# Reactive Bisimulation Semantics for a Process Algebra with Time-Outs

Rob van Glabbeek
Data61, CSIRO, Sydney, Australia
School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
rvg@cs.stanford.edu

This paper introduces the counterpart of strong bisimilarity for labelled transition systems extended with time-out transitions. It supports this concept through a modal characterisation, congruence results for a standard process algebra with recursion, and a complete axiomatisation.

## 1 Introduction

This is a contribution to classic untimed non-probabilistic process algebra, modelling systems that move from state to state by performing discrete, uninterpreted actions. A system is modelled as a process-algebraic expression, whose standard semantics is a state in a labelled transition system (LTS). An LTS consists of a set of states, with action-labelled transitions between them. The execution of an action is assumed to be instantaneous, so when any time elapses the system must be in one of its states. With "untimed" I mean that I will refrain from quantifying the passage of time; however, whether a system can pause in some state or not will be part of my model.

Following [33], I consider *reactive* systems that interact with their environments through the synchronous execution of visible actions $a$, $b$, $c$, ... taken from an alphabet $A$. At any time, the environment *allows* a set of actions $X \subseteq A$, while *blocking* all other actions. At discrete moments the environment can change the set of actions it allows. In a metaphor from [33], the environment of a system can be seen as a user interacting with it. This user has a button for each action $a \in A$, on which it can exercise pressure. When the user exercises pressure *and* the system is in a state where it can perform action $a$, the action occurs. For the system this involves taking an $a$-labelled transition to a following state; for the environment it entails the button going down, thus making the action occurrence observable. This can trigger the user to alter the set of buttons on which it exercises pressure.

The current paper considers two special actions that can occur as transition labels: the traditional *hidden action* $\tau$ [33], modelling the occurrence of an instantaneous action from which we abstract, and the *time-out* action $\mathrm{t}$, modelling the end of a time-consuming activity from which we abstract. The latter was introduced in [18] and constitutes the main novelty of the present paper with respect to [33] and forty years of research in process algebra. Both special actions are assumed to be unobservable, in the sense that their occurrence cannot trigger any state-change in the environment. Conversely, the environment cannot cause or block the occurrence of these actions.

Following [18], I model the passage of time in the following way. When a system arrives in a state $P$, and at that time $X$ is the set of actions allowed by the environment, there are two possibilities. If $P$ has an outgoing transition $P \stackrel{\alpha}{\longrightarrow} Q$ with $\alpha \in X \cup \{\tau\}$, the system immediately takes one of the outgoing transitions $P \stackrel{\alpha}{\longrightarrow} Q$ with $\alpha \in X \cup \{\tau\}$, without spending any time in state $P$. The choice between these actions is entirely nondeterministic. The system cannot immediately take a transition $\stackrel{b}{\longrightarrow}$

An extended abstract of this paper appears in the proceedings of CONCUR 2020. doi:10.4230/LIPIcs.CONCUR.2020.6

with $b \in A \backslash X$, because the action $b$ is blocked by the environment. Neither can it immediately take a transition $P \xrightarrow{\mathrm{t}} Q$, because such transitions model the end of an activity with a finite but positive duration that started when reaching state $P$.

In case $P$ has no outgoing transition $P \xrightarrow{\alpha} Q$ with $\alpha \in X \cup \{\tau\}$, the system idles in state $P$ for a positive amount of time. This idling can end in two possible ways. Either one of the time-out transitions $P \xrightarrow{\mathrm{t}} Q$ occurs, or the environment spontaneously changes the set of actions it allows into a different set $Y$ with the property that $P \xrightarrow{a} Q$ for some $a \in Y$. In the latter case a transition $P \xrightarrow{a} Q$ occurs, with $a \in Y$. The choice between the various ways to end a period of idling is entirely nondeterministic. It is possible to stay forever in state $P$ only if there are no outgoing time-out transitions $P \xrightarrow{\mathrm{t}} Q$.

The addition of time-outs enhances the expressive power of LTSs and process algebras. The process $a.P + \mathrm{t}.b.Q$, for instance, models a choice between $a.P$ and $b.Q$ where the former has priority. In an environment where $a$ is allowed it will always choose $a.P$ and never $b.Q$; but in an environment that blocks $a$ the process will, after some delay, proceed with $b.Q$. Such a priority mechanism cannot be modelled in standard process algebras without time-outs, such as CCS [33], CSP [6, 28] and ACP [2, 10]. Additionally, mutual exclusion cannot be correctly modelled in any of these standard process algebras [20], but adding time-outs makes it possible—see Section 11 for a more precise statement.

In [18] I characterised the coarsest reasonable semantic equivalence on LTSs with time-outs—the one induced by *may testing*, as proposed by De Nicola & Hennessy [8]. In the absence of time-outs, may testing yields *weak trace equivalence*, where two processes are defined equivalent iff they have the same *weak traces*: sequence of actions the system can perform, while eliding hidden actions. In the presence of time-outs weak trace equivalence fails to be a congruence for common process algebraic operators, and may testing yields its congruence closure, characterised in [18] as *(rooted) failure trace equivalence*.

The present paper aims to characterise one of the finest reasonable semantic equivalences on LTSs with time-outs—the counterpart of strong bisimilarity for LTSs without time-outs. Naturally, strong bisimilarity can be applied verbatim to LTSs with time-outs—and has been in [18]—by treating $\mathrm{t}$ exactly like any visible action. Here, however, I aim to take into account the essence of time-outs, and propose an equivalence that satisfies some natural laws discussed in [18], such as $\tau.P + \mathrm{t}.Q = \tau.P$ and $a.P + \mathrm{t}.(Q + \tau.R + a.S) = a.P + \mathrm{t}.(Q + \tau.R)$. To motivate the last law, note that the time-out transition $a.P + \mathrm{t}.(Q + \tau.R + a.S) \xrightarrow{\mathrm{t}} Q + \tau.R + a.S$ can occur only in an environment that blocks the action $a$, for otherwise $a$ would have taken place before the time-out went off. The occurrence of this transition is not observable by the environment, so right afterwards the state of the environment is unchanged, and the action $a$ is still blocked. Therefore, the process $Q + \tau.R + a.S$ will, without further ado, proceed with the $\tau$-transition to $R$, or any action from $Q$, just as if the $a.S$ summand were not present.

Standard process algebras and LTSs without time-outs can model systems whose behaviour is triggered by input signals from the environment in which they operate. This is why they are called "reactive systems". By means of time-outs one can additionally model systems whose behaviour is triggered by the *absence* of input signals from the environment, during a sufficiently long period. This creates a greater symmetry between a system and its environment, as it has always been understood that the environment or user of a system can change its behaviour as a result of sustained inactivity of the system it is interacting with. Hence one could say that process algebras and LTSs enriched with time-outs form a more faithful model of reactivity. It is for this reason that I use the name *reactive bisimilarity* for the appropriate form of bisimilarity on systems modelled in this fashion.

Section 2 introduces strong reactive bisimilarity as the proper counterpart of strong bisimilarity in the presence of time-out transitions. Naturally, it coincides with strong bisimilarity when there are no time-out transitions. Section 3 derives a modal characterisation; a reactive variant of the Hennessy-Milner logic. Section 4 offers an alternative characterisation of strong reactive bisimilarity that will be more convenient in proofs, although it is lacks the intuitive appeal to be used as the initial definition. Appendix C, reporting on work by Max Pohlmann [37], offers yet another characterisation of strong reactive bisimilarity; one that reduces it to strong bisimilarity in a context that models a system together with its environment.

Section 5 recalls the process algebra CCSP, a common mix of CCS and CSP, and adds the time-out action, as well as two auxiliary operators that will be used in the forthcoming axiomatisation. Section 6 states that in this process algebra one can express all countably branching transition systems, and only those, or all and only the finitely branching ones when restricting to guarded recursion.

Section 7 recalls the concept of a congruence, focusing on the congruence property for the recursion operator, which is commonly the hardest to establish. It then shows that the simple *initials equivalence*, as well as Milner's strong bisimilarity, are congruences. Due to the presence of negative premises in the operational rules for the auxiliary operators, these proofs are not entirely trivial. Using these results as a stepping stone, Section 8 shows that strong reactive bisimilarity is a congruence for my extension of CCSP. Here the congruence property for one of the auxiliary operators with negative premises is needed in establishing the result for the common CCSP operators, such as parallel composition.

Section 9 shows that guarded recursive specifications have unique solutions up to strong reactive bisimilarity. Using this, Section 10 provides a sound and complete axiomatisation for processes with guarded recursion. My completeness proof combines three innovations in establishing completeness of process algebraic axiomatisations. First of all, following [22], it applies to *all* processes in a Turing powerful language like guarded CCSP, rather than the more common fragment merely employing finite sets of recursion equations featuring only choice and action prefixing. Secondly, instead of the classic technique of *merging guarded recursive equations* [31, 32, 40, 11, 30], which in essence proves two bisimilar systems $P$ and $Q$ equivalent by equating both to an intermediate variant that is essentially a *product* of $P$ and $Q$, I employ the novel method of *canonical solutions* [24, 29], which equates both $P$ and $Q$ to a canonical representative within the bisimulation equivalence class of $P$ and $Q$—one that has only one reachable state for each bisimulation equivalence class of states of $P$ and $Q$. In fact I tried so hard, and in vain, to apply the traditional technique of merging guarded recursive equations, that I came to believe that it fundamentally does not work for this axiomatisation. The third innovation is the use of the axiom of choice [41] in defining the transition relation on my canonical representative, in order to keep this process finitely branching.

Section 11 describes a worthwhile gain in expressiveness caused by the addition of time-outs, and presents an agenda for future work.

## 2 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 I consider LTSs with $Act := A \uplus \{\tau, \mathrm{t}\}$, where $A$ is a set of *visible actions*, $\tau$ is the *hidden action*, and $\mathrm{t}$ the *time-out* action. The set of *initial* actions of a process $P \in \mathbb{P}$ is $\mathcal{I}(P) := \{\alpha \in A \cup \{\tau\} \mid P \xrightarrow{\alpha}\}$. Here $P \xrightarrow{\alpha}$ means that there is a $Q$ with $P \xrightarrow{\alpha} Q$.

**Definition 1** A *strong reactive bisimulation* is a symmetric relation $\mathscr{R} \subseteq (\mathbb{P} \times \mathscr{P}(A) \times \mathbb{P}) \cup (\mathbb{P} \times \mathbb{P})$ (meaning that $(P,X,Q) \in \mathscr{R} \Leftrightarrow (Q,X,P) \in \mathscr{R}$ and $(P,Q) \in \mathscr{R} \Leftrightarrow (Q,P) \in \mathscr{R}$), such that,

- if $(P,Q) \in \mathscr{R}$ and $P \xrightarrow{\tau} P'$, then there exists a $Q'$ such that $Q \xrightarrow{\tau} Q'$ and $(P',Q') \in \mathscr{R}$,
- if $(P,Q) \in \mathscr{R}$ then $(P,X,Q) \in \mathscr{R}$ for all $X \subseteq A$,

and for all $(P,X,Q) \in \mathscr{R}$,

- if $P \xrightarrow{a} P'$ with $a \in X$, then there exists a $Q'$ such that $Q \xrightarrow{a} Q'$ and $(P', Q') \in \mathscr{R}$,
- if $P \xrightarrow{\tau} P'$, then there exists a $Q'$ such that $Q \xrightarrow{\tau} Q'$ and $(P', X, Q') \in \mathscr{R}$,
- if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$, then $(P, Q) \in \mathscr{R}$, and
- if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$, then $\exists Q'$ such that $Q \xrightarrow{\mathrm{t}} Q'$ and $(P', X, Q') \in \mathscr{R}$.

Processes $P, Q \in \mathbb{P}$ are *strongly $X$-bisimilar*, denoted $P \leftrightarroweq_r^X Q$, if $(P, X, Q) \in \mathscr{R}$ for some strong reactive bisimulation $\mathscr{R}$. They are *strongly reactive bisimilar*, denoted $P \leftrightarroweq_r Q$, if $(P, Q) \in \mathscr{R}$ for some strong reactive bisimulation $\mathscr{R}$.

Intuitively, $(P, X, Q) \in \mathscr{R}$ says that processes $P$ and $Q$ behave the same way, as witnessed by the relation $\mathscr{R}$, when placed in the environment $X$—meaning any environment that allows exactly the actions in $X$ to occur—whereas $(P, Q) \in \mathscr{R}$ says they behave the same way in an environment that has just been triggered to change. An environment can be thought of as an unknown process placed in parallel with $P$ and $Q$, using the operator $\|_A$, enforcing synchronisation on all visible actions. The environment $X$ can be seen as a process $\sum_{i \in I} a_i.R_i + \mathrm{t}.R$ where $\{a_i \mid i \in I\} = X$. A triggered environment, on the other hand, can execute a sequence of instantaneous hidden actions before stabilising as an environment $Y$, for $Y \subseteq A$. During this execution, actions can be blocked and allowed in rapid succession. Since the environment is unknown, the bisimulation should be robust under any such environment.

The first clause for $(P, X, Q) \in \mathscr{R}$ is like the common transfer property of strong bisimilarity [33]: a visible $a$-transition of $P$ can be matched by one of $Q$, such that the resulting processes $P'$ and $Q'$ are related again. However, I require it only for actions $a \in X$, because actions $b \in A \backslash X$ cannot happen at all in the environment $X$, and thus need not be matched by $Q$. Since the occurrence of $a$ is observable by the environment, this can trigger the environment to change the set of actions it allows, so $P'$ and $Q'$ ought to be related in a triggered environment.

The second clause is the transfer property for $\tau$-transitions. Since these are not observable by the environment, they cannot trigger a change in the set of actions allowed by it, so the resulting processes $P'$ and $Q'$ should be related only in the same environment $X$.

The first clause for $(P, Q) \in \mathscr{R}$ expresses the transfer property for $\tau$-transitions in a triggered environment. Here it may happen that the $\tau$-transition occurs before the environment stabilises, and hence $P'$ and $Q'$ will still be related in a triggered environment. A similar transfer property for $a$-transitions is already implied by the next two clauses.

The second clause allows a triggered environment to stabilise into any environment $X$.

The first two clauses for $(P, X, Q) \in \mathscr{R}$ imply that if $(P, X, Q) \in \mathscr{R}$ then $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \mathcal{I}(Q) \cap (X \cup \{\tau\})$. So $P \leftrightarroweq_r Q$ implies $\mathcal{I}(P) = \mathcal{I}(Q)$. The condition $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ is necessary and sufficient for the system to remain a positive amount of time in state $P$ when $X$ is the set of allowed actions. The next clause says that during this time the environment may be triggered to change the set of actions it allows by an event outside our model, that is, by a time-out in the environment. So $P$ and $Q$ should be related in a triggered environment.

The last clause says that also a t-transition of $P$ should be matched by one of $Q$. Naturally, the t-transition of $P$ can be taken only when the system is idling in $P$, i.e., when $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$. The resulting processes $P'$ and $Q'$ should be related again, but only in the same environment allowing $X$.

**Proposition 2** Strong $X$-bisimilarity and strong reactive bisimilarity are equivalence relations.

**Proof:** $\leftrightarroweq_r^X$, $\leftrightarroweq_r$ are reflexive, as $\{(P, X, P), (P, P) \mid P \in \mathbb{P} \wedge X \subseteq A\}$ is a strong reactive bisimulation.
$\leftrightarroweq_r^X$ and $\leftrightarroweq_r$ are symmetric, since strong reactive bisimulations are symmetric by definition.
$\leftrightarroweq_r^X$ and $\leftrightarroweq_r$ are transitive, for if $\mathscr{R}$ and $\mathscr{S}$ are strong reactive bisimulations, then so is

$$\mathscr{R}; \mathscr{S} = \{(P, X, R) \mid \exists Q.\ (P, X, Q) \in \mathscr{R} \wedge (Q, X, R) \in \mathscr{S}\} \cup \{(P, R) \mid \exists Q.\ (P, Q) \in \mathscr{R} \wedge (Q, R) \in \mathscr{S}\}. \quad \square$$

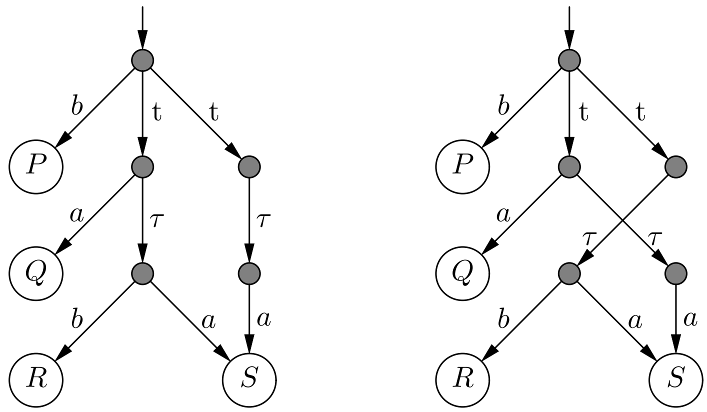


Figure 1: Two strongly reactive bisimilar processes

Note that the union of arbitrarily many strong reactive bisimulations is itself a strong reactive bisimulation. Therefore the family of relations $\underline{\leftrightarrow}_r$, $\underline{\leftrightarrow}_r^X$ for $X \subseteq A$ can be seen as a strong reactive bisimulation.

