|_{\{a\}} (\tau.0 + a.0) \leftrightarrow_{br}^{A} P_1 \|_{\{a\}} (\tau.0 + a.0)$ because of the dashed a-transition, and then $Q_0 \|_{\{a\}} (\tau.0 + a.0) \leftrightarrow_{br}^{A} Q_1 \|_{\{a\}} (\tau.0 + a.0)$ because of the a-transition between Q_0 and R_0. Note that $Q_0 \|_{\{a\}} (\tau.0 + a.0) \xrightarrow{\tau} Q_0 \|_{\{a\}} 0$ $\xrightarrow{\mathrm{t}} T_0 \|_{\{a\}} 0 \xrightarrow{b} U_0 \|_{\{a\}} 0$ and $\mathcal{I}(Q_0 \|_{\{a\}} 0) \cap (A \cup \{\tau\}) = \emptyset$. As before, $Q_0 \|_{\{a\}} (\tau.0 + a.0)$ can time-out into a state in which b is executable, whereas this behaviour is impossible in $Q_1 \|_{\{a\}} (\tau.0 + a.0)$. As a result, restricting the scope of the first clause of Definition 1 to τ-transitions prevents $\leftrightarrow_{br}$ from being a congruence for parallel composition.

Necessity of the Stability Respecting Clause

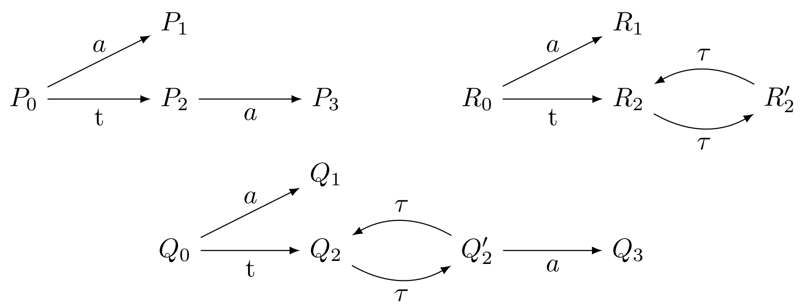

Figure 3 Counter-Example to the Absence of a Stability Respecting Clause.

In Figure 3, three processes are represented as LTSs. Take $A := \{a\}$. According to Definition 1, $\neg(P_0 \leftrightarrow_{br} Q_0)$ and $Q_0 \leftrightarrow_{br} R_0$.

Let's try to build a branching reactive bisimulation between the top-left and bottom processes. Matching the time-out between Q_0 and Q_2 implies that $Q_2 \leftrightarrow^{\emptyset}_{br} P_0$ or $Q_2 \leftrightarrow^{\emptyset}_{br} P_2$. However, $P_0 \stackrel{\tau}{\not\to}$ and $P_2 \stackrel{\tau}{\not\to}$, thus, there should be a path $Q_2 \Longrightarrow Q'_2 \stackrel{\tau}{\not\to}$, but this is not the case.

The symmetric closure of

$$\mathscr{R} := \{(Q_0, R_0), (Q_1, R_1), (Q_2, \emptyset, R_2), (Q'_2, \emptyset, R'_2)\} \cup \{(Q_0, X, R_0), (Q_1, X, R_1) \mid X \subseteq A\}$$

is a branching reactive bisimulation. The a-transition between Q'_2 and Q_3 does not have to be matched since Q'_2 is considered only when the environment disallows a.

Now, suppose that the stability respecting condition is removed from Definition 1. As a result, a branching reactive bisimulation can be built between the top-left and bottom processes. The symmetric closure of

$$\begin{aligned}\mathscr{R}' :=& \{(P_0, Q_0), (P_1, Q_1), (P_2, Q_2), (P_2, Q'_2), (P_3, Q_3)\}\\ &\cup \{(P_0, X, Q_0), (P_1, X, Q_1), (P_2, X, Q_2), (P_2, X, Q'_2), (P_3, X, Q_3) \mid X \subseteq A\}\end{aligned}$$

would be a branching reactive bisimulation. Moreover, $\mathscr{R}$ would still be a branching reactive bisimulation, since Definition 1 has merely been weakened. Therefore, according to the modified Definition 1, $P_0 \leftrightarrow_{br} Q_0$ and $Q_0 \leftrightarrow_{br} R_0$. However, when trying to construct a branching reactive bisimulation between P_0 and R_0, because of the time-out transition, R_2 has to be matched to P_0 or P_2 and no a-transition is reachable from R_2; therefore, $\neg(P_0 \leftrightarrow_{br} R_0)$. As a result, removing the stability respecting clause from Definition 1 prevents $\leftrightarrow_{br}$ from being an equivalence relation.