To get a firm grasp on strong reactive bisimilarity, the reader is invited to check the two laws mentioned in the introduction, and then to construct a strong reactive bisimulation between the two systems depicted in Figure 1. Here $P$, $Q$, $R$ and $S$ are arbitrary subprocesses. The four processes that are targets of $\mathrm{t}$-transitions always run in an environment that blocks $b$. In an environment that allows $a$, the branch $b.R$ disappears, so that the left branch of the first process can be matched with the left branch of the second process, and similarly for the two right branches. In an environment that blocks $a$, this matching won't fly, as the branch $b.R$ now survives. However, the branches $a.Q$ will disappear, so that the left branch of the first process can be matched with the right branch of the second, and vice versa.

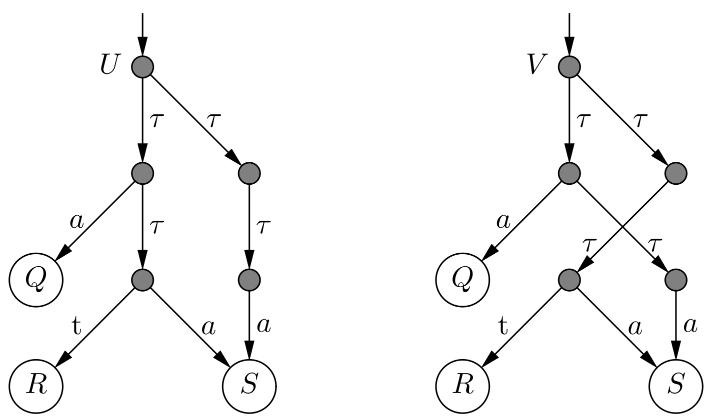


Figure 2: Reactive bisimilarity is not fully determined by reactive $X$-bisimilarity

The processes $U$ and $V$ of Figure 2 show that the pairs that occur in a strong reactive bisimulation are not completely determined by the triples. One has $U \;\underline{\leftrightarrow}_r^X\; V$ for any $X \subseteq A$, yet $U \;\not\underline{\leftrightarrow}_r\; V$. In particular, when $a \in X$ the branch $\mathrm{t}.R$ is redundant, and when $a \notin X$ the branch $a.Q$ is redundant.

Appendix C, reporting on work by Max Pohlmann [37], offers a context $\mathscr{C}$ with the property that $P \;\underline{\leftrightarrow}_r\; Q$ iff $\mathscr{C}(P) \;\underline{\leftrightarrow}\; \mathscr{C}(Q)$, thereby reducing strong reactive bisimilarity to strong bisimilarity. The context $\mathscr{C}$ places a system in a most general environment in which it could be running. This result allows any toolset for checking strong bisimilarity to be applicable for checking strong reactive bisimilarity.

### 2.1 A more general form of reactive bisimulation

The following notion of a *generalised strong reactive bisimulation* (gsrb) generalises that of a strong reactive bisimulation; yet it induces the same concept of strong reactive bisimilarity. This makes the relation convenient to use for further analysis. I did not introduce it as the original definition, because it lacks a strong motivation.

**Definition 3** A *gsrb* is a symmetric relation $\mathscr{R} \subseteq (\mathbb{P} \times \mathscr{P}(A) \times \mathbb{P}) \cup (\mathbb{P} \times \mathbb{P})$ such that, for all $(P,Q) \in \mathscr{R}$,

- if $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$, then there exists a $Q'$ such that $Q \xrightarrow{\alpha} Q'$ and $(P',Q') \in \mathscr{R}$,
- if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ with $X \subseteq A$ and $P \xrightarrow{\mathrm{t}} P'$, then $\exists Q'$ with $Q \xrightarrow{\mathrm{t}} Q'$ and $(P',X,Q') \in \mathscr{R}$,

and for all $(P,Y,Q) \in \mathscr{R}$,

- if $P \xrightarrow{a} P'$ with either $a \in Y$ or $\mathcal{I}(P) \cap (Y \cup \{\tau\}) = \emptyset$, then $\exists Q'$ with $Q \xrightarrow{a} Q'$ and $(P',Q') \in \mathscr{R}$,
- if $P \xrightarrow{\tau} P'$, then there exists a $Q'$ such that $Q \xrightarrow{\tau} Q'$ and $(P',Y,Q') \in \mathscr{R}$,
- if $\mathcal{I}(P) \cap (X \cup Y \cup \{\tau\}) = \emptyset$ with $X \subseteq A$ and $P \xrightarrow{\mathrm{t}} P'$ then $\exists Q'$ with $Q \xrightarrow{\mathrm{t}} Q'$ and $(P',X,Q') \in \mathscr{R}$.

Unlike Definition 1, a gsrb needs the triples $(P,X,Q)$ only after encountering a $\mathrm{t}$-transition; two systems without $\mathrm{t}$-transitions can be related without using these triples at all.

**Proposition 4** $P \leftrightarroweq_r Q$ iff there exists a gsrb $\mathscr{R}$ with $(P,Q) \in \mathscr{R}$.
Likewise, $P \leftrightarroweq_r^X Q$ iff there exists a gsrb $\mathscr{R}$ with $(P,X,Q) \in \mathscr{R}$.

**Proof:** Clearly, each strong reactive bisimulation satisfies the five clauses of Definition 3 and thus is a gsrb. In the other direction, given a gsrb $\mathscr{B}$, let

$$\begin{aligned}\mathscr{R} := \mathscr{B} &\cup \{(P,X,Q) \mid (P,Q) \in \mathscr{B} \wedge X \subseteq A\} \\ &\cup \{(P,Q),(P,X,Q) \mid \exists Y.\ (P,Y,Q) \in \mathscr{B} \wedge \mathcal{I}(P) \cap (Y \cup \{\tau\}) = \emptyset \wedge X \subseteq A\}\,.\end{aligned}$$

It is straightforward to check that $\mathscr{R}$ satisfies the six clauses of Definition 1. □

The above proof has been formalised in [37], using the interactive proof assistant Isabelle. The formalisation takes up around 250 lines of code.

## 3 A modal characterisation of strong reactive bisimilarity

The Hennessy-Milner logic [27] expresses properties of the behaviour of processes in an LTS.

**Definition 5** The class $\mathbb{O}$ of *infinitary HML formulas* is defined as follows, where $I$ ranges over all index sets and $\alpha$ over $A \cup \{\tau\}$:

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

$\top$ abbreviates the empty conjunction, and $\varphi_1 \wedge \varphi_2$ stands for $\bigwedge_{i \in \{1,2\}} \varphi_i$.

$P \models \varphi$ denotes that process $P$ satisfies formula $\varphi$. The first two operators represent the standard Boolean operators conjunction and negation. By definition, $P \models \langle\alpha\rangle\varphi$ iff $P \xrightarrow{\alpha} P'$ for some $P'$ with $P' \models \varphi$.

A famous result stemming from [27] states that

$$P \leftrightarroweq Q \ \Leftrightarrow\ \forall \varphi \in \mathbb{O}.\ (P \models \varphi \Leftrightarrow Q \models \varphi)$$

where $\leftrightarroweq$ denotes strong bisimilarity [33, 27], formally defined in Section 7.2. It states that the Hennessy-Milner logic yields a *modal characterisation* of strong bisimilarity. I will now adapt this result to obtain a modal characterisation of strong reactive bisimilarity.

To this end I extend the Hennessy-Milner logic with a new modality $\langle X\rangle$, for $X \subseteq A$, and auxiliary satisfaction relations $\models_X \subseteq \mathbb{P}\times\mathbb{O}$ for each $X \subseteq A$. The formula $P \models \langle X\rangle\varphi$ says that in an environment $X$, allowing exactly the actions in $X$, process $P$ can perform a time-out transition to a process that satisfies $\varphi$. $P \models_X \varphi$ says that $P$ satisfies $\varphi$ when placed in environment $X$. The relations $\models$ and $\models_X$ are the smallest ones satisfying:

$$
\begin{array}{lll}
P \models \bigwedge_{i\in I}\varphi_i & \text{if} & \forall i\in I.\ P\models\varphi_i\\
P \models \neg\varphi & \text{if} & P \not\models \varphi\\
P \models \langle\alpha\rangle\varphi \quad \text{with } \alpha\in A\cup\{\tau\} & \text{if} & \exists P'.\ P \stackrel{\alpha}{\longrightarrow} P' \wedge P'\models\varphi\\
P \models \langle X\rangle\varphi \quad \text{with } X\subseteq A & \text{if} & \mathcal{I}(P)\cap(X\cup\{\tau\})=\emptyset \wedge \exists P'.\ P\stackrel{\mathrm{t}}{\longrightarrow} P' \wedge P'\models_X\varphi
\end{array}
$$

$$
\begin{array}{lll}
P \models_X \bigwedge_{i\in I}\varphi_i & \text{if} & \forall i\in I.\ P\models_X\varphi_i\\
P \models_X \neg\varphi & \text{if} & P \not\models_X \varphi\\
P \models_X \langle a\rangle\varphi \quad \text{with } a\in A & \text{if} & a\in X \wedge \exists P'.\ P \stackrel{a}{\longrightarrow} P' \wedge P'\models\varphi\\
P \models_X \langle\tau\rangle\varphi & \text{if} & \exists P'.\ P \stackrel{\tau}{\longrightarrow} P' \wedge P'\models_X\varphi\\
P \models_X \varphi & \text{if} & \mathcal{I}(P)\cap(X\cup\{\tau\})=\emptyset \wedge P\models\varphi
\end{array}
$$

Note that a formula $\langle a\rangle\varphi$ is less often true under $\models_X$ than under $\models$, due to the side condition $a \in X$. This reflects the fact that $a$ cannot happen in an environment that blocks it. The last clause in the above definition reflects the fifth clause of Definition 1. If $\mathcal{I}(P)\cap(X\cup\{\tau\}) = \emptyset$, then process $P$, operating in environment $X$, idles for a while, during which the environment can change. This ends the blocking of actions $a \notin X$ and makes any formula valid under $\models$ also valid under $\models_X$.

**Example 6** Both systems from Figure 1 satisfy $\langle\emptyset\rangle\langle\tau\rangle\langle b\rangle\top \wedge \langle\emptyset\rangle\langle\tau\rangle\neg\langle b\rangle\top \wedge \langle\{a\}\rangle\langle a\rangle\top \wedge \langle\{a\}\rangle\neg\langle a\rangle\top$ and neither satisfies $\langle\emptyset\rangle(\langle a\rangle\top \wedge \langle\tau\rangle\langle b\rangle\top)$ or $\langle\{a\}\rangle(\langle a\rangle\top \wedge \langle\tau\rangle\langle b\rangle\top)$.

**Theorem 7** Let $P,Q\in\mathbb{P}$ and $X\subseteq A$. Then $P \leftrightarroweq_r Q \;\Leftrightarrow\; \forall\varphi\in\mathbb{O}.\ (P\models\varphi \Leftrightarrow Q\models\varphi)$
and $P \leftrightarroweq_r^X Q \;\Leftrightarrow\; \forall\varphi\in\mathbb{O}.\ (P\models_X\varphi \Leftrightarrow Q\models_X\varphi)$.

**Proof:** "$\Rightarrow$": I prove by simultaneous structural induction on $\varphi\in\mathbb{O}$ that, for all $P,Q\in\mathbb{P}$ and $X\subseteq A$, $P \leftrightarroweq_r Q \wedge P\models\varphi \;\Rightarrow\; Q\models\varphi$ and $P \leftrightarroweq_r^X Q \wedge P\models_X\varphi \;\Rightarrow\; Q\models_X\varphi$. For each $\varphi$, the converse implications ($Q\models\varphi \Rightarrow P\models\varphi$ and $Q\models_X\varphi \Rightarrow P\models_X\varphi$) follow by symmetry. In particular, these converse directions may be used when invoking the induction hypothesis.

- Let $P \leftrightarroweq_r Q \wedge P\models\varphi$.
  - Let $\varphi = \bigwedge_{i\in I}\varphi_i$. Then $P\models\varphi_i$ for all $i\in I$. By induction $Q\models\varphi_i$ for all $i$, so $Q\models\bigwedge_{i\in I}\varphi_i$.
  - Let $\varphi=\neg\psi$. Then $P\not\models\psi$. By induction $Q\not\models\psi$, so $Q\models\neg\psi$.
  - Let $\varphi=\langle\alpha\rangle\psi$ with $\alpha\in A\cup\{\tau\}$. Then $P\stackrel{\alpha}{\longrightarrow}P'$ for some $P'$ with $P'\models\psi$. By Definition 3, $Q\stackrel{\alpha}{\longrightarrow}Q'$ for some $Q'$ with $P' \leftrightarroweq_r Q'$. So by induction $Q'\models\psi$, and thus $Q\models\langle\alpha\rangle\psi$.
  - Let $\varphi=\langle X\rangle\psi$ for some $X\subseteq A$. Then $\mathcal{I}(P)\cap(X\cup\{\tau\})=\emptyset$ and $P\stackrel{\mathrm{t}}{\longrightarrow}P'$ for some $P'$ with $P'\models_X\psi$. By Definition 3, $Q\stackrel{\mathrm{t}}{\longrightarrow}Q'$ for some $Q'$ with $P' \leftrightarroweq_r^X Q'$. So by induction $Q'\models_X\psi$. Moreover, $\mathcal{I}(Q)=\mathcal{I}(P)$, as $P \leftrightarroweq_r Q$, so $\mathcal{I}(Q)\cap(X\cup\{\tau\})=\emptyset$. Thus $Q\models\langle X\rangle\psi$.
- Let $P \leftrightarroweq_r^X Q \wedge P\models_X\varphi$.
  - Let $\varphi=\bigwedge_{i\in I}\varphi_i$, and $P\models_X\varphi_i$ for all $i\in I$. By induction $Q\models_X\varphi_i$ for all $i\in I$, so $q\models_X\bigwedge_{i\in I}\varphi_i$.
  - Let $\varphi=\neg\psi$, and $P\not\models_X\psi$. By induction $Q\not\models_X\psi$, so $Q\models_X\neg\psi$.
  - Let $\varphi=\langle a\rangle\psi$ with $a\in X$ and $P\stackrel{a}{\longrightarrow}P'$ for some $P'$ with $P'\models\psi$. By Definition 1, $Q\stackrel{a}{\longrightarrow}Q'$ for some $Q'$ with $P' \leftrightarroweq_r Q'$. By induction $Q'\models\psi$, so $Q\models_X\langle a\rangle\psi$.

- Let $\varphi = \langle\tau\rangle\psi$, and $P \xrightarrow{\tau} P'$ for some $P'$ with $P' \models_X \psi$. By Definition 1, $Q \xrightarrow{\tau} Q'$ for some $Q'$ with $P' \leftrightarrow_r^X Q'$. By induction $Q' \models_X \psi$, so $Q \models_X \langle\tau\rangle\psi$.
- Let $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \models \varphi$. By the fifth clause of Definition 1, $P \leftrightarrow_r Q$. Hence, by the previous case in this proof, $Q \models \varphi$. Moreover, $\mathcal{I}(Q) \cap (X \cup \{\tau\}) = \mathcal{I}(P) \cap (X \cup \{\tau\})$, since $P \leftrightarrow_r^X Q$. Thus $Q \models_X \varphi$.

"$\Leftarrow$": Write $P \equiv Q$ for $\forall\varphi\in\mathbb{O}.\ (P \models \varphi \Leftrightarrow Q \models \varphi)$, and $P \equiv_X Q$ for $\forall\varphi\in\mathbb{O}.\ (P \models_X \varphi \Leftrightarrow Q \models_X \varphi)$. I show that the family of relations $\equiv$, $\equiv_X$ for $X \subseteq A$ constitutes a gsrb.

- Suppose $P \equiv Q$ and $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$. Let $\mathcal{Q}^\dagger := \{Q^\dagger \in \mathbb{P} \mid Q \xrightarrow{\alpha} Q^\dagger \wedge P' \not\equiv Q^\dagger\}$. For each $Q^\dagger \in \mathcal{Q}^\dagger$, let $\varphi_{Q^\dagger} \in \mathbb{O}$ be a formula such that $P' \models \varphi_{Q^\dagger}$ and $Q^\dagger \not\models \varphi_{Q^\dagger}$. (Such a formula always exists because $\mathbb{O}$ is closed under negation.) Define $\varphi := \bigwedge_{Q^\dagger\in\mathcal{Q}^\dagger} \varphi_{Q^\dagger}$. Then $P' \models \varphi$, so $P \models \langle a\rangle\varphi$. Consequently, also $Q \models \langle a\rangle\varphi$. Hence there is a $Q'$ with $Q \xrightarrow{\alpha} Q'$ and $Q' \models \varphi$. Since none of the $Q^\dagger \in \mathcal{Q}^\dagger$ satisfies $\varphi$, one obtains $Q' \notin \mathcal{Q}^\dagger$ and thus $P' \equiv Q'$.
- Suppose $P \equiv Q,\ X \subseteq A,\ \mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$. Let

$$\mathcal{Q}^\dagger := \{Q^\dagger \in \mathbb{P} \mid Q \xrightarrow{\mathrm{t}} Q^\dagger \wedge P' \not\equiv_X Q^\dagger\}.$$

  For each $Q^\dagger \in \mathcal{Q}^\dagger$, let $\varphi_{Q^\dagger} \in \mathbb{O}$ be a formula such that $P' \models_X \varphi_{Q^\dagger}$ and $Q^\dagger \not\models_X \varphi_{Q^\dagger}$. Define $\varphi := \bigwedge_{Q^\dagger\in\mathcal{Q}^\dagger} \varphi_{Q^\dagger}$. Then $P' \models_X \varphi$, so $P \models \langle X\rangle\varphi$. Consequently, also $Q \models \langle X\rangle\varphi$. Hence there is a $Q'$ with $Q \xrightarrow{\mathrm{t}} Q'$ and $Q' \models_X \varphi$. Again $Q' \notin \mathcal{Q}^\dagger$ and thus $P' \equiv_X Q'$.
- Suppose $P \equiv_Y Q$ and $P \xrightarrow{\alpha} P'$ with $a \in A$ and either $a \in Y$ or $\mathcal{I}(P) \cap (Y \cup \{\tau\}) = \emptyset$. Let $\mathcal{Q}^\dagger := \{Q^\dagger \in \mathbb{P} \mid Q \xrightarrow{\alpha} Q^\dagger \wedge P' \not\equiv Q^\dagger\}$. For each $Q^\dagger \in \mathcal{Q}^\dagger$, let $\varphi_{Q^\dagger} \in \mathbb{O}$ be a formula such that $P' \models \varphi_{Q^\dagger}$ and $Q^\dagger \not\models \varphi_{Q^\dagger}$. Define $\varphi := \bigwedge_{Q^\dagger\in\mathcal{Q}^\dagger} \varphi_{Q^\dagger}$. Then $P' \models \varphi$, so $P \models \langle a\rangle\varphi$, and also $P \models_Y \langle a\rangle\varphi$, using either the third or last clause in the definition of $\models_X$. Hence also $Q \models_Y \langle a\rangle\varphi$. Therefore there is a $Q'$ with $Q \xrightarrow{\alpha} Q'$ and $Q' \models \varphi$, using the third clause of either $\models_X$ or $\models$. Since none of the $Q^\dagger \in \mathcal{Q}^\dagger$ satisfies $\varphi$, one obtains $Q' \notin \mathcal{Q}^\dagger$ and thus $P' \equiv Q'$.
- The fourth clause of Definition 3 is obtained exactly like the first, but using $\models_Y$ instead of $\models$.
- Suppose $P \equiv_Y Q$, $P \xrightarrow{\mathrm{t}} P'$ and $\mathcal{I}(P) \cap (X \cup Y \cup \{\tau\}) = \emptyset$, with $X \subseteq A$. Let

$$\mathcal{Q}^\dagger := \{Q^\dagger \in \mathbb{P} \mid Q \xrightarrow{\mathrm{t}} Q^\dagger \wedge P' \not\equiv_X Q^\dagger\}.$$

  For each $Q^\dagger \in \mathcal{Q}^\dagger$, let $\varphi_{Q^\dagger} \in \mathbb{O}$ be a formula such that $P' \models_X \varphi_{Q^\dagger}$ and $Q^\dagger \not\models_X \varphi_{Q^\dagger}$. Define $\varphi := \bigwedge_{Q^\dagger\in\mathcal{Q}^\dagger} \varphi_{Q^\dagger}$. Then $P' \models_X \varphi$, so $P \models \langle X\rangle\varphi$, and thus $P \models_Y \langle X\rangle\varphi$. Consequently, also $Q \models_Y \langle X\rangle\varphi$ and therefore $Q \models \langle X\rangle\varphi$. Hence there is a $Q'$ with $Q \xrightarrow{\mathrm{t}} Q'$ and $Q' \models_X \varphi$. Again $Q' \notin \mathcal{Q}^\dagger$ and thus $P' \equiv_X Q'$. □

## 4 Time-out bisimulations

I will now present a characterisation of strong reactive bisimilarity in terms of a binary relation $\mathscr{B}$ on processes—a *strong time-out bisimulation*—not parametrised by the set of allowed actions $X$. To this end I need a family of unary operators $\theta_X$ on processes, for $X \subseteq A$. These *environment* operators place a process in an environment that allows exactly the actions in $X$ to occur. They are defined by the following structural operational rules.

$$\frac{x \xrightarrow{\tau} y}{\theta_X(x) \xrightarrow{\tau} \theta_X(y)} \qquad \frac{x \xrightarrow{a} y}{\theta_X(x) \xrightarrow{a} y}\ (a \in X) \qquad \frac{x \xrightarrow{\alpha} y \quad x \stackrel{\beta}{\nrightarrow} \text{ for all } \beta \in X \cup \{\tau\}}{\theta_X(x) \xrightarrow{\alpha} y}\ (\alpha \in A \cup \{\mathrm{t}\})$$

The operator $\theta_X$ modifies its argument by inhibiting all initial transitions (here including also those that occur after a $\tau$-transition) that cannot occur in the specified environment. When an observable transition does occur, the environment may be triggered to change, and the inhibiting effect of the $\theta_X$-operator comes to an end. The premises $x \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$ in the third rule guarantee that the process $x$ will idle for a positive amount of time in its current state. During this time, the environment may be triggered to change, and again the inhibiting effect of the $\theta_X$-operator comes to an end.

Below I assume that $\mathbb{P}$ is closed under $\theta$, that is, if $P \in \mathbb{P}$ and $X \subseteq A$ then $\theta_X(P) \in \mathbb{P}$.

**Definition 8** A *strong time-out bisimulation* is a symmetric relation $\mathscr{B} \subseteq \mathbb{P} \times \mathbb{P}$, such that, for $P \mathrel{\mathscr{B}} Q$,

- if $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$, then $\exists Q'$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \mathrel{\mathscr{B}} Q'$,
- if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$, then $\exists Q'$ such that $Q \xrightarrow{\mathrm{t}} Q'$ and $\theta_X(P') \mathrel{\mathscr{B}} \theta_X(Q')$.

**Proposition 9** $P \leftrightarroweq_r Q$ iff there exists a strong time-out bisimulation $\mathscr{B}$ with $P \mathrel{\mathscr{B}} Q$.

**Proof:** Let $\mathscr{R}$ be a gsrb on $\mathbb{P}$. Define $\mathscr{B} \subseteq \mathbb{P} \times \mathbb{P}$ by $P \mathrel{\mathscr{B}} Q$ iff either $(P,Q) \in \mathscr{R}$ or $P = \theta_X(P^\dagger)$, $Q = \theta_X(Q^\dagger)$ and $(P^\dagger, X, Q^\dagger) \in \mathscr{R}$. I show that $\mathscr{B}$ is a strong time-out bisimulation.

- Let $P \mathrel{\mathscr{B}} Q$ and $P \xrightarrow{a} P'$ with $a \in A$. First suppose $(P,Q) \in \mathscr{R}$. Then, by the first clause of Definition 3, there exists a $Q'$ such that $Q \xrightarrow{a} Q'$ and $(P',Q') \in \mathscr{R}$. So $P' \mathrel{\mathscr{B}} Q'$.

  Next suppose $P = \theta_X(P^\dagger)$, $Q = \theta_X(Q^\dagger)$ and $(P^\dagger, X, Q^\dagger) \in \mathscr{R}$. Since $\theta_X(P^\dagger) \xrightarrow{a} P'$ it must be that $P^\dagger \xrightarrow{a} P'$ and either $a \in X$ or $P^\dagger \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Hence there exists a $Q'$ such that $Q^\dagger \xrightarrow{a} Q'$ and $(P',Q') \in \mathscr{R}$, using the third clause of Definition 3. Recall that $P^\dagger \leftrightarroweq_r^X Q^\dagger$ implies $I(P^\dagger) \cap (X \cup \{\tau\}) = I(Q^\dagger) \cap (X \cup \{\tau\})$, and thus either $a \in X$ or $Q^\dagger \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. It follows that $Q = \theta_X(Q^\dagger) \xrightarrow{a} Q'$ and $P' \mathrel{\mathscr{B}} Q'$.
- Let $P \mathrel{\mathscr{B}} Q$ and $P \xrightarrow{\tau} P'$. First suppose $(P,Q) \in \mathscr{R}$. Then, using the first clause of Definition 3, there is a $Q'$ with $Q \xrightarrow{\tau} Q'$ and $(P',Q') \in \mathscr{R}$. So $P' \mathrel{\mathscr{B}} Q'$.

  Next suppose $P = \theta_X(P^\dagger)$, $Q = \theta_X(Q^\dagger)$ and $(P^\dagger, X, Q^\dagger) \in \mathscr{R}$. Since $\theta_X(P^\dagger) \xrightarrow{\tau} P'$, it must be that $P'$ has the form $\theta_X(P^\ddagger)$, and $P^\dagger \xrightarrow{\tau} P^\ddagger$. Thus, by the fourth clause of Definition 3, there is a $Q^\ddagger$ with $Q^\dagger \xrightarrow{\tau} Q^\ddagger$ and $(P^\ddagger, X, Q^\ddagger) \in \mathscr{R}$. Now $Q = \theta_X(Q^\dagger) \xrightarrow{\tau} \theta_X(Q^\ddagger) =: Q'$ and $P' \mathrel{\mathscr{B}} Q'$.
- Let $P \mathrel{\mathscr{B}} Q$, $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$. First suppose $(P,Q) \in \mathscr{R}$. Then, by the second clause of Definition 3, there is a $Q'$ with $Q \xrightarrow{\mathrm{t}} Q'$ and $(P', X, Q') \in \mathscr{R}$. So $\theta_X(P') \mathrel{\mathscr{B}} \theta_X(Q')$.

  Next suppose $P = \theta_Y(P^\dagger)$, $Q = \theta_Y(Q^\dagger)$ and $(P^\dagger, Y, Q^\dagger) \in \mathscr{R}$. Since $\theta_Y(P^\dagger) \xrightarrow{\mathrm{t}} P'$, it must be that $P^\dagger \xrightarrow{\mathrm{t}} P'$ and $P^\dagger \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in Y \cup \{\tau\}$. Consequently, $\mathcal{I}(P^\dagger) = \mathcal{I}(P)$ and thus $\mathcal{I}(P^\dagger) \cap (X \cup Y \cup \{\tau\}) = \emptyset$. By the last clause of Definition 3 there is a $Q'$ such that $Q^\dagger \xrightarrow{\mathrm{t}} Q'$ and $(P, X, Q') \in \mathscr{R}$. So $\theta_X(P') \mathrel{\mathscr{B}} \theta_X(Q')$. From $(P^\dagger, Y, Q^\dagger) \in \mathscr{R}$ and $\mathcal{I}(P^\dagger) \cap (Y \cup \{\tau\}) = \emptyset$, I infer $\mathcal{I}(Q^\dagger) \cap (Y \cup \{\tau\}) = \emptyset$. So $Q^\dagger \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in Y \cup \{\tau\}$. This yields $Q = \theta_Y(Q^\dagger) \xrightarrow{\mathrm{t}} Q'$.

Now let $\mathscr{B}$ be a time-out bisimulation. Define $\mathscr{R} \subseteq \mathbb{P} \times \mathscr{P}(A) \times \mathbb{P}$ by $(P,Q) \in \mathscr{R}$ iff $P \mathrel{\mathscr{B}} Q$, and $(P,X,Q) \in \mathscr{R}$ iff $\theta_X(P) \mathrel{\mathscr{B}} \theta_X(Q)$. I need to show that $\mathscr{R}$ is a gsrb.

- Suppose $(P,Q) \in \mathscr{R}$ and $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$. Then $P \mathrel{\mathscr{B}} Q$, so there is a $Q'$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \mathrel{\mathscr{B}} Q'$. Hence $(P',Q') \in \mathscr{R}$.
- Suppose $(P,Q) \in \mathscr{R}$, $X \subseteq A$, $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$. Then $P \mathrel{\mathscr{B}} Q$, so $\exists Q'$ such that $Q \xrightarrow{\mathrm{t}} Q'$ and $\theta_X(P') \mathrel{\mathscr{B}} \theta_X(Q')$. Thus $(P',X,Q') \in \mathscr{R}$.
- Suppose $(P,X,Q) \in \mathscr{R}$ and $P \xrightarrow{a} P'$ with either $a \in X$ or $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$. Then $\theta_X(P) \mathrel{\mathscr{B}} \theta_X(Q)$. Moreover, $\theta_X(P) \xrightarrow{a} P'$. Hence there is a $Q'$ such that $\theta_X(Q) \xrightarrow{a} Q'$ and $P' \mathrel{\mathscr{B}} Q'$. It must be that $Q \xrightarrow{a} Q'$. Moreover, $(P',Q') \in \mathscr{R}$.

- Suppose $(P,X,Q) \in \mathscr{R}$ and $P \xrightarrow{\tau} P'$. Then $\theta_X(P) \mathrel{\mathscr{B}} \theta_X(Q)$. Since $P \xrightarrow{\tau} P'$, one has $\theta_X(P) \xrightarrow{\tau} \theta_X(P')$. Hence there is an $R$ such that $\theta_X(Q) \xrightarrow{\tau} R$ and $\theta_X(P') \mathrel{\mathscr{B}} R$. The process $R$ must have the form $\theta_X(Q')$ for some $Q'$ with $Q \xrightarrow{\tau} Q'$. It follows that $(P',X,Q') \in \mathscr{R}$.
- Suppose $(P,Y,Q) \in \mathscr{R}$, $X \subseteq A$, $\mathcal{I}(P) \cap (X \cup Y \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$. Then $\theta_Y(P) \mathrel{\mathscr{B}} \theta_Y(Q)$ and $\theta_Y(P) \xrightarrow{\mathrm{t}} P'$. Moreover, $\mathcal{I}(\theta_Y(P)) = \mathcal{I}(P)$, so by the second clause of Definition 8 there exists a $Q'$ such that $\theta_Y(Q) \xrightarrow{\mathrm{t}} Q'$ and $\theta_X(P') \mathrel{\mathscr{B}} \theta_X(Q')$. So $Q \xrightarrow{\mathrm{t}} Q'$ and $(P',X,Q') \in \mathscr{R}$. □

Note that the union of arbitrarily many strong time-out bisimulations is itself a strong time-out bisimulation. Consequently, the relation $\leftrightarrows_r$ is a strong time-out bisimulation.

## 5 The process algebra $\text{CCSP}_\mathrm{t}^\theta$

Let $A$ be a set of *visible actions* and $Var$ an infinite set of *variables*. The syntax of $\text{CCSP}_\mathrm{t}^\theta$ is given by

$$E ::= 0 \mid \alpha.E \mid E+E \mid E \|_S E \mid \tau_I(E) \mid \mathcal{R}(E) \mid \theta_L^U(E) \mid \psi_X(E) \mid x \mid \langle x|\mathcal{S}\rangle \text{ (with } x \in V_\mathcal{S})$$

with $\alpha \in Act := A \uplus \{\tau, \mathrm{t}\}$, $S,I,U,L,X \subseteq A$, $L \subseteq U$, $\mathcal{R} \subseteq A \times A$, $x \in Var$ and $\mathcal{S}$ a *recursive specification*: a set of equations $\{y = \mathcal{S}_y \mid y \in V_\mathcal{S}\}$ with $V_\mathcal{S} \subseteq Var$ (the *bound variables* of $\mathcal{S}$) and each $\mathcal{S}_y$ a $\text{CCSP}_\mathrm{t}^\theta$ expression. I require that all sets $\{b \mid (a,b) \in \mathcal{R}\}$ are finite.

The constant 0 represents a process that is unable to perform any action. The process $\alpha.E$ first performs the action $\alpha$ and then proceeds as $E$. The process $E+F$ behaves as either $E$ or $F$. $\|_S$ is a partially synchronous parallel composition operator; actions $a \in S$ must synchronise—they can occur only when both arguments are ready to perform them—whereas actions $\alpha \notin S$ from both arguments are interleaved. $\tau_I$ is an abstraction operator; it conceals the actions in $I$ by renaming them into the hidden action $\tau$. The operator $\mathcal{R}$ is a relational renaming: it renames a given action $a \in A$ into a choice between all actions $b$ with $(a,b) \in \mathcal{R}$. The *environment operators* $\theta_L^U$ and $\psi_X$ are new in this paper and explained below. Finally, $\langle x|\mathcal{S}\rangle$ represents the $x$-component of a solution of the system of recursive equations $\mathcal{S}$.

The language CCSP is a common mix of the process algebras CCS [33] and CSP [6, 28]. It first appeared in [34], where it was named following a suggestion by M. Nielsen. The family of parallel composition operators $\|_S$ stems from [35], and incorporates the two CSP parallel composition operators from [6]. The relation renaming operators $\mathcal{R}(\_)$ stem from [39]; they combine both the (functional) renaming operators that are common to CCS and CSP, and the inverse image operators of CSP. The choice operator $+$ stems from CCS, and the abstraction operator from CSP, while the inaction constant 0, action prefixing operators $a.\_$ for $a \in A$, and the recursion construct are common to CCS and CSP. The time-out prefixing operator $\mathrm{t}.\_$ was added by me in [18]. The syntactic form of inaction 0, action prefixing $\alpha.E$ and choice $E+F$ follows CCS, whereas the syntax of abstraction $\tau_I(\_)$ and recursion $\langle x|\mathcal{S}\rangle$ follows ACP [2, 10]. The fragment of $\text{CCSP}_\mathrm{t}^\theta$ without $\theta_L^U$ and $\psi_X$ is called $\text{CCSP}_\mathrm{t}$ [18].

An occurrence of a variable $x$ in a $\text{CCSP}_\mathrm{t}^\theta$ expression $E$ is *bound* iff it occurs in a subexpression $\langle y|\mathcal{S}\rangle$ of $E$ with $x \in V_\mathcal{S}$; otherwise it is *free*. Here each $\mathcal{S}_y$ for $y \in V_\mathcal{S}$ counts as a subexpression of $\langle x|\mathcal{S}\rangle$. An expression $E$ is *invalid* if it has a subexpression $\theta_L^U(F)$ or $\psi_X(F)$ such that a variable occurrence in $F$ is free in $F$ but bound in $E$. Let $\mathbb{E}$ be the set of valid $\text{CCSP}_\mathrm{t}^\theta$ expressions. Furthermore, $\mathbb{P} \subseteq \mathbb{E}$ is the set of *closed* valid $\text{CCSP}_\mathrm{t}^\theta$ expressions, or *processes*; those in which every variable occurrence is bound.