B Concrete Time-out Version

Before studying $\leftrightarrow_{br}$, we looked at another version which is not eliding any time-out transitions. More formally, it is defined by replacing Clause 2.d of Definition 1 by

2. d. if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there exists a path $Q \Longrightarrow Q_1 \xrightarrow{\mathrm{t}} Q_2$ with $\mathscr{R}(P', X, Q_2)$.

It is not necessary to require to match with an executable time-out (i.e. $\mathcal{I}(Q_1) \cap (X \cup \{\tau\}) = \emptyset$) since this is implied by the other clauses. It is also implied that $\mathscr{R}(P, X, Q_1)$ in the above clause. This bisimilarity has properties similar to $\leftrightarrow_{br}$, to be recapped below. No proof will be provided here since they rely on the same strategies and are actually simpler because of the absence of time-out omission. However, a technical report [21] is available. In the remainder of this appendix, $\leftrightarrow^{c}_{br}$ stands for the concrete time-out version.

The stuttering lemma (Lemma 3) still holds and $\leftrightarrow^{c}_{br}$ and $(\leftrightarrow^{cX}_{br})_{X \subseteq A}$ are still equivalence relations (Proposition 4). The rooted version of $\leftrightarrow^{c}_{br}$ is exactly Definition 5 and $\leftrightarrow^{r}_{br}$ and $(\leftrightarrow^{rX}_{br})_{X \subseteq A}$ are still equivalence relations (Proposition 6). The Pohlmann encoding (Table 4) is simplified as the rooted variants are no longer needed: $P \leftrightarrow^{cr}_{br} Q \Leftrightarrow \vartheta(P) \leftrightarrow^{r}_{b} \vartheta(Q)$. If $\leftrightarrow_{b}$ stands for the classical stability respecting branching bisimulation [13, 7], $P \leftrightarrow^{c}_{br} Q \Leftrightarrow \vartheta(P) \leftrightarrow_{b} \vartheta(Q)$; $P \leftrightarrow^{cX}_{br} Q \Leftrightarrow \vartheta_X(P) \leftrightarrow_{b} \vartheta_X(Q)$; $P \leftrightarrow^{cr}_{br} Q \Leftrightarrow \vartheta(P) \leftrightarrow^{r}_{b} \vartheta(Q)$ and $P \leftrightarrow^{crX}_{br} Q \Leftrightarrow \vartheta_X(P) \leftrightarrow^{r}_{b} \vartheta_X(Q)$.

The generalised definition of $\leftrightarrow^{c}_{br}$ can be obtained by replacing Clause 1.b. and 2.c. in Definition 33 by

1. b. If $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there exists a path $Q \Longrightarrow Q_1 \xrightarrow{\mathrm{t}} Q_2$ with $\mathscr{R}(P', X, Q_2)$

2. c. If $\mathcal{I}(P) \cap (X \cup Y \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there exists a path $Q \Longrightarrow Q_1 \xrightarrow{\mathrm{t}} Q_2$ with $\mathscr{R}(P', Y, Q_2)$

The rooted generalised version is exactly Definition 34 and they induce the same bisimilarities as the previous definitions (Proposition 35). In the modal characterisation, $X\varphi$ is not useful anymore, nor $\varphi\langle\varepsilon_X\rangle\varphi'$. Replacing the fifth induction rule of $\mathbb{L}_b$ by $\langle\varepsilon\rangle\langle \mathrm{t}_X\rangle\varphi$ yields the counterpart of Theorem 11.

The corresponding time-out bisimulation can be obtained by replacing Clause 2. of Definition 13 by

2. if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there exists a path $P \Longrightarrow P_1 \xrightarrow{\mathrm{t}} P_2$ with $\theta_X(P') \mathscr{B} \theta_X(Q_2)$.

The rooted time-out bisimulation is exactly Definition 14 and they agree with the previous definitions (Proposition 15). $\leftrightarrow^{c}_{br}$ is a congruence for prefixing, parallel composition, abstraction, renaming and the operator θ^U_L (Proposition 17). $\leftrightarrow^{cr}_{br}$ is a full congruence (Theorem 18).

As $\leftrightarrow^{cr}_{br} \subseteq \leftrightarrow$, RDP holds for $\leftrightarrow^{cr}_{br}$. The definition of well-guarded recursion can be weakened by allowing t as a guard and RSP holds for $\leftrightarrow^{cr}_{br}$ on processes that are guarded in this sense. Lemma 22 is not useful anymore since time-out omissions are not considered. The set of all axioms of Table 3 except the t-branching and τ/t-branching ones is a complete axiomatisation of $\leftrightarrow^{c}_{br}$ (Theorem 32). Moreover, to obtain the complete axiomatisation of $\leftrightarrow^{c}_{br}$ on recursion-free processes, it suffices to remove RDP and RSP.

C Generalised branching reactive bisimulation

The second clause of Definition 1 is quite tedious to check; thus, an equivalent definition of the bisimilarity would be useful. Actually, it is possible to define the exact same notion in a more general way at the cost of some clear motivations.

▶ **Definition 33.** A *generalised branching reactive bisimulation* is a symmetric relation $\mathscr{R} \subseteq (\mathbb{P} \times \mathbb{P}) \cup (\mathbb{P} \times \mathcal{P}(A) \times \mathbb{P})$ such that, for all $P, Q \in \mathbb{P}$ and $X \subseteq A$,

1. if $\mathscr{R}(P, Q)$
   - **a.** if $P \xrightarrow{\alpha} P'$ with $\alpha \in A_\tau$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{(\alpha)} Q_2$ with $\mathscr{R}(P, Q_1)$ and $\mathscr{R}(P', Q_2)$,
   - **b.** if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there is a path $Q = Q_0 \Longrightarrow Q_1 \xrightarrow{\mathrm{t}} Q_2 \Longrightarrow Q_3 \xrightarrow{\mathrm{t}} \dots \Longrightarrow Q_{2r-1} \xrightarrow{(\mathrm{t})} Q_{2r}$ with $r > 0$, such that $Q_1 \stackrel{\tau}{\nrightarrow}$, $\forall i \in [1, r-1]$, $\mathscr{R}(P, X, Q_{2i}) \wedge \mathcal{I}(Q_{2i+1}) \cap (X \cup \{\tau\}) = \emptyset$ and $\mathscr{R}(P', X, Q_{2r})$,
   - **c.** if $P \stackrel{\tau}{\nrightarrow}$ then there exists a path $Q \Longrightarrow Q_0 \stackrel{\tau}{\nrightarrow}$;

2. if $\mathscr{R}(P, X, Q)$
   - **a.** if $P \xrightarrow{\tau} P'$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{(\tau)} Q_2$ with $\mathscr{R}(P, X, Q_1)$ and $\mathscr{R}(P', X, Q_2)$,
   - **b.** if $P \xrightarrow{a} P'$ with $a \in X \vee \mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ then there is a path $Q \Longrightarrow Q_1 \xrightarrow{a} Q_2$ with $\mathscr{R}(P, X, Q_1)$ and $\mathscr{R}(P', Q_2)$,
   - **c.** if $\mathcal{I}(P) \cap ((X \cup Y) \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there is a path $Q = Q_0 \Longrightarrow Q_1 \xrightarrow{\mathrm{t}} Q_2 \Longrightarrow Q_3 \xrightarrow{\mathrm{t}} \dots \Longrightarrow Q_{2r-1} \xrightarrow{(\mathrm{t})} Q_{2r}$ with $r > 0$, such that $Q_1 \stackrel{\tau}{\nrightarrow}$, $\forall i \in [1, r-1]$, $\mathscr{R}(P, Y, Q_{2i}) \wedge \mathcal{I}(Q_{2i+1}) \cap (Y \cup \{\tau\}) = \emptyset$ and $\mathscr{R}(P', Y, Q_{2r})$,
   - **d.** if $P \stackrel{\tau}{\nrightarrow}$ then there is a path $Q \Longrightarrow Q_0 \stackrel{\tau}{\nrightarrow}$.

The strong point of the generalised definitions is the restriction on the use of triplets, making use of them only after performing a time-out. A generalised version of rooted branching reactive bisimulation can be defined in a similar fashion.