A substitution is a partial function $\rho\colon Var \rightharpoonup \mathbb{E}$. The application $E[\rho]$ of a substitution $\rho$ to an expression $E \in \mathbb{E}$ is the result of simultaneous replacement, for all $x \in \mathrm{dom}(\rho)$, of each free occurrence of $x$ in $E$ by the expression $\rho(x)$, while renaming bound variables in $E$ if necessary to prevent name clashes.

Table 1: Structural operational interleaving semantics of $\mathrm{CCSP}_{\mathrm{t}}^{\theta}$

$$\alpha.x \xrightarrow{\alpha} x \qquad \frac{x \xrightarrow{\alpha} x'}{x+y \xrightarrow{\alpha} x'} \qquad \frac{y \xrightarrow{\alpha} y'}{x+y \xrightarrow{\alpha} y'} \qquad \frac{x \xrightarrow{\alpha} x'}{\mathcal{R}(x) \xrightarrow{\beta} \mathcal{R}(x')} \begin{pmatrix} \alpha=\beta=\tau \\ \vee\ \alpha=\beta=\mathrm{t} \\ \vee\ (\alpha,\beta)\in\mathcal{R} \end{pmatrix}$$

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

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

$$\frac{x \xrightarrow{\tau} y}{\theta_L^U(x) \xrightarrow{\tau} \theta_L^U(y)} \qquad \frac{x \xrightarrow{a} y}{\theta_L^U(x) \xrightarrow{a} y}\ (a \in U) \qquad \frac{x \xrightarrow{\alpha} y \quad x \not\xrightarrow{\beta} \text{ for all } \beta \in L \cup \{\tau\}}{\theta_L^U(x) \xrightarrow{\alpha} y}\ (\alpha \in A \cup \{\mathrm{t}\})$$

$$\frac{x \xrightarrow{\alpha} y}{\psi_X(x) \xrightarrow{\alpha} y}\ (\alpha \in A \cup \{\tau\}) \qquad \frac{x \xrightarrow{\mathrm{t}} y \quad x \not\xrightarrow{\beta} \text{ for all } \beta \in X \cup \{\tau\}}{\psi_X(x) \xrightarrow{\mathrm{t}} \theta_X(y)}$$

The semantics of $\mathrm{CCSP}_{\mathrm{t}}^{\theta}$ is given by the labelled transition relation $\rightarrow\ \subseteq \mathbb{P} \times Act \times \mathbb{P}$, where the transitions $P \xrightarrow{\alpha} Q$ are derived from the rules of Table 1. Here $\langle E|\mathcal{S}\rangle$ for $E \in \mathbb{E}$ and $\mathcal{S}$ a recursive specification denotes the result of substituting $\langle y|\mathcal{S}\rangle$ for $y$ in $E$, for all $y \in V_{\mathcal{S}}$.

The auxiliary operators $\theta_L^U$ and $\psi_X$ are added here to facilitate complete axiomatisation, similar to the left merge and communication merge of ACP [2, 10]. The operator $\theta_X^X$ is the same as what was called $\theta_X$ in Section 4. It inhibits those transitions of its argument that are blocked in the environment $X$, allowing only the actions from $X \subseteq A$. It stops inhibiting as soon as the system performs a visible action or takes a break, as this may trigger a change in the environment. The operator $\theta_L^U$ preserves those transitions that are allowed in some environment $X$ with $L \subseteq X \subseteq U$. The letters $L$ and $U$ stand for *lower* and *upper* bound. The operator $\psi_X$ places a process in the environment $X$ when a time-out transition occurs; it is inert if any other transition occurs. If $P \xrightarrow{\beta}$ for $\beta \in A \cup \{\tau\}$, then a time-out transition $P \xrightarrow{\mathrm{t}} Q$ cannot occur in an environment that allows $\beta$. Thus the transition $P \xrightarrow{\mathrm{t}} Q$ survives only when considering an environments that blocks $\beta$, meaning $\beta \notin X \cup \{\tau\}$. Taking the contrapositive, $\beta \in X \cup \{\tau\}$ implies $P \not\xrightarrow{\beta}$.

The operator $\theta_\emptyset^U$ features in the forthcoming law L3, which is a convenient addition to my axiomatisation, although only $\psi_X$ and $\theta_X$ $(= \theta_X^X)$ are necessary for completeness.

**Stratification.** Even though negative premises occur in Table 1, the meaning of this transition system specification is well-defined, for instance by the method of *stratification* explained in [25, 15]. Assign inductively to each expression $E \in \mathbb{E}$ an ordinal $\lambda_E$ that counts the nesting depth of recursive specifications: if $E = \langle x|\mathcal{S}\rangle$ then $\lambda_E$ is 1 more than the supremum of the $\lambda_{S_y}$ for $y \in V_{\mathcal{S}}$; otherwise $\lambda_E$ is the supremum of $\lambda\langle x|\mathcal{S}\rangle$ for all subterms $\langle x|\mathcal{S}\rangle$ of $E$. Moreover $\kappa_E \in \mathbb{N}$ is the nesting depth of $\theta_L^U$ and $\psi_X$ operators in $E$ that remain after replacing any subterm $F$ of $E$ with $\lambda_F < \lambda_E$ by 0. Now the ordered pair $(\lambda_P, \kappa_P)$ constitutes a valid stratification for closed literals $P \xrightarrow{\alpha} P'$. Namely, whenever a transition $P \xrightarrow{\alpha} P'$ depends on a transition $Q \xrightarrow{\beta} Q'$, in the sense that that there is a closed substitution instance $\mathfrak{r}$ of a rule from Table 1 with conclusion $P \xrightarrow{\alpha} P'$, and $Q \xrightarrow{\beta} Q'$ occurring in its premises, then either $\lambda_Q < \lambda_P$, or $\lambda_Q = \lambda_P$ and $\kappa_Q \leq \kappa_P$. Moreover, when $P \xrightarrow{\alpha} P'$ depends on a negative literal $Q \not\xrightarrow{\beta}$, then $\lambda_Q = \lambda_P$ and $\kappa_Q < \kappa_P$.

The above argument hinges on the exclusion of invalid $\mathrm{CCSP}^\theta_{\rm t}$ expressions. The invalid expression $P := \langle x \mid \{x = \theta^{\{a\}}_{\{a\}}(b.0 + \mathcal{R}(x))\}\rangle$ for instance, with $\mathcal{R} = \{(b,a)\}$, does not have a well-defined meaning, since the transition $P \stackrel{b}{\longrightarrow} 0$ is derivable iff one has the premise $P \stackrel{b}{\not\longrightarrow}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{b.0 \stackrel{b}{\longrightarrow} 0}}{b.0 + \mathcal{R}(P) \stackrel{b}{\longrightarrow} 0} \qquad \dfrac{\dfrac{P \stackrel{b}{\not\longrightarrow}}{\mathcal{R}(P) \stackrel{a}{\not\longrightarrow}}}{b.0 + \mathcal{R}(P) \stackrel{a}{\not\longrightarrow}} \qquad \dfrac{\dfrac{P \stackrel{\tau}{\not\longrightarrow}\ \ (\text{OK})}{\mathcal{R}(P) \stackrel{\tau}{\not\longrightarrow}}}{b.0 + \mathcal{R}(P) \stackrel{\tau}{\not\longrightarrow}}}{\theta^{\{a\}}_{\{a\}}(b.0 + \mathcal{R}(P)) \stackrel{b}{\longrightarrow} 0}}{P \stackrel{b}{\longrightarrow} 0}$$

However, the meaning of the valid expression $\langle x \mid \{x = \theta^{\{a\}}_{\{a\}}(\langle y|\{y = b.y\}\rangle) \,\|_\emptyset\, \mathcal{R}(x)\}\rangle$, for instance, is entirely unproblematic.

## 6 Guarded recursion and finitely branching processes

In many process algebraic specification approaches, only guarded recursive specifications are allowed.

**Definition 10** An occurrence of a variable $x$ in an expression $E$ is *guarded* if $x$ occurs in a subexpression $\alpha.F$ of $E$, with $\alpha \in Act$. An expression $E$ is *guarded* if all free occurrences of variables in $E$ are guarded. A recursive specification $\mathcal{S}$ is *manifestly guarded* if all expressions $\mathcal{S}_y$ for $y \in V_\mathcal{S}$ are guarded. It is *guarded* if it can be converted into a manifestly guarded recursive specification by repeated substitution of expressions $\mathcal{S}_y$ for variables $y \in V_\mathcal{S}$ occurring in the expressions $\mathcal{S}_z$ for $z \in V_\mathcal{S}$. Let *guarded* $\mathrm{CCSP}^\theta_{\rm t}$ be the fragment of $\mathrm{CCSP}^\theta_{\rm t}$ allowing only guarded recursion.

**Definition 11** The set of processes *reachable* from a given process $P \in \mathbb{P}$ is inductively defined by

(i) $P$ is reachable from $P$, and

(ii) if $Q$ is reachable from $P$ and $Q \stackrel{\alpha}{\longrightarrow} R$ for some $\alpha \in Act$ then $R$ is reachable from $P$.

A process $P$ is *finitely branching* if for all $Q \in \mathbb{P}$ reachable from $P$ there are only finitely many pairs $(\alpha, R)$ such that $Q \stackrel{\alpha}{\longrightarrow} R$. Likewise, $P$ is *countably branching* if there are countably many such pairs. A process is *finite* iff it is finitely branching, has finitely many reachable states, and is loop-free, in the sense that there are no $Q_0 \stackrel{\alpha_1}{\longrightarrow} Q_1 \stackrel{\alpha_2}{\longrightarrow} \cdots \stackrel{\alpha_n}{\longrightarrow} Q_n$ with $n > 0$ and $Q_0 = Q_n$ reachable from $P$.

**Proposition 12** Each $\mathrm{CCSP}^\theta_{\rm t}$ process is countably branching.

**Proof:** I show that for each $\mathrm{CCSP}^\theta_{\rm t}$ process $Q$ there are only countably many transitions $Q \stackrel{\alpha}{\longrightarrow} R$. Each such transition must be derivable from the rules of Table 1. So it suffices to show that for each $Q$ there are only countably many derivations of transitions $Q \stackrel{\alpha}{\longrightarrow} R$.

A derivation of a transition is a well-founded, upwardly branching tree, in which each node models an application of one of the rules of Table 1. Since each of these rules has finitely many positive premises, such a proof tree is finitely branching, and thus finite. Let $d(\pi)$, the *depth* of $\pi$, be the length of the longest branch in a derivation $\pi$. If $\pi$ derives a transition $Q \stackrel{\alpha}{\longrightarrow} R$, then I call $Q$ the *source* of $\pi$.

It suffices to show that for each $n \in \mathbb{N}$ there are only finitely many derivations of depth $n$ with a given source. This I do by induction on $n$.

In case $Q = f(Q_1, \ldots, Q_k)$, with $f$ an $k$-ary $\mathrm{CCSP}^\theta_{\rm t}$ operator, a derivation $\pi$ of depth $n$ is completely determined by the concluding rule from Table 1, deriving a transition $Q \stackrel{\beta}{\longrightarrow} R$, the subderivations of $\pi$ with source $Q_i$ for some of the $i \in \{1, \ldots, k\}$, and the transition label $\beta$. (For the purposes of this proof, Table 1 is understood to have only 15 rules, even if each of them can be seen as a template, with an instance for each choice of $\mathcal{R}$, $S$, $I$, $\mathcal{S}$ etc., and for each fitting choice of a transition labels $a$, $\alpha$ and/or $\beta$.) The choice of the concluding rule depends on $f$, and for each $f$ there are at most three choices. The subderivations of $\pi$ with source $Q_i$ have depth $< n$, so by induction there are only finitely many. When $f$ is not a renaming operator $\mathcal{R}$, there is no further choice for the transition label $\beta$, as it is completely determined by the premises of the rule, and thus by the subderivations of those premises. In case $f = \mathcal{R}$, there are finitely many choices for $\beta$ when faced with a given transition label $\alpha$ contributed by the premise of the rule for renaming. Here I use the requirement of Section 5 that all sets $\{b \mid (a,b) \in \mathcal{R}\}$ are finite. This shows there are only finitely many choices for $\pi$.

In case $Q = \langle x|\mathcal{S}\rangle$, the last step in $\pi$ must be application of the rule for recursion, so $\pi$ is completely determined by a subderivation $\pi'$ of a transition with source $\langle \mathcal{S}_x|\mathcal{S}\rangle$. By induction there are only finitely many choices for $\pi'$, and hence also for $\pi$. □

**Proposition 13** Each $\text{CCSP}_\text{t}^\theta$ process with guarded recursion is finitely branching.

**Proof:** A trivial structural induction shows that if $P$ is a $\text{CCSP}_\text{t}^\theta$ process with guarded recursion and $Q$ is reachable from $P$, then also $Q$ has guarded recursion. Hence it suffices to show that for each $\text{CCSP}_\text{t}^\theta$ process $Q$ with guarded recursion there are only finitely many derivations with source $Q$.

Let $\rightsquigarrow$ be the smallest binary relation on $\mathbb{P}$ such that (i) $f(P_1,\dots,P_k) \rightsquigarrow P_i$ for each $k$-ary $\text{CCSP}_\text{t}^\theta$ operator $f$ except action prefixing, and each $i \in \{1,\dots,k\}$, and (ii) $\langle x|\mathcal{S}\rangle \rightsquigarrow \langle \mathcal{S}_x|\mathcal{S}\rangle$. This relation is finitely branching. Moreover, on processes with guarded recursion, $\rightsquigarrow$ has no forward infinite chains $P_0 \rightsquigarrow P_1 \rightsquigarrow \dots$. In fact, this could have been used as an alternative definition of guarded recursion. Let, for any process $Q$ with guarded recursion, $e(Q)$ be the length of the longest forward chain $Q \rightsquigarrow P_1 \rightsquigarrow \cdots \rightsquigarrow P_{e(Q)}$. I show with induction on $e(Q)$ that there are only finitely many derivations with source $Q$. In fact, this proceeds exactly as in the previous proof. □

**Proposition 14 ([13])** Each finitely branching processes in an LTS can be denoted by a closed $\text{CCSP}_\text{t}$ expression with guarded recursion. Here I only need the operations inaction (0), action prefixing ($\alpha.\_$) and choice ($+$), as well as recursion $\langle x|\mathcal{S}\rangle$.

**Proof:** Let $P$ be a finitely branching process in an LTS $(\mathbb{P}', Act, \rightarrow)$. Let

$$V_\mathcal{S} := \{x_Q \mid Q \in \mathbb{P}' \text{ is reachable from } P\} \subseteq Var.$$

For each $Q$ reachable from $P$, let *next*$(Q)$ be the finite set of pairs $(\alpha, R) \in Act \times \mathbb{P}'$ such that there is a transition $Q \xrightarrow{\alpha} R$. Define the recursive specification $\mathcal{S}$ as $\{x_Q = \sum_{(\alpha,R)\in next(Q)} \alpha.x_R \mid x_Q \in V_\mathcal{S}\}$. Here the finite choice operator $\sum_{i\in I} \alpha_i.P_i$ can easily be expressed in terms of inaction, action prefixing and choice. Now the $\text{CCSP}_\text{t}$ process $\langle x_P|\mathcal{S}\rangle$ denotes $P$. □

In fact, $\langle x_P|\mathcal{S}\rangle \leftrightarrow P$, where $\leftrightarrow$ denotes strong bisimilarity [33], formally defined in the next section.

Likewise, recursion-free $\text{CCSP}_\text{t}^\theta$ processes are finite, and, up to strong bisimilarity, each finite process is denotable by a closed recursion-free $\text{CCSP}_\text{t}^\theta$ expression, using only 0, $\alpha.\_$ and $+$.

**Proposition 15 ([13])** Each countably branching processes in an LTS can be denoted by a closed $\text{CCSP}_\text{t}$ expression. Again I only need the $\text{CCSP}_\text{t}$ operations inaction, action prefixing, choice and recursion.

**Proof:** The proof is the same as the previous one, except that *next*$(Q)$ now is a countable set, rather than a finite one, and consequently I need a countable choice operator $\sum_{i\in\mathbb{N}} \alpha_i.P_i$. The latter can be expressed in $\text{CCSP}_\text{t}$ with unguarded recursion by $\sum_{i\in\mathbb{N}} \alpha_i.P_i := \langle z_o|\{z_i = \alpha_i.P_i + z_{i+1} \mid i \in \mathbb{N}\}\rangle$. □

# 7 Congruence

Given an arbitrary process algebra with a collection of operators $f$, each with an arity $n$, and a recursion construct $\langle x|\mathcal{S}\rangle$ as in Section 5, let $\mathbb{P}$ and $\mathbb{E}$ be the sets of [closed] valid expressions, and let a substitution instance $E[\rho] \in \mathbb{E}$ for $E \in \mathbb{E}$ and $\rho : \mathit{Var} \rightharpoonup \mathbb{E}$ be defined as in Section 5. Any semantic equivalence $\sim\, \subseteq \mathbb{P} \times \mathbb{P}$ extends to $\sim\, \subseteq \mathbb{E} \times \mathbb{E}$ by defining $E \sim F$ iff $E[\rho] \sim F[\rho]$ for each closed substitution $\rho : \mathit{Var} \to \mathbb{P}$. It extends to substitutions $\rho, \nu : \mathit{Var} \rightharpoonup \mathbb{E}$ by $\rho \sim \nu$ iff $\mathrm{dom}(\rho) = \mathrm{dom}(\nu)$ and $\rho(x) \sim \nu(x)$ for each $x \in \mathrm{dom}(\rho)$.

**Definition 16 ([16])** A semantic equivalence $\sim$ is a *lean congruence* if $E[\rho] \sim E[\nu]$ for any expression $E \in \mathbb{E}$ and any substitutions $\rho$ and $\nu$ with $\rho \sim \nu$. It is a *full congruence* if it satisfies

$$P_i \sim Q_i \text{ for all } i = 1, ..., n \quad \Rightarrow \quad f(P_1, ..., P_n) \sim f(Q_1, ..., Q_n) \tag{1}$$

$$\mathcal{S}_y \sim \mathcal{S}'_y \text{ for all } y \in V_{\mathcal{S}} \quad \Rightarrow \quad \langle x|\mathcal{S}\rangle \sim \langle x|\mathcal{S}'\rangle \tag{2}$$

for all functions $f$ of arity $n$, processes $P_i, Q_i \in \mathbb{P}$, and recursive specifications $\mathcal{S}, \mathcal{S}'$ with $x \in V_{\mathcal{S}} = V_{\mathcal{S}'}$ and $\langle x|\mathcal{S}\rangle, \langle x|\mathcal{S}'\rangle \in \mathbb{P}$.

Clearly, each full congruence is also a lean congruence, and each lean congruence satisfies (1) above. Both implications are strict, as illustrated in [16].

A main result of the present paper will be that strong reactive bisimilarity is a full congruence for the process algebra $\mathrm{CCSP}_{\mathrm{t}}^{\theta}$. To achieve it I need to establish first that strong bisimilarity [33], $\leftrightarroweq$, and initials equivalence [14, Section 16], $=_{\mathcal{I}}$, are full congruences for $\mathrm{CCSP}_{\mathrm{t}}^{\theta}$.

## 7.1 Initials equivalence

**Definition 17** Two $\mathrm{CCSP}_{\mathrm{t}}^{\theta}$ processes $P$ and $Q$ are *initials equivalent*, denoted $P =_{\mathcal{I}} Q$, if $\mathcal{I}(P) = \mathcal{I}(Q)$.

**Theorem 18** Initials equivalence is a full congruence for $\mathrm{CCSP}_{\mathrm{t}}^{\theta}$.

**Proof:** In Appendix A. □

## 7.2 Strong bisimilarity

**Definition 19** A *strong bisimulation* is a symmetric relation $\mathscr{B}$ on $\mathbb{P}$, such that, whenever $P \mathrel{\mathscr{B}} Q$,

- if $P \xrightarrow{\alpha} P'$ with $\alpha \in \mathit{Act}$ then $Q \xrightarrow{\alpha} Q'$ for some $Q'$ with $P' \mathrel{\mathscr{B}} Q'$.

Two processes $P, Q \in \mathbb{P}$ are *strongly bisimilar*, $P \leftrightarroweq Q$, if $P \mathrel{\mathscr{R}} Q$ for some strong bisimulation $\mathscr{B}$.

Contrary to reactive bisimilarity, strong bisimilarity treats the time-out action $\mathrm{t}$, as well as the hidden action $\tau$, just like any visible action. In the absence of time-out actions, there is no difference between a strong bisimulation and a time-out bisimulation, so $\leftrightarroweq_r$ and $\leftrightarroweq$ coincide. In general, strong bisimulation is a finer equivalence relation than strong reactive bisimilarity and initials equivalence: $P \leftrightarroweq Q \Rightarrow P \leftrightarroweq_r Q \Rightarrow P =_{\mathcal{I}} Q$, and both implications are strict.

**Lemma 1** For each $\mathrm{CCSP}_{\mathrm{t}}^{\theta}$ process $P$ there exists a $\mathrm{CCSP}_{\mathrm{t}}$ process $Q$ only built using inaction, action prefixing, choice and recursion, such that $P \leftrightarroweq Q$.

**Proof:** Immediately from Propositions 12 and 15. □

**Theorem 20** Strong bisimilarity is a full congruence for $\mathrm{CCSP}_\mathrm{t}^\theta$.

**Proof:** The structural operational rules for $\mathrm{CCSP}_\mathrm{t}$ (that is, $\mathrm{CCSP}_\mathrm{t}^\theta$ without the operators $\theta_L^U$ and $\psi_X$) fit the *tyft/tyxt format with recursion* of [16]. By [16, Theorem 3] this implies that $\leftrightarrow$ is a full congruence for $\mathrm{CCSP}_\mathrm{t}$. (In fact, when omitting the recursion construct, the operational rules for $\mathrm{CCSP}_\mathrm{t}$ fit the *tyft/tyxt format* of [26], and by the main theorem of [26], $\leftrightarrow$ is a congruence *for the operators of* $\mathrm{CCSP}_\mathrm{t}$, that is, it satisfies (1) in Definition 16. The work of [16] extends this result of [26] with recursion.)

The structural operational rules for all of $\mathrm{CCSP}_\mathrm{t}^\theta$ fit the *ntyft/ntyxt format with recursion* of [16]. By [16, Theorem 2] this implies that $\leftrightarrow$ is a lean congruence for $\mathrm{CCSP}_\mathrm{t}^\theta$. (In fact, when omitting the recursion construct, the operational rules for $\mathrm{CCSP}_\mathrm{t}^\theta$ fit the *ntyft/ntyxt format* of [25], and by the main theorem of [25], $\leftrightarrow$ is a congruence for the operators of $\mathrm{CCSP}_\mathrm{t}^\theta$. The work of [16] extends this result of [25] with recursion.)

To verify (2) for the whole language $\mathrm{CCSP}_\mathrm{t}^\theta$, let $\mathcal{S}$ and $\mathcal{S}'$ be recursive specifications with $x \in V_\mathcal{S} = V_{\mathcal{S}'}$, such that $\langle x|\mathcal{S}\rangle, \langle x|\mathcal{S}'\rangle \in \mathbb{P}$ and $\mathcal{S}_y \leftrightarrow \mathcal{S}'_y$ for all $y \in V_\mathcal{S}$. Let $\{P_i \mid i \in I\}$ be the collection of processes of the form $\theta_L^U(Q)$ or $\psi_X(Q)$, for some $L$, $U$, $X$, that occur as a closed subexpression of $\mathcal{S}_y$ or $\mathcal{S}'_y$ for one of the $y \in V_\mathcal{S}$, not counting strict subexpressions of a closed subexpression $R$ of $\mathcal{S}_y$ or $\mathcal{S}'_y$ that is itself of the form $\theta_L^U(Q)$ or $\psi_X(Q)$. Pick a fresh variable $z_i \notin V_\mathcal{S}$ for each $i \in I$, and let, for $y \in V_\mathcal{S}$, $\widehat{\mathcal{S}}_y$ be the result of replacing each occurrence of $P_i$ in $\mathcal{S}_y$ by $z_i$. Then $\widehat{\mathcal{S}}_y$ does not contain the operators $\theta_L^U(Q)$ or $\psi_X(Q)$. In deriving this conclusion it is essential that $\langle x|\mathcal{S}\rangle$ is a valid expression, for this implies that the term $\mathcal{S}_y \in \mathbb{E}$, which may contain free occurrences of the variables $y \in V_\mathcal{S}$, does not have a subterm of the form $\theta_L^U(F)$ or $\psi_X(F)$ that contains free occurrences of these variables. Let $\widehat{\mathcal{S}} := \{y = \widehat{\mathcal{S}}_y \mid y \in V_\mathcal{S}\}$; it is a recursive specification in the language $\mathrm{CCSP}_\mathrm{t}$. The recursive specification $\widehat{\mathcal{S}}'$ is defined in the same way.

For each $i \in I$ there is, by Lemma 1, a process $Q_i$ in the language $\mathrm{CCSP}_\mathrm{t}$ such that $P_i \leftrightarrow Q_i$. Now let $\rho, \eta : \{z_i \mid i \in I\} \rightarrow \mathbb{P}$ be the substitutions defined by $\rho(z_i) = P_i$ and $\eta(z_i) = Q_i$ for all $i \in I$. Then $\rho \leftrightarrow \eta$. Since $\leftrightarrow$ is a lean congruence for $\mathrm{CCSP}_\mathrm{t}^\theta$, one has $\langle x|\widehat{\mathcal{S}}\rangle[\rho] \leftrightarrow \langle x|\widehat{\mathcal{S}}\rangle[\eta]$ and likewise $\langle x|\widehat{\mathcal{S}}'\rangle[\rho] \leftrightarrow \langle x|\widehat{\mathcal{S}}'\rangle[\eta]$. For the same reason one has $\widehat{\mathcal{S}}_y[\eta] \leftrightarrow \widehat{\mathcal{S}}_y[\rho] = \mathcal{S}_y \leftrightarrow \mathcal{S}'_y \leftrightarrow \widehat{\mathcal{S}}'_y[\rho] \leftrightarrow \widehat{\mathcal{S}}'_y[\eta]$ for all $y \in V_\mathcal{S}$. Since $\widehat{\mathcal{S}}[\eta]$ and $\widehat{\mathcal{S}}'[\eta]$ are recursive specifications over $\mathrm{CCSP}_\mathrm{t}$, $\langle x|\widehat{\mathcal{S}}[\eta]\rangle \leftrightarrow \langle x|\widehat{\mathcal{S}}'[\eta]\rangle$. Hence $\langle x|\mathcal{S}\rangle = \langle x|\widehat{\mathcal{S}}[\rho]\rangle = \langle x|\widehat{\mathcal{S}}\rangle[\rho] \leftrightarrow \langle x|\widehat{\mathcal{S}}\rangle[\eta] = \langle x|\widehat{\mathcal{S}}[\eta]\rangle \leftrightarrow \langle x|\widehat{\mathcal{S}}'[\eta]\rangle \leftrightarrow \langle x|\widehat{\mathcal{S}}'[\rho]\rangle = \langle x|\mathcal{S}'\rangle$. □

The following lemmas on the relation between $\theta_X$ and the other operators of $\mathrm{CCSP}_\mathrm{t}^\theta$ deal with strong bisimilarity, but are needed in the congruence proof for strong reactive bisimilarity. Their proofs can be found in Appendix B.

**Lemma 2** If $P \stackrel{\tau}{\nrightarrow}$, $\mathcal{I}(P) \cap X \subseteq S$ and $Y = X \setminus (S \setminus \mathcal{I}(P))$, then $\theta_X(P \|_S Q) \leftrightarrow \theta_X(P \|_S \theta_Y(Q))$.

**Lemma 3** $\theta_X(\tau_I(P)) \leftrightarrow \theta_X(\tau_I(\theta_{X\cup I}(P)))$.

**Lemma 4** $\theta_X(\mathcal{R}(P)) \leftrightarrow \theta_X(\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P)))$.

## 8 Strong reactive bisimilarity is a full congruence for $\mathrm{CCSP}_\mathrm{t}^\theta$

The forthcoming proofs showing that $\leftrightarrow_r$ is a full congruence for $\mathrm{CCSP}_\mathrm{t}^\theta$ follow the lines of Milner [33], but are more complicated due to the nature of reactive bisimilarity. A crucial tool is Milner's notion of *bisimilarity up-to*. The above three lemmas play an essential rôle. Even if we would not be interested in the operators $\theta_L^U$ and $\psi_X$, the proof needs to take the operator $\theta_X$ ($= \theta_X^X$) along in order to deal with the other operators. This is a consequence of the occurrence of $\theta_X$ in Definition 8.

**Definition 21** Given a relation $\sim\ \subseteq \mathbb{P} \times \mathbb{P}$, a *strong time-out bisimulation up to* $\sim$ is a symmetric relation $\mathscr{B} \subseteq \mathbb{P} \times \mathbb{P}$, such that, for $P \mathrel{\mathscr{B}} Q$,

- if $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$, then $\exists Q'$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \sim\mathscr{B}\sim Q'$,
- if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$, then $\exists Q'$ with $Q \xrightarrow{\mathrm{t}} Q'$ and $\theta_X(P') \sim\mathscr{B}\sim \theta_X(Q')$.

Here $\sim\mathscr{B}\sim\ := \{(R,T) \mid \exists R', T'.\ R \sim R' \mathrel{\mathscr{B}} T' \sim T\}$.

**Proposition 22** If $P \mathrel{\mathscr{B}} Q$ for some strong time-out bisimulation $\mathscr{B}$ up to $\leftrightarroweq$, then $P \leftrightarroweq_r Q$.

**Proof:** Using the reflexivity of $\leftrightarroweq$ it suffices to show that $\leftrightarroweq\mathscr{B}\leftrightarroweq$ is a strong time-out bisimulation. Clearly this relation is symmetric, and that it satisfies the first clause of Definition 8 is straightforward, using transitivity of $\leftrightarroweq$. So assume $P \leftrightarroweq R \mathrel{\mathscr{B}} T \leftrightarroweq Q$, $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$. Then $\mathcal{I}(R) \cap (X \cup \{\tau\}) = \emptyset$. By the transfer property of $\leftrightarroweq$, there exists an $R'$ with $R \xrightarrow{\mathrm{t}} R'$ and $P' \leftrightarroweq R'$. Since $\leftrightarroweq$ is a congruence for $\theta_X$ it follows that $\theta_X(P') \leftrightarroweq \theta_X(R')$. By Definition 21, there exists a $T'$ with $T \xrightarrow{\mathrm{t}} T'$ and $\theta_X(R') \leftrightarroweq\mathscr{B}\leftrightarroweq \theta_X(T')$. Again using the transfer property of $\leftrightarroweq$, there exists a $Q'$ with $Q \xrightarrow{\mathrm{t}} Q'$ and $\theta_X(T') \leftrightarroweq \theta_X(Q')$. Thus, $\theta_X(P') \leftrightarroweq\mathscr{B}\leftrightarroweq \theta_X(Q')$. □

**Theorem 23** Strong reactive bisimilarity is a lean congruence for $\mathrm{CCSP}_{\mathrm{t}}^{\theta}$. In other words, if $\rho, \nu : \mathit{Var} \rightharpoonup \mathbb{E}$ are substitutions with $\rho \leftrightarroweq_r \nu$, then $E[\rho] \leftrightarroweq_r E[\nu]$ for any expression $E \in \mathbb{E}$.

**Proof:** It suffices to prove this theorem for the special case that $\rho, \nu : \mathit{Var} \to \mathbb{P}$ are closed substitutions; the general case then follows by means of composition of substitutions. Let $\mathscr{B} \subseteq \mathbb{P} \times \mathbb{P}$ be the smallest relation satisfying

- if $P \leftrightarroweq_r Q$, then $P \mathrel{\mathscr{B}} Q$,
- if $P \mathrel{\mathscr{B}} Q$ and $\alpha \in A \cup \{\tau, \mathrm{t}\}$, then $\alpha.P \mathrel{\mathscr{B}} \alpha.Q$,
- if $P_1 \mathrel{\mathscr{B}} Q_1$ and $P_2 \mathrel{\mathscr{B}} Q_2$, then $P_1 + P_2 \mathrel{\mathscr{B}} Q_1 + Q_2$,
- if $P \mathrel{\mathscr{B}} Q$, $L \subseteq U \subseteq A$ and $X \subseteq A$, then $\theta_L^U(P) \mathrel{\mathscr{B}} \theta_L^U(Q)$ and $\psi_X(P) \mathrel{\mathscr{B}} \psi_X(Q)$,
- if $P_1 \mathrel{\mathscr{B}} Q_1$, $P_2 \mathrel{\mathscr{B}} Q_2$ and $S \subseteq A$, then $P_1 \|_S P_2 \mathrel{\mathscr{B}} Q_1 \|_S Q_2$,
- if $P \mathrel{\mathscr{B}} Q$ and $I \subseteq A$, then $\tau_I(P) \mathrel{\mathscr{B}} \tau_I(Q)$,
- if $P \mathrel{\mathscr{B}} Q$ and $\mathcal{R} \subseteq A \times A$, then $\mathcal{R}(P) \mathrel{\mathscr{B}} \mathcal{R}(Q)$,
- if $\mathcal{S}$ is a recursive specification with $z \in V_{\mathcal{S}}$, and $\rho, \nu : \mathit{Var} \setminus V_{\mathcal{S}} \to \mathbb{P}$ are substitutions satisfying $\rho(x) \mathrel{\mathscr{B}} \nu(x)$ for all $x \in \mathit{Var} \setminus V_{\mathcal{S}}$, then $\langle z|\mathcal{S}\rangle[\rho] \mathrel{\mathscr{B}} \langle z|\mathcal{S}\rangle[\nu]$.

A straightforward induction on the derivation of $P \mathrel{\mathscr{B}} Q$, employing Theorem 18, yields that

$$\text{if } P \mathrel{\mathscr{B}} Q \text{ then } \mathcal{I}(P) = \mathcal{I}(Q)\text{, i.e., } P =_{\mathcal{I}} Q. \tag{@}$$

(For the last case, the assumption that $\rho(x) \mathrel{\mathscr{B}} \nu(x)$ for all $x \in \mathit{Var} \setminus V_{\mathcal{S}}$ implies $\rho =_{\mathcal{I}} \nu$ by induction. Since $=_{\mathcal{I}}$ is a lean congruence by Theorem 18, this implies $\langle z|\mathcal{S}\rangle[\rho] =_{\mathcal{I}} \langle z|\mathcal{S}\rangle[\nu]$.)
A trivial structural induction on $E \in \mathbb{E}$ shows that

$$\text{if } \rho, \nu : \mathit{Var} \to \mathbb{P} \text{ satisfy } \rho(x) \mathrel{\mathscr{B}} \nu(x) \text{ for all } x \in \mathit{Var}\text{, then } E[\rho] \mathrel{\mathscr{B}} E[\nu]. \tag{*}$$

For $\mathcal{S}$ a recursive specification and $\rho : \mathit{Var} \setminus V_{\mathcal{S}} \to \mathbb{P}$, let $\rho_{\mathcal{S}} : \mathit{Var} \to \mathbb{P}$ be the closed substitution given by $\rho_{\mathcal{S}}(x) := \langle x|\mathcal{S}\rangle[\rho]$ if $x \in V_{\mathcal{S}}$ and $\rho_{\mathcal{S}}(x) := \rho(x)$ otherwise. Then $\langle E|\mathcal{S}\rangle[\rho] = E[\rho_{\mathcal{S}}]$ for all $E \in \mathbb{E}$. Hence an application of (*) with $\rho_{\mathcal{S}}$ and $\nu_{\mathcal{S}}$ yields that under the conditions of the last clause for $\mathscr{B}$ above one even has $\langle E|\mathcal{S}\rangle[\rho] \mathrel{\mathscr{B}} \langle E|\mathcal{S}\rangle[\nu]$ for all expressions $E \in \mathbb{E}$. ($)
It suffices to show that $\mathscr{B}$ is a strong time-out bisimulation up to $\leftrightarroweq$, because then $P \leftrightarroweq_r Q \Leftrightarrow P \mathrel{\mathscr{B}} Q$, and (*) implies that $\mathscr{B}$ is a lean congruence. Because $\leftrightarroweq_r$ is symmetric, so is $\mathscr{B}$. So I need to show that $\mathscr{B}$ satisfies the two clauses of Definition 21.

- Let $P \mathscr{B} Q$ and $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$. I have to find a $Q'$ with $Q \xrightarrow{\alpha} Q'$ and $P' \leftrightarrow \mathscr{B} \leftrightarrow Q'$. In fact, I show that even $P' \mathscr{B} Q'$. This I will do by structural induction on the proof $\pi$ of $P \xrightarrow{\alpha} P'$ from the rules of Table 1. I make a case distinction based on the derivation of $P \mathscr{B} Q$.
  - Let $P \leftrightarrow_r Q$. Using that the relation $\leftrightarrow_r$ is a strong time-out bisimulation, there must be a process $Q'$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \leftrightarrow_r Q'$. Hence $P' \mathscr{B} Q'$.
  - Let $P = \beta.P^\dagger$ and $Q = \beta.Q^\dagger$ with $\beta \in A \cup \{\tau, \mathrm{t}\}$ and $P^\dagger \mathscr{B} Q^\dagger$. Then $\alpha = \beta$ and $P' = P^\dagger$. Take $Q' := Q^\dagger$. Then $Q \xrightarrow{\alpha} Q'$ and $P' \mathscr{B} Q'$.
  - Let $P = P_1 + P_2$ and $Q = Q_1 + Q_2$ with $P_1 \mathscr{B} Q_1$ and $P_2 \mathscr{B} Q_2$. I consider the first rule from Table 1 that could have been responsible for the derivation of $P \xrightarrow{\alpha} P'$; the other proceeds symmetrically. So suppose that $P_1 \xrightarrow{\alpha} P'$. Then by induction $Q_1 \xrightarrow{\alpha} Q'$ for some $Q'$ with $P' \mathscr{B} Q'$. By the same rule from Table 1, $Q \xrightarrow{\alpha} Q'$.
  - Let $P = \theta_L^U(P^\dagger)$, $Q = \theta_L^U(Q^\dagger)$ and $P^\dagger \mathscr{B} Q^\dagger$. First suppose $\alpha \in A$. Since $\theta_L^U(P^\dagger) \xrightarrow{\alpha} P'$, it must be that $P^\dagger \xrightarrow{\alpha} P'$ and either $\alpha \in U$ or $P^\dagger \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in L \cup \{\tau\}$. In the latter case, (@) yields $\mathcal{I}(P^\dagger) = \mathcal{I}(Q^\dagger)$, and thus $Q^\dagger \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in L \cup \{\tau\}$. By induction there exists a $Q'$ such that $Q^\dagger \xrightarrow{\alpha} Q'$ and $P' \mathscr{B} Q'$. So, in both cases, $Q = \theta_L^U(Q^\dagger) \xrightarrow{\alpha} Q'$.
    Now suppose $\alpha = \tau$. Since $\theta_L^U(P^\dagger) \xrightarrow{\tau} P'$ it must be that $P'$ has the form $\theta_L^U(P^\ddagger)$, and $P^\dagger \xrightarrow{\tau} P^\ddagger$. By induction, there exists a $Q^\ddagger$ such that $Q^\dagger \xrightarrow{\tau} Q^\ddagger$ and $P^\ddagger \mathscr{B} Q^\ddagger$. Now $Q = \theta_L^U(Q^\dagger) \xrightarrow{\tau} \theta_L^U(Q^\ddagger) =: Q'$ and $P' \mathscr{B} Q'$.
  - Let $P = \psi_X(P^\dagger)$, $Q = \psi_X(Q^\dagger)$ and $P^\dagger \mathscr{B} Q^\dagger$. Since $\psi_X(P^\dagger) \xrightarrow{\alpha} P'$, one has $P^\dagger \xrightarrow{\alpha} P'$. By induction there exists a $Q'$ with $Q^\dagger \xrightarrow{\alpha} Q'$ and $P' \mathscr{B} Q'$. So $Q = \psi_X(Q^\dagger) \xrightarrow{\alpha} Q'$.
  - Let $P = P_1 \|_S P_2$ and $Q = Q_1 \|_S Q_2$ with $P_1 \mathscr{B} Q_1$ and $P_2 \mathscr{B} Q_2$. I consider the three rules from Table 1 that could have been responsible for the derivation of $P \xrightarrow{\alpha} P'$.
    First suppose that $\alpha \notin S$, $P_1 \xrightarrow{\alpha} P_1'$ and $P' = P_1' \|_S P_2$. By induction, $Q_1 \xrightarrow{\alpha} Q_1'$ for some $Q_1'$ with $P_1' \mathscr{B} Q_1'$. Consequently, $Q_1 \|_S Q_2 \xrightarrow{\alpha} Q_1' \|_S Q_2$, and $P' = P_1' \|_S P_2 \mathscr{B} Q_1' \|_S Q_2$.
    Next suppose that $\alpha \in S$, $P_1 \xrightarrow{\alpha} P_1'$, $P_2 \xrightarrow{\alpha} P_2'$ and $P' = P_1' \|_S P_2'$. By induction, $Q_1 \xrightarrow{\alpha} Q_1'$ for some $Q_1'$ with $P_1' \mathscr{B} Q_1'$, and $Q_2 \xrightarrow{\alpha} Q_2'$ for some $Q_2'$ with $P_2' \mathscr{B} Q_2'$. Consequently, $Q_1 \|_S Q_2 \xrightarrow{\alpha} Q_1' \|_S Q_2'$, and $P' = P_1' \|_S P_2' \mathscr{B} Q_1' \|_S Q_2'$.
    The remaining case proceeds symmetrically to the first.
  - Let $P = \tau_I(P^\dagger)$ and $Q = \tau_I(Q^\dagger)$ with $I \subseteq A$ and $P^\dagger \mathscr{B} Q^\dagger$. Then $P^\dagger \xrightarrow{\beta} P^\ddagger$ for some $P^\ddagger$ with $P' = \tau_I(P^\ddagger)$, and either $\beta = \alpha \notin I$, or $\beta \in I$ and $\alpha = \tau$. By induction, $Q^\dagger \xrightarrow{\beta} Q^\ddagger$ for some $Q^\ddagger$ with $P^\ddagger \mathscr{B} Q^\ddagger$. Consequently, $Q = \tau_I(Q^\dagger) \xrightarrow{\alpha} \tau_I(Q^\ddagger) =: Q'$ and $P' \mathscr{B} Q'$.
  - Let $P = \mathcal{R}(P^\dagger)$ and $Q = \mathcal{R}(Q^\dagger)$ with $\mathcal{R} \subseteq A \times A$ and $P^\dagger \mathscr{B} Q^\dagger$. Then $P^\dagger \xrightarrow{\beta} P^\ddagger$ for some $P^\ddagger$ with $P' = \mathcal{R}(P^\ddagger)$, and either $(\beta, \alpha) \in \mathcal{R}$ or $\beta = \alpha = \tau$. By induction, $Q^\dagger \xrightarrow{\beta} Q^\ddagger$ for some $Q^\ddagger$ with $P^\ddagger \mathscr{B} Q^\ddagger$. Consequently, $Q = \mathcal{R}(Q^\dagger) \xrightarrow{\alpha} \mathcal{R}(Q^\ddagger) =: Q'$ and $P' \mathscr{B} Q'$.
  - Let $P = \langle z|\mathcal{S}\rangle[\rho] = \langle z|\mathcal{S}[\rho]\rangle$ and $Q = \langle z|\mathcal{S}\rangle[\nu] = \langle z|\mathcal{S}[\nu]\rangle$ where $\mathcal{S}$ is a recursive specification with $z \in V_\mathcal{S}$, and $\rho, \nu : \mathit{Var} \setminus V_\mathcal{S} \to \mathbb{P}$ satisfy $\rho(x) \mathscr{B} \nu(x)$ for all $x \in \mathit{Var} \setminus V_\mathcal{S}$. By Table 1 the transition $\langle \mathcal{S}_z|\mathcal{S}[\rho]\rangle \xrightarrow{\alpha} P'$ is provable by means of a strict subproof of the proof $\pi$ of $\langle z|\mathcal{S}\rangle[\rho] \xrightarrow{\alpha} P'$. By ($) above one has $\langle \mathcal{S}_z|\mathcal{S}[\rho]\rangle \mathscr{B} \langle \mathcal{S}_z|\mathcal{S}[\nu]\rangle$. So by induction there is a $Q'$ such that $\langle \mathcal{S}_z|\mathcal{S}[\nu]\rangle \xrightarrow{a} Q'$ and $P' \mathscr{B} Q'$. By Table 1, $Q = \langle z|\mathcal{S}[\nu]\rangle \xrightarrow{\alpha} Q'$.
- Let $P \mathscr{B} Q$, $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$. I have to find a $Q'$ such that $Q \xrightarrow{\mathrm{t}} Q'$ and $\theta_X(P') \leftrightarrow \mathscr{B} \leftrightarrow \theta_X(Q')$. This I will do by structural induction on the proof $\pi$ of $P \xrightarrow{\mathrm{t}} P'$ from the rules of Table 1. I make a case distinction based on the derivation of $P \mathscr{B} Q$.
  - Let $P \leftrightarrow_r Q$. Using that the relation $\leftrightarrow_r$ is a strong time-out bisimulation, there must be a process $Q'$ such that $Q \xrightarrow{\mathrm{t}} Q'$ and $\theta_X(P') \leftrightarrow_r \theta_X(Q')$. Thus $\theta_X(P') \leftrightarrow \mathscr{B} \leftrightarrow \theta_X(Q')$.

- Let $P = \beta.P^\dagger$ and $Q = \beta.Q^\dagger$ with $\beta \in A \cup \{\tau, \mathrm{t}\}$ and $P^\dagger \mathscr{B} Q^\dagger$. Then $\beta = \mathrm{t}$ and $P' = P^\dagger$. Take $Q' := Q^\dagger$. Then $Q \xrightarrow{\mathrm{t}} Q'$ and $P' \mathscr{B} Q'$. Thus $\theta_X(P') \leftrightarrow \mathscr{B} \leftrightarrow \theta_X(Q')$.
- Let $P = P_1 + P_2$ and $Q = Q_1 + Q_2$ with $P_1 \mathscr{B} Q_1$ and $P_2 \mathscr{B} Q_2$. I consider the first rule from Table 1 that could have been responsible for the derivation of $P \xrightarrow{\mathrm{t}} P'$; the other proceeds symmetrically. So suppose that $P_1 \xrightarrow{\mathrm{t}} P'$. Since $\mathcal{I}(P_1) \cap (X \cup \{\tau\}) \subseteq \mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$, by induction $Q_1 \xrightarrow{\mathrm{t}} Q'$ for some $Q'$ with $P' \leftrightarrow \mathscr{B} \leftrightarrow Q'$. Hence $Q \xrightarrow{\mathrm{t}} Q'$.
- Let $P = \theta_L^U(P^\dagger)$, $Q = \theta_L^U(Q^\dagger)$ and $P^\dagger \mathscr{B} Q^\dagger$. Since $\theta_L^U(P^\dagger) \xrightarrow{\mathrm{t}} P'$ it must be that $P^\dagger \xrightarrow{\mathrm{t}} P'$ and $P^\dagger \not\xrightarrow{\beta}$ for all $\beta \in L \cup \{\tau\}$. Consequently, $P \xrightarrow{\alpha} P^\ddagger$ iff $P^\dagger \xrightarrow{\alpha} P^\ddagger$, for all $\alpha \in A \cup \{\mathrm{t}\}$. So $\mathcal{I}(P^\dagger) \cap (X \cup \{\tau\}) = \emptyset$. By induction, $Q^\dagger \xrightarrow{\mathrm{t}} Q'$ for some $Q'$ with $\theta_X(P') \leftrightarrow \mathscr{B} \leftrightarrow \theta_X(Q')$. By (@), $Q^\dagger \not\xrightarrow{\beta}$ for all $\beta \in L \cup \{\tau\}$. Hence $Q = \theta_L^U(Q^\dagger) \xrightarrow{\mathrm{t}} Q'$.
- Let $P = \psi_Y(P^\dagger)$, $Q = \psi_Y(Q^\dagger)$ and $P^\dagger \mathscr{B} Q^\dagger$. Since $\psi_Y(P^\dagger) \xrightarrow{\mathrm{t}} P'$ one has $P^\dagger \xrightarrow{\mathrm{t}} P^\ddagger$ for some $P^\ddagger$ with $P' = \theta_Y(P^\ddagger)$, and $P^\dagger \not\xrightarrow{\beta}$ for all $\beta \in Y \cup \{\tau\}$, i.e., $\mathcal{I}(P^\dagger) \cap (Y \cup \{\tau\}) = \emptyset$. By induction, $Q^\dagger \xrightarrow{\mathrm{t}} Q^\ddagger$ for a $Q^\ddagger$ with $\theta_Y(P^\ddagger) \leftrightarrow \mathscr{B} \leftrightarrow \theta_Y(Q^\ddagger)$. By (@), $\mathcal{I}(P^\dagger) = \mathcal{I}(Q^\dagger)$, so $Q^\dagger \not\xrightarrow{\beta}$ for all $\beta \in Y \cup \{\tau\}$. Let $Q' := \theta_Y(Q^\ddagger)$, so that $Q = \psi_Y(Q^\dagger) \xrightarrow{\mathrm{t}} \theta_Y(Q^\ddagger) = Q'$. From $\theta_Y(P^\ddagger) \leftrightarrow P'' \mathscr{B} Q'' \leftrightarrow \theta_Y(Q^\ddagger)$ one obtains

$$\theta_X(\theta_Y(P^\ddagger)) \leftrightarrow \theta_X(P'') \mathscr{B} \theta_X(Q'') \leftrightarrow \theta_X(\theta_Y(Q^\ddagger)),$$

  using that $\leftrightarrow$ is a congruence for $\theta_X$ $(= \theta_X^X)$. Thus $\theta_X(P') \leftrightarrow \mathscr{B} \leftrightarrow \theta_X(Q')$.
- Let $P = P_1 \|_S P_2$ and $Q = Q_1 \|_S Q_2$ with $P_1 \mathscr{B} Q_1$ and $P_2 \mathscr{B} Q_2$. I consider the last rule from Table 1 that could have been responsible for the derivation of $P \xrightarrow{\mathrm{t}} P'$. The other proceeds symmetrically. So suppose that $P_2 \xrightarrow{\mathrm{t}} P_2'$ and $P' = P_1 \|_S P_2'$. Let $Y := X \setminus (S \setminus \mathcal{I}(P_1)) = (X \setminus S) \cup (X \cap S \cap \mathcal{I}(P_1))$. Then $\mathcal{I}(P_2) \cap (Y \cup \{\tau\}) = \emptyset$. By induction, $Q_2 \xrightarrow{\mathrm{t}} Q_2'$ for some $Q_2'$ with $\theta_Y(P_2') \leftrightarrow \mathscr{B} \leftrightarrow \theta_Y(Q_2')$. Let $Q' := Q_1 \|_S Q_2'$, so that $Q = Q_1 \|_S Q_2 \xrightarrow{\mathrm{t}} Q_1 \|_S Q_2' = Q'$. From $\theta_Y(P_2') \leftrightarrow P_2'' \mathscr{B} Q_2'' \leftrightarrow \theta_Y(Q_2')$ and $P_1 \mathscr{B} Q_1$ one obtains $P_1 \|_S \theta_Y(P_2') \leftrightarrow P_1 \|_S P_2'' \mathscr{B} Q_1 \|_S Q_2'' \leftrightarrow Q_1 \|_S \theta_Y(Q_2')$, using that $\leftrightarrow$ is a congruence for $\|_S$. Therefore, since $\leftrightarrow$ is also a congruence for $\theta_X$ $(= \theta_X^X)$,

$$\theta_X(P_1 \|_S \theta_Y(P_2')) \leftrightarrow \theta_X(P_1 \|_S P_2'') \mathscr{B} \theta_X(Q_1 \|_S Q_2'') \leftrightarrow \theta_X(Q_1 \|_S \theta_Y(Q_2')).$$

  Since $\mathcal{I}(P_1 \|_S P_2) \cap (X \cup \{\tau\}) = \emptyset$, one has $P_1 \not\xrightarrow{\tau}$ and $\mathcal{I}(P_1) \cap X \subseteq S$. Moreover, since $P_1 \mathscr{B} Q_1$, one has $\mathcal{I}(P_1) = \mathcal{I}(Q_1)$. Hence $\theta_X(P') = \theta_X(P_1 \|_S P_2') \leftrightarrow \theta_X(P_1 \|_S \theta_Y(P_2')) \leftrightarrow \mathscr{B} \leftrightarrow \theta_X(Q_1 \|_S \theta_Y(Q_2')) \leftrightarrow \theta_X(Q_1 \|_S Q_2') = \theta_X(Q')$ by Lemma 2.
- Let $P = \tau_I(P^\dagger)$ and $Q = \tau_I(Q^\dagger)$ with $I \subseteq A$ and $P^\dagger \mathscr{B} Q^\dagger$. Then $P^\dagger \xrightarrow{\mathrm{t}} P^\ddagger$ for some $P^\ddagger$ with $P' = \tau_I(P^\ddagger)$. Moreover, $\mathcal{I}(P^\dagger) \cap (X \cup I \cup \{\tau\}) = \emptyset$. By induction, $Q^\dagger \xrightarrow{\mathrm{t}} Q^\ddagger$ for some $Q^\ddagger$ with $\theta_{X\cup I}(P^\ddagger) \leftrightarrow \mathscr{B} \leftrightarrow \theta_{X\cup I}(Q^\ddagger)$. Let $Q' := \tau_I(Q^\ddagger)$, so that $Q = \tau_I(Q^\dagger) \xrightarrow{\mathrm{t}} \tau_I(Q^\ddagger) = Q'$. From $\theta_{X\cup I}(P^\ddagger) \leftrightarrow P'' \mathscr{B} Q'' \leftrightarrow \theta_{X\cup I}(Q^\ddagger)$ one obtains

$$\theta_X(\tau_I(P^\ddagger)) \leftrightarrow \theta_X(\tau_I(\theta_{X\cup I}(P^\ddagger))) \leftrightarrow \theta_X(\tau_I(P'')) \mathscr{B} \theta_X(\tau_I(Q'')) \leftrightarrow \ldots \leftrightarrow \theta_X(\tau_I(Q^\ddagger)),$$

  using Lemma 3 and that $\leftrightarrow$ is a congruence for $\tau_I$ and $\theta_X$. Thus $\theta_X(P') \leftrightarrow \mathscr{B} \leftrightarrow \theta_X(Q')$.
- Let $P = \mathcal{R}(P^\dagger)$ and $Q = \mathcal{R}(Q^\dagger)$ with $\mathcal{R} \subseteq A \times A$ and $P^\dagger \mathscr{B} Q^\dagger$. Then $P^\dagger \xrightarrow{\mathrm{t}} P^\ddagger$ for some $P^\ddagger$ with $P' = \mathcal{R}(P^\ddagger)$. Moreover, $\mathcal{I}(P^\dagger) \cap (\mathcal{R}^{-1}(X) \cup \{\tau\}) = \emptyset$. By induction, $Q^\dagger \xrightarrow{\mathrm{t}} Q^\ddagger$ for some $Q^\ddagger$ with $\theta_{\mathcal{R}^{-1}(X)}(P^\ddagger) \leftrightarrow \mathscr{B} \leftrightarrow \theta_{\mathcal{R}^{-1}(X)}(Q^\ddagger)$. Let $Q' := \mathcal{R}(Q^\ddagger)$, so that $Q = \mathcal{R}(Q^\dagger) \xrightarrow{\mathrm{t}} \mathcal{R}(Q^\ddagger) = Q'$. From $\theta_{\mathcal{R}^{-1}(X)}(P^\ddagger) \leftrightarrow P'' \mathscr{B} Q'' \leftrightarrow \theta_{\mathcal{R}^{-1}(X)}(Q^\ddagger)$ one obtains

$$\theta_X(\mathcal{R}(P^\ddagger)) \leftrightarrow \theta_X(\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P^\ddagger))) \leftrightarrow \theta_X(\mathcal{R}(P'')) \mathscr{B} \theta_X(\mathcal{R}(Q'')) \leftrightarrow \ldots \leftrightarrow \theta_X(\mathcal{R}(Q^\ddagger)),$$

  using Lemma 4 and that $\leftrightarrow$ is a congruence for $\mathcal{R}$ and $\theta_X$. Thus $\theta_X(P') \leftrightarrow \mathscr{B} \leftrightarrow \theta_X(Q')$.

– Let $P = \langle z|\mathcal{S}\rangle[\rho] = \langle z|\mathcal{S}[\rho]\rangle$ and $Q = \langle z|\mathcal{S}\rangle[\nu] = \langle z|\mathcal{S}[\nu]\rangle$ where $\mathcal{S}$ is a recursive specification with $z \in V_\mathcal{S}$, and $\rho, \nu : Var \setminus V_\mathcal{S} \to \mathbb{P}$ satisfy $\rho(x) \mathscr{B} \nu(x)$ for all $x \in Var \setminus V_\mathcal{S}$. By Table 1 the transition $\langle \mathcal{S}_z|\mathcal{S}[\rho]\rangle \xrightarrow{\mathrm{t}} P'$ is provable by means of a strict subproof of the proof $\pi$ of $\langle z|\mathcal{S}\rangle[\rho] \xrightarrow{\mathrm{t}} P'$. The rule for recursion in Table 1 also implies that $\mathcal{I}(\langle z|\mathcal{S}\rangle[\rho]) = \mathcal{I}(\langle \mathcal{S}_z|\mathcal{S}\rangle[\rho])$. Therefore, $\mathcal{I}(\langle \mathcal{S}_z|\mathcal{S}\rangle[\rho]) \cap (X \cup \{\tau\}) = \emptyset$. By ($) above one has $\langle \mathcal{S}_z|\mathcal{S}[\rho]\rangle \mathscr{B} \langle \mathcal{S}_z|\mathcal{S}[\nu]\rangle$. So by induction there is a $Q'$ such that $\langle \mathcal{S}_z|\mathcal{S}[\nu]\rangle \xrightarrow{\mathrm{t}} Q'$ and $\theta_X(P') \leftrightarrow \mathscr{B} \leftrightarrow \theta_X(Q')$. By Table 1, $Q = \langle z|\mathcal{S}[\nu]\rangle \xrightarrow{\mathrm{t}} Q'$. □

**Proposition 24** If $P \mathscr{B} Q$ for some strong time-out bisimulation $\mathscr{B}$ up to $\leftrightarrow_r$, then $P \leftrightarrow_r Q$.

**Proof:** Exactly as the proof of Proposition 22, now using that $\leftrightarrow_r$ is a congruence for $\theta_X$. □

**Theorem 25** Strong reactive bisimilarity is a full congruence for $\mathrm{CCSP}_\mathrm{t}^\theta$.

**Proof:** Let $\mathscr{B} \subseteq \mathbb{P} \times \mathbb{P}$ be the smallest relation satisfying

- if $\mathcal{S}$ and $\mathcal{S}'$ are recursive specifications with $x \in V_\mathcal{S} = V_{\mathcal{S}'}$ and $\langle x|\mathcal{S}\rangle, \langle x|\mathcal{S}'\rangle \in \mathbb{P}$, such that $\mathcal{S}_y \leftrightarrow \mathcal{S}'_y$ for all $y \in V_\mathcal{S}$, then $\langle x|\mathcal{S}\rangle \mathscr{B} \langle x|\mathcal{S}'\rangle$,

in addition to the eight or nine clauses listed in the proof of Theorem 23. Again, a straightforward induction on the derivation of $P \mathscr{B} Q$, employing Theorem 18, yields that

$$\text{if } P \mathscr{B} Q \text{ then } \mathcal{I}(P) = \mathcal{I}(Q), \text{ i.e., } P =_\mathcal{I} Q. \qquad (@)$$

(For the new case, the assumption that $\mathcal{S}_y \leftrightarrow \mathcal{S}'_y$ for all $y \in V_\mathcal{S}$ implies $\mathcal{S}_y =_\mathcal{I} \mathcal{S}'_y$ for all $y \in V_\mathcal{S}$. So by Theorem 18, $\langle x|\mathcal{S}\rangle =_\mathcal{I} \langle x|\mathcal{S}'\rangle$.) A trivial structural induction on $E \in \mathbb{E}$ shows again that

$$\text{if } \rho, \nu : Var \to \mathbb{P} \text{ satisfy } \rho(x) \mathscr{B} \nu(x) \text{ for all } x \in Var, \text{ then } E[\rho] \mathscr{B} E[\nu]. \qquad (*)$$

This again implies that in the last clause for $\mathscr{B}$ one even has $\langle E|\mathcal{S}\rangle[\rho] \mathscr{B} \langle E|\mathcal{S}'\rangle[\nu]$ for all $E \in \mathbb{E}$, ($)
and likewise, in the new clause, $\langle E|\mathcal{S}\rangle \mathscr{B} \langle E|\mathcal{S}'\rangle$ for all $E \in \mathbb{E}$ with variables from $V_\mathcal{S}$. (#)

It suffices to show that $\mathscr{B}$ is a strong time-out bisimulation up to $\leftrightarrow_r$, because then $\mathscr{B} \subseteq \leftrightarrow_r$ with Proposition 24, and the new clause for $\mathscr{B}$ implies (2). By construction $\mathscr{B}$ is symmetric.

- Let $P \mathscr{B} Q$ and $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$. I have to find a $Q'$ with $Q \xrightarrow{\alpha} Q'$ and $P' \leftrightarrow_r \mathscr{B} \leftrightarrow_r Q'$. In fact, I show that even $P' \mathscr{B} \leftrightarrow_r Q'$. This I will do by structural induction on the proof $\pi$ of $P \xrightarrow{\alpha} P'$ from the rules of Table 1. I make a case distinction based on the derivation of $P \mathscr{B} Q$.

  – Let $P = \langle x|\mathcal{S}\rangle \in \mathbb{P}$ and $Q = \langle x|\mathcal{S}'\rangle \in \mathbb{P}$ where $\mathcal{S}$ and $\mathcal{S}'$ are recursive specifications with $x \in V_\mathcal{S} = V_{\mathcal{S}'}$, such that $\mathcal{S}_y \leftrightarrow \mathcal{S}'_y$ for all $y \in V_\mathcal{S}$, meaning that for all $y \in W$ and $\sigma : V_\mathcal{S} \to \mathbb{P}$ one has $\mathcal{S}_y[\sigma] \leftrightarrow_r \mathcal{S}'_y[\sigma]$.
  By Table 1 the transition $\langle \mathcal{S}_x|\mathcal{S}\rangle \xrightarrow{\alpha} P'$ is provable by means of a strict subproof of $\pi$. By (#) above one has $\langle \mathcal{S}_x|\mathcal{S}\rangle \mathscr{B} \langle \mathcal{S}_x|\mathcal{S}'\rangle$. So by induction there is an $R' \in \mathbb{P}$ such that $\langle \mathcal{S}_x|\mathcal{S}'\rangle \xrightarrow{\alpha} R'$ and $P' \mathscr{B} \leftrightarrow_r R'$. Since $\langle \_|\mathcal{S}'\rangle$ is the application of a substitution of the form $\sigma : V_{\mathcal{S}'} \to \mathbb{P}$, one has $\langle \mathcal{S}_x|\mathcal{S}'\rangle \leftrightarrow_r \langle \mathcal{S}'_x|\mathcal{S}'\rangle$. Hence there is a $Q'$ with $P \vdash \langle \mathcal{S}'_x|\mathcal{S}'\rangle \xrightarrow{\alpha} Q'$ and $R' \leftrightarrow_r Q'$. So $P' \mathscr{B} \leftrightarrow_r Q'$. By Table 1, $Q = \langle x|\mathcal{S}'\rangle \xrightarrow{\alpha} Q'$.

  – The remaining nine cases proceed just as in the proof of Theorem 23, but with $\mathscr{B} \leftrightarrow_r$ substituted for the blue occurrences of $\mathscr{B}$. In the case for $\theta_L^U$ with $\alpha = \tau$, I conclude from $P^\ddagger \mathscr{B} \leftrightarrow_r Q^\ddagger$ that $\theta_L^U(P^\ddagger) \mathscr{B} \leftrightarrow_r \theta_L^U(Q^\ddagger)$. Besides applying the definition of $\mathscr{B}$, this also involves the application of Theorem 23 that $\leftrightarrow_r$ is already known to be a congruence for $\theta_L^U$. The same reasoning applies in the cases for $\|_S$, $\tau_I$ and $\mathcal{R}$.

- Let $P \mathscr{B} Q$, $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\mathrm{t}} P'$. I will find a $Q'$ such that $Q \xrightarrow{\mathrm{t}} Q'$ and $\theta_X(P') \mathrel{\underline{\leftrightarrow}} \mathscr{B} \mathrel{\underline{\leftrightarrow}}_r \theta_X(Q')$. This I will do by structural induction on the proof $\pi$ of $P \xrightarrow{\mathrm{t}} P'$ from the rules of Table 1. I make a case distinction based on the derivation of $P \mathscr{B} Q$.
  - Let $P = \langle x|\mathcal{S}\rangle \in \mathbb{P}$ and $Q = \langle x|\mathcal{S}'\rangle \in \mathbb{P}$ where $\mathcal{S}$ and $\mathcal{S}'$ are recursive specifications with $x \in V_\mathcal{S} = V_{\mathcal{S}'}$, such that for all $y \in W$ and $\sigma : V_\mathcal{S} \to \mathbb{P}$ one has $\mathcal{S}_y[\sigma] \mathrel{\underline{\leftrightarrow}}_r \mathcal{S}'_y[\sigma]$. By Table 1 the transition $\langle \mathcal{S}_x|\mathcal{S}\rangle \xrightarrow{\mathrm{t}} P'$ is provable by means of a strict subproof of the proof $\pi$ of $\langle x|\mathcal{S}\rangle \xrightarrow{\mathrm{t}} P'$. The rule for recursion in Table 1 also implies that $\mathcal{I}(\langle x|\mathcal{S}\rangle) = \mathcal{I}(\langle \mathcal{S}_x|\mathcal{S}\rangle)$. Therefore, $\mathcal{I}(\langle \mathcal{S}_x|\mathcal{S}\rangle) \cap (X \cup \{\tau\}) = \emptyset$. By (#) above one has $\langle \mathcal{S}_x|\mathcal{S}\rangle \mathscr{B} \langle \mathcal{S}_x|\mathcal{S}'\rangle$. So by induction there is an $R' \in \mathbb{P}$ such that $\langle \mathcal{S}_x|\mathcal{S}'\rangle \xrightarrow{\mathrm{t}} R'$ and $\theta_X(P') \mathrel{\underline{\leftrightarrow}} \mathscr{B} \mathrel{\underline{\leftrightarrow}}_r \theta_X(R')$. Since $\langle \_|\mathcal{S}'\rangle$ is the application of a substitution of the form $\sigma : V_{\mathcal{S}'} \to \mathbb{P}$, $\langle \mathcal{S}_x|\mathcal{S}'\rangle \mathrel{\underline{\leftrightarrow}}_r \langle \mathcal{S}'_x|\mathcal{S}'\rangle$. Using (@), $\mathcal{I}(\langle \mathcal{S}_x|\mathcal{S}'\rangle) \cap (X \cup \{\tau\}) = \emptyset$. Hence $\exists Q'$ with $P \vdash \langle \mathcal{S}'_x|\mathcal{S}'\rangle \xrightarrow{\mathrm{t}} Q'$ and $R' \mathrel{\underline{\leftrightarrow}}_r Q'$, and thus $\theta_X(R') \mathrel{\underline{\leftrightarrow}}_r \theta_X(Q')$, using Theorem 23. So $\theta_X(P') \mathrel{\underline{\leftrightarrow}} \mathscr{B} \mathrel{\underline{\leftrightarrow}}_r \theta_X(Q')$. By Table 1, $Q = \langle x|\mathcal{S}'\rangle \xrightarrow{\mathrm{t}} Q'$.
  - The remaining eight cases proceed just as in the proof of Theorem 23, but with $\mathscr{B} \mathrel{\underline{\leftrightarrow}}_r$ substituted for the blue occurrences of $\mathscr{B} \mathrel{\underline{\leftrightarrow}}$. □

# 9 The Recursive Specification Principle

For $W \subseteq \mathit{Var}$ a set of variables, a $W$-tuple of expressions is a function $\vec{E} \in \mathbb{E}^W$. It has a component $\vec{E}(x)$ for each variable $x \in W$. Note that a $W$-tuple of expressions is nothing else than a substitution. Let $\mathit{id}_W$ be the identity function, given by $\mathit{id}_W(x) = x$ for all $x \in W$. If $G \in \mathbb{E}$ and $\vec{E} \in \mathbb{E}^W$ then $G[\vec{E}]$ denotes the result of simultaneous substitution of $\vec{E}(x)$ for $x$ in $G$, for all $x \in W$. Likewise, if $\vec{G} \in \mathbb{E}^V$ and $\vec{E} \in \mathbb{E}^W$ then $\vec{G}[\vec{E}] \in \mathbb{E}^V$ denotes the $V$-tuple with components $G(y)[\vec{E}]$ for $y \in V$. Henceforth, I regard a recursive specification $\mathcal{S}$ as a $V_\mathcal{S}$-tuple with components $\mathcal{S}(y) = \mathcal{S}_y$ for $y \in V_\mathcal{S}$. If $\vec{E} \in \mathbb{E}^W$ and $\mathcal{S} \in \mathbb{E}^V$, then $\langle \vec{E}|\mathcal{S}\rangle \in \mathbb{E}^W$ is the $W$-tuple with components $\langle \vec{E}(x)|\mathcal{S}\rangle \in \mathbb{E}^W$ for $x \in W$.

For $\mathcal{S}$ a recursive specification and $\vec{E} \in \mathbb{E}^{V_\mathcal{S}}$ a $V_\mathcal{S}$-tuple of expressions, $\vec{E} \mathrel{\underline{\leftrightarrow}}_r \mathcal{S}[\vec{E}]$ states that $\vec{E}$ is a *solution* of $\mathcal{S}$, up to strong reactive bisimilarity. The tuple $\langle \mathit{id}_{V_\mathcal{S}}|\mathcal{S}\rangle \in \mathbb{E}^{V_\mathcal{S}}$ is called the *default solution*.

In [2, 10] two requirements occur for process algebras with recursion. The *recursive definition principle* (RDP) says that each recursive specification must have a solution, and the *recursive specification principle* (RSP) says that guarded recursive specifications have at most one solution. When dealing with process algebras where the meaning of a closed expression is a semantic equivalence class of processes, these principles become requirements on the semantic equivalence employed.

**Proposition 26** Let $\mathcal{S}$ be a recursive specification, and $x \in V_\mathcal{S}$. Then $\langle x|\mathcal{S}\rangle \mathrel{\underline{\leftrightarrow}}_r \langle \mathcal{S}_x|\mathcal{S}\rangle$.

**Proof:** Let $\sigma : \mathit{Var} \to \mathbb{P}$ be a closed substitution. I have to show that $\langle x|\mathcal{S}\rangle[\sigma] \mathrel{\underline{\leftrightarrow}}_r \langle \mathcal{S}_x|\mathcal{S}\rangle[\sigma]$. Equivalently I may show this for $\sigma : \mathit{Var}\backslash V_\mathcal{S} \to \mathbb{P}$. Now $\langle x|\mathcal{S}\rangle[\sigma] = \langle x|\mathcal{S}[\sigma]\rangle \in \mathbb{P}$ and $\langle \mathcal{S}_x|\mathcal{S}\rangle[\sigma] = \langle \mathcal{S}_x[\sigma]|\mathcal{S}[\sigma]\rangle \in \mathbb{P}$. Consequently, it suffices to prove the proposition under the assumption that $\langle x|\mathcal{S}\rangle, \langle \mathcal{S}_x|\mathcal{S}\rangle \in \mathbb{P}$. This follows immediately from the rule for recursion in Table 1 and Definition 8. □

Proposition 26 says that the recursive definition principle holds for strong reactive bisimulation semantics. The "default solution" of a recursive specification is in fact a solution. Note that the conclusion of Proposition 26 can be restated as $\langle \mathit{id}_{V_\mathcal{S}}|\mathcal{S}\rangle \mathrel{\underline{\leftrightarrow}}_r \langle \mathcal{S}|\mathcal{S}\rangle$, and that $\mathcal{S}[\langle \mathit{id}_{V_\mathcal{S}}|\mathcal{S}\rangle] = \langle \mathcal{S}|\mathcal{S}\rangle$.

The following theorem establishes the recursive specification principle for strong reactive bisimulation semantics. Some aspects of the proof that are independent of the notion of bisimilarity employed are delegated to the following two lemmas.

**Lemma 5** Let $H \in \mathbb{E}$ be guarded and have free variables from $W \subseteq \mathit{Var}$ only, and let $\vec{P},\vec{Q} \in \mathbb{P}^W$. Then $\mathcal{I}(H[\vec{P}]) = \mathcal{I}(H[\vec{Q}])$.

**Proof:** In Appendix A. □

**Lemma 6** Let $H \in \mathbb{E}$ be guarded and have free variables from $W \subseteq \mathit{Var}$ only, and let $\vec{P},\vec{Q} \in \mathbb{P}^W$. If $H[\vec{P}] \stackrel{\alpha}{\longrightarrow} R'$ with $\alpha \in \mathit{Act}$, then $R'$ has the form $H'[\vec{P}]$ for some term $H' \in \mathbb{E}$ with free variables in $W$ only. Moreover $H[\vec{Q}] \stackrel{\alpha}{\longrightarrow} H'[\vec{Q}]$.

**Proof:** By induction on the derivation of $H[\vec{P}] \stackrel{\alpha}{\longrightarrow} R'$, making a case distinction on the shape of $H$.

Let $H = \alpha.G$, so that $H[\vec{P}] = \alpha.G[\vec{P}]$. Then $R' = G[\vec{P}]$ and $H[\vec{Q}] \stackrel{\alpha}{\longrightarrow} G[\vec{Q}]$.

The case $H = 0$ cannot occur. Nor can the case $H = x \in \mathit{Var}$, as $H$ is guarded.

Let $H = H_1 \|_S H_2$, so that $H[\vec{P}] = H_1[\vec{P}] \|_S H_2[\vec{P}]$. Note that $H_1$ and $H_2$ are guarded and have free variables in $W$ only. One possibility is that $a \notin S$, $H_1[\vec{P}] \stackrel{\alpha}{\longrightarrow} R_1$ and $R' = R_1 \|_S H_2[\vec{P}]$. By induction, $R_1'$ has the form $H_1'[\vec{P}]$ for some term $H_1' \in \mathbb{E}$ with free variables in $W$ only. Moreover, $H_1[\vec{Q}] \stackrel{\alpha}{\longrightarrow} H_1'[\vec{Q}]$. Thus $R' = (H_1' \|_S H_2)[\vec{P}]$, and $H' := H_1' \|_S H_2$ has free variables in $W$ only. Moreover, $H[\vec{Q}] = H_1[\vec{Q}] \|_S H_2[\vec{Q}] \stackrel{\alpha}{\longrightarrow} H_1'[\vec{Q}] \|_S H_2[\vec{Q}] = H'[\vec{Q}]$.

The other two cases for $\|_S$, and the cases for the operators $+$, $\tau_I$ and $\mathcal{R}$, are equally trivial.

Let $H = \theta_L^U(H^\dagger)$, so that $H[\vec{P}] = \theta_L^U(H^\dagger[\vec{P}])$. Note that $H^\dagger$ is guarded and has free variables in $W$ only. The case $\alpha = \tau$ is again trivial, so assume $\alpha \neq \tau$. Then $H^\dagger[\vec{P}] \stackrel{\alpha}{\longrightarrow} R'$ and either $\alpha \in X$ or $H^\dagger[\vec{P}] \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in L \cup \{\tau\}$. By induction, $R'$ has the form $H'[\vec{P}]$ for some term $H' \in \mathbb{E}$ with free variables in $W$ only. Moreover, $H^\dagger[\vec{Q}] \stackrel{\alpha}{\longrightarrow} H'[\vec{Q}]$. Since $\mathcal{I}(H^\dagger[\vec{P}]) = \mathcal{I}(H^\dagger[\vec{Q}])$ by Lemma 5, either $\alpha \in X$ or $H^\dagger[\vec{Q}] \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in L \cup \{\tau\}$. Consequently, $H[\vec{Q}] = \theta_L^U(H^\dagger[\vec{Q}]) \stackrel{\alpha}{\longrightarrow} H'[\vec{Q}]$.

Let $H = \psi_X(H^\dagger)$, so that $H[\vec{P}] = \psi_X(H^\dagger[\vec{P}])$. Note that $H^\dagger$ is guarded and has free variables in $W$ only. The case $\alpha \in A \cup \{\tau\}$ is trivial, so assume $\alpha = \mathrm{t}$. Then $H^\dagger[\vec{P}] \stackrel{\mathrm{t}}{\longrightarrow} R^\dagger$ for some $R^\dagger$ such that $R' = \theta_X(R^\dagger)$. Moreover, $H^\dagger[\vec{P}] \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. By induction, $R^\dagger$ has the form $H'[\vec{P}]$ for some term $H' \in \mathbb{E}$ with free variables in $W$ only. Moreover, $H^\dagger[\vec{Q}] \stackrel{\mathrm{t}}{\longrightarrow} H'[\vec{Q}]$. Since $\mathcal{I}(H^\dagger[\vec{P}]) = \mathcal{I}(H^\dagger[\vec{Q}])$ by Lemma 5, $H^\dagger[\vec{Q}] \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Consequently, $H[\vec{Q}] = \psi_X(H^\dagger[\vec{Q}]) \stackrel{\mathrm{t}}{\longrightarrow} \theta_X(H'[\vec{Q}])$.

Finally, let $H = \langle x|\mathcal{S}\rangle$, so that $H[\vec{P}] = \langle x|\mathcal{S}[\vec{P}^\dagger]\rangle$, where $\vec{P}^\dagger$ is the $W\backslash V_\mathcal{S}$-tuple that is left of $\vec{P}$ after deleting the $y$-components, for $y \in V_\mathcal{S}$. The transition $\langle \mathcal{S}_x[\vec{P}^\dagger]|\mathcal{S}[\vec{P}^\dagger]\rangle \stackrel{\alpha}{\longrightarrow} R'$ is derivable through a subderivation of the one for $\langle x|\mathcal{S}[\vec{P}^\dagger]\rangle \stackrel{\alpha}{\longrightarrow} R'$. Moreover, $\langle \mathcal{S}_x[\vec{P}^\dagger]|\mathcal{S}[\vec{P}^\dagger]\rangle = \langle \mathcal{S}_x|\mathcal{S}\rangle[\vec{P}]$. So by induction, $R'$ has the form $H'[\vec{P}]$ for some $H' \in \mathbb{E}$ with free variables in $W$ only, and $\langle \mathcal{S}_x|\mathcal{S}\rangle[\vec{Q}] \stackrel{\alpha}{\longrightarrow} H'[\vec{Q}]$. Since $\langle \mathcal{S}_x|\mathcal{S}\rangle[\vec{Q}] = \langle \mathcal{S}_x[\vec{Q}^\dagger]|\mathcal{S}[\vec{Q}^\dagger]\rangle$, it follows that $H[\vec{Q}] = \langle x|\mathcal{S}\rangle[\vec{Q}] = \langle x|\mathcal{S}[\vec{Q}^\dagger]\rangle \stackrel{\alpha}{\longrightarrow} H'[\vec{Q}]$. □

**Theorem 27** Let $\mathcal{S}$ be a guarded recursive specification. If $\vec{E} \leftrightarroweq_r \mathcal{S}[\vec{E}]$ and $\vec{F} \leftrightarroweq_r \mathcal{S}[\vec{F}]$ with $\vec{E},\vec{F} \in \mathbb{E}^{V_\mathcal{S}}$, then $\vec{E} \leftrightarroweq_r \vec{F}$.

**Proof:** It suffices to prove Theorem 27 under the assumptions that $\vec{E},\vec{F} \in \mathbb{P}^{V_\mathcal{S}}$ and only the variables from $V_\mathcal{S}$ occur free in the expressions $\mathcal{S}_x$ for $x \in V_\mathcal{S}$. For in the general case I have to establish that $\vec{E}[\sigma] \leftrightarroweq_r \vec{F}[\sigma]$ for an arbitrary closed substitution $\sigma : \mathit{Var} \rightarrow \mathbb{P}$. Let $\hat{\sigma} : \mathit{Var}\backslash V_\mathcal{S} \rightarrow \mathbb{P}$ be given by $\hat{\sigma}(x) = \sigma(x)$ for all $x \in \mathit{Var}\backslash V_\mathcal{S}$. Then $\vec{E} \leftrightarroweq_r \mathcal{S}[\vec{E}]$ implies $\vec{E}[\sigma] \leftrightarroweq_r \mathcal{S}[\vec{E}][\sigma] = \mathcal{S}[\hat{\sigma}][\vec{E}[\sigma]]$. Hence, I merely have to prove the theorem with $\vec{E}[\sigma]$, $\vec{F}[\sigma]$and $\mathcal{S}[\hat{\sigma}]$ in place of $\vec{E}$, $\vec{F}$ and $\mathcal{S}$.

It also suffices to prove Theorem 27 under the assumption that $\mathcal{S}$ is a manifestly guarded recursive specification. Namely, for a general guarded recursive specification $\mathcal{S}$, let $\mathcal{S}'$ be the manifestly guarded specification into which $\mathcal{S}$ can be converted. Then $\vec{E} \leftrightarroweq_r \mathcal{S}[\vec{E}]$ implies $\vec{E} \leftrightarroweq_r \mathcal{S}'[\vec{E}]$ by Theorem 23.

So let $\mathcal{S}$ be manifestly guarded with free variables from $V_\mathcal{S}$ only, and let $\vec{P},\vec{Q} \in \mathbb{P}^{V_\mathcal{S}}$ be two of its solutions, that is, $\vec{P} \leftrightarroweq_r \mathcal{S}[\vec{P}]$ and $\vec{Q} \leftrightarroweq_r \mathcal{S}[\vec{Q}]$. I will show that the symmetric closure of

$$\mathscr{B} := \{H[\mathcal{S}[\vec{P}]], H[\mathcal{S}[\vec{Q}]] \mid H \in \mathbb{E} \text{ has free variables in } V_\mathcal{S} \text{ only}\}$$

is a strong time-out bisimulation up to $\leftrightarrow_r$. Once I have that, taking $H := x \in V_{\mathcal{S}}$ yields $\mathcal{S}_x[\vec{P}] \leftrightarrow_r \mathcal{S}_x[\vec{Q}]$ by Proposition 24, and thus $P(x) \leftrightarrow_r \mathcal{S}_x[\vec{P}] \leftrightarrow_r \mathcal{S}_x[\vec{Q}] \leftrightarrow_r Q(x)$ for all $x \in V_{\mathcal{S}}$. So $\vec{P} \leftrightarrow_r \vec{Q}$.

- Let $R \mathscr{B} T$ and $R \xrightarrow{\alpha} R'$ with $\alpha \in A \cup \{\tau\}$. I have to find a $T'$ with $T \xrightarrow{\alpha} T'$ and $P' \leftrightarrow_r \mathscr{B} \leftrightarrow_r Q'$. Assume that $R = H[\mathcal{S}[\vec{P}]]$ and $T = H[\mathcal{S}[\vec{Q}]]$—the case that $R = H[\mathcal{S}[\vec{Q}]]$ will follow by symmetry. Note that $H[\mathcal{S}[\vec{P}]]$ can also be written as $H[\mathcal{S}][\vec{P}]$. Since the expressions $\mathcal{S}_x$ for $x \in V_{\mathcal{S}}$ have free variables from $V_{\mathcal{S}}$ only, so does $H[\mathcal{S}]$. Moreover, since $\mathcal{S}$ is manifestly guarded, the expression $H[\mathcal{S}]$ must be guarded. By Lemma 6, $R'$ must have the form $H'[\vec{P}]$, where $H' \in \mathbb{E}$ has free variables in $V_{\mathcal{S}}$ only. Moreover, $T = H[\mathcal{S}[\vec{Q}]] = H[\mathcal{S}][\vec{Q}] \xrightarrow{\alpha} H'[\vec{Q}] =: T'$. Furthermore, by Theorem 23, $H'[\vec{P}] \leftrightarrow_r H'[\mathcal{S}[\vec{P}]]$ and $H'[\mathcal{S}[\vec{Q}]] \leftrightarrow_r H'[\vec{Q}]$. Thus, $R' = H'[\vec{P}] \leftrightarrow_r \mathscr{B} \leftrightarrow_r H'[\vec{Q}] = T'$.
- Let $R \mathscr{B} T$, $\mathcal{I}(R) \cap (X \cup \{\tau\}) = \emptyset$ and $R \xrightarrow{\mathrm{t}} R'$. I have to find a $T'$ such that $T \xrightarrow{\mathrm{t}} T'$ and $\theta_X(R') \leftrightarrow \mathscr{B} \leftrightarrow_r \theta_X(T')$. The proof for this case proceeds exactly as that of the previous case, up to the last sentence; the condition $\mathcal{I}(R) \cap (X \cup \{\tau\}) = \emptyset$ is not even used. Now from $R' = H'[\vec{P}] \leftrightarrow_r H'[\mathcal{S}[\vec{P}]] \mathscr{B} H'[\mathcal{S}[\vec{Q}]] \leftrightarrow_r H'[\vec{Q}] = T'$ it follows that
$$\theta_X(R') \leftrightarrow_r \theta_X(H'[\mathcal{S}][\vec{P}]) \mathscr{B} \theta_X(H'[\mathcal{S}][\vec{Q}]) \leftrightarrow_r \theta_X(T')$$
using Theorem 23 and the observation that $\theta_X(H'[\mathcal{S}[\vec{P}]]) = \theta_X(H')[\mathcal{S}[\vec{P}]]$. □

# 10 Complete axiomatisations

Let *Ax* denote the collection of axioms from Tables 2, 3 and 4, $Ax'$ the ones from Tables 2 and 3, and $Ax''$ merely the ones from Table 2. Moreover, let $Ax_f$, resp. $Ax'_f$ and $Ax''_f$, be same collections without the two axioms using the recursion construct $\langle x|\mathcal{S}\rangle$, RDP and RSP. In this section I establish the following.

Let $P$ and $Q$ be recursion-free $\mathrm{CCSP_t}$ processes. Then $P \leftrightarrow Q