▶ **Definition 34.** A *generalised rooted branching reactive bisimulation* is a symmetric relation $\mathscr{R} \subseteq (\mathbb{P} \times \mathbb{P}) \cup (\mathbb{P} \times \mathcal{P}(A) \times \mathbb{P})$ such that, for all $P, Q \in \mathbb{P}$ and $X \subseteq A$,

1. if $\mathscr{R}(P, Q)$
   a. if $P \xrightarrow{\alpha} P'$ with $\alpha \in A_\tau$ then there is a transition $Q \xrightarrow{\alpha} Q'$ such that $P' \leftrightarroweq_{br} Q'$,
   b. if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there is a transition $Q \xrightarrow{\mathrm{t}} Q'$ with $P' \leftrightarroweq_{br}^X Q'$,
2. if $\mathscr{R}(P, X, Q)$
   a. if $P \xrightarrow{\tau} P'$ then there is a transition $Q \xrightarrow{\tau} Q'$ such that $P' \leftrightarroweq_{br}^X Q'$,
   b. if $P \xrightarrow{a} P'$ with $a \in X \vee \mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ then there is a transition $Q \xrightarrow{a} Q'$ such that $P' \leftrightarroweq_{br} Q'$,
   c. if $\mathcal{I}(P) \cap ((X \cup Y) \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$ then there is a transition $Q \xrightarrow{\mathrm{t}} Q'$ such that $P' \leftrightarroweq_{br}^Y Q'$.

Note that if a system has no time-out, then a generalised [rooted] branching reactive bisimulation is a stability respecting [rooted] branching bisimulation, thus proving that [rooted] branching reactive bisimilarity is indeed an extension of stability respecting [rooted] branching bisimilarity to reactive systems with time-outs.

▶ **Proposition 35.** *Let $P, Q \in \mathbb{P}$ and $X \subseteq A$,*

- *$P \leftrightarroweq_{br} Q$ (resp. $P \leftrightarroweq_{br}^X Q$) iff there exists a generalised branching reactive bisimulation $\mathscr{R}$ with $\mathscr{R}(P, Q)$ (resp. $\mathscr{R}(P, X, Q)$),*
- *$P \leftrightarroweq_{br}^r Q$ (resp. $P \leftrightarroweq_{br}^{rX} Q$) iff there exists a rooted generalised branching reactive bisimulation $\mathscr{R}$ with $\mathscr{R}(P, Q)$ (resp. $\mathscr{R}(P, X, Q)$).*

Proof. It suffices to verify that any [rooted] branching reactive bisimulation is a [rooted] generalised branching reactive bisimulation and that, for any [rooted] generalised branching reactive bisimulation $\mathscr{R}$, $\mathscr{R}' := \mathscr{R} \cup \{(P, X, Q) \mid \mathscr{R}(P, Q) \wedge X \subseteq A\} \cup \{(P, Y, Q), (P, Q) \mid \exists X \subseteq A, \mathscr{R}(P, X, Q) \wedge (\mathcal{I}(P) \cup \mathcal{I}(Q)) \cap (X \cup \{\tau\}) = \emptyset \wedge Y \subseteq A\}$ is a [rooted] branching reactive bisimulation. Details can be found in [20, Appendix C]. ◀

D Pohlmann Encoding

Reactive bisimulations are sometimes complicated to check because of the large number of potential sets of allowed actions. In [19], Pohlmann introduces an encoding which reduces strong reactive bisimilarity to strong bisimilarity. To this end he introduces unary operators ϑ and ϑ_X for $X \subseteq A$ that model placing their argument process in an environment that is triggered to change, or allows exactly the actions in X, respectively. The actions $\mathrm{t}_\varepsilon \notin A$ and $\varepsilon_X \notin A$ for $X \subseteq A$ are generated by the new operators, but may not be used by processes substituted for their arguments P. They model a time-out action taken by the environment, and the stabilisation of an environment into one that allows exactly the set of actions X, respectively. After a slight modification of the encoding, a similar result can be obtained for branching reactive bisimilarity. We also introduce variants ϑ^r and ϑ^r_X of these operators that are targeting rooted branching reactive bisimilarity.

In [19], the first rule only applies to τ-transitions; this echoes the previous remark about applying the first clause of Definition 1 only to invisible actions. As the intermediary actions t_ε and $(\varepsilon_X)_{X \subseteq A}$ interfere with rootedness, the actions $(\mathrm{t}_X)_{X \subseteq A}$ are added when rootedness has to be preserved. One can think of these as doing the actions ε_X and t in one (instead of two) steps. Note that the encoding rules mirror the clauses of Definition 1. The encoding transforms $\leftrightarroweq_{br}$ into $\leftrightarroweq_{tb}$ (see Definition 7), and $\leftrightarroweq_{br}^r$ in $\leftrightarroweq_{tb}^r$ (Definition 8).

Table 4 Operational semantics of ϑ, ϑ^r, $(\vartheta_X)_{X\subseteq A}$ and $(\vartheta^r_X)_{X\subseteq A}$.

$$\begin{array}{lclcl}
\vartheta(P) \xrightarrow{\alpha} \vartheta(P') & \wedge & \vartheta^r(P) \xrightarrow{\alpha} \vartheta(P') & \Leftrightarrow & P \xrightarrow{\alpha} P' \wedge \alpha \in A_\tau \\
 & & \vartheta^r(P) \xrightarrow{\mathrm{t}_X} \vartheta_X(P') & \Leftrightarrow & \mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset \wedge P \xrightarrow{\mathrm{t}} P' \\
\vartheta(P) \xrightarrow{\varepsilon_X} \vartheta_X(P) & \wedge & \vartheta^r(P) \xrightarrow{\varepsilon_X} \vartheta_X(P) & & \\
\vartheta_X(P) \xrightarrow{\tau} \vartheta_X(P') & \wedge & \vartheta^r_X(P) \xrightarrow{\tau} \vartheta_X(P') & \Leftrightarrow & P \xrightarrow{\tau} P' \\
\vartheta_X(P) \xrightarrow{a} \vartheta(P') & \wedge & \vartheta^r_X(P) \xrightarrow{a} \vartheta(P') & \Leftrightarrow & P \xrightarrow{a} P' \wedge \alpha \in X \\
\vartheta_X(P) \xrightarrow{\mathrm{t}_\varepsilon} \vartheta(P) & \wedge & \vartheta^r_X(P) \xrightarrow{\mathrm{t}_\varepsilon} \vartheta^r(P) & \Leftrightarrow & \mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset \\
\vartheta_X(P) \xrightarrow{\mathrm{t}} \vartheta_X(P') & \wedge & \vartheta^r_X(P) \xrightarrow{\mathrm{t}} \vartheta_X(P') & \Leftrightarrow & \mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset \wedge P \xrightarrow{\mathrm{t}} P'
\end{array}$$

▶ **Proposition 36.** *Let $P, Q \in \mathbb{P}$.*

- $P \leftrightarroweq_{br} Q \Leftrightarrow \vartheta(P) \leftrightarroweq_{tb} \vartheta(Q)$
- $P \leftrightarroweq^r_{br} Q \Leftrightarrow \vartheta^r(P) \leftrightarroweq^r_{tb} \vartheta^r(Q)$
- $P \leftrightarroweq^X_{br} Q \Leftrightarrow \vartheta_X(P) \leftrightarroweq_{tb} \vartheta_X(Q)$
- $P \leftrightarroweq^{rX}_{br} Q \Leftrightarrow \vartheta^r_X(P) \leftrightarroweq^r_{tb} \vartheta^r_X(Q)$

Proof. It suffices to prove that: if R is a branching reactive bisimulation then $R' := \{(\vartheta(P), \vartheta(Q)) \mid R(P,Q)\} \cup \{(\vartheta_X(P), \vartheta_X(Q)) \mid R(P,X,Q)\}$ is a t-branching bisimulation; and if R is a t-branching bisimulation then $R' := \{(P,Q),(P,X,Q) | R(\vartheta(P),\vartheta(Q)) \wedge X \subseteq A\} \cup \{(P,X,Q) | R(\vartheta_X(P),\vartheta_X(Q))\}$ is a branching reactive bisimulation. The rooted case is very similar. Details can be found in [20, Appendix D]. ◀

It would have been possible to define the t-branching bisimilarity differently while preserving the same result. The encoded processes are part of a sub-class with specific properties. For instance, an encoded process cannot have an outgoing τ-transition and an outgoing time-out by definition of ϑ and $(\vartheta_X)_{X\subseteq A}$, i.e., for any encoded process P, $P \xrightarrow{\mathrm{t}} \Rightarrow P \stackrel{\tau}{\nrightarrow}$. Thus, adding the condition $\forall i \in [0, r-1], Q_{2i+1} \stackrel{\tau}{\nrightarrow}$ in clause 2 of Definition 7 does not interfere with our result even though it obviously defines a different bisimilarity. We settled on Definition 7 because it is the one that yields the simplest proofs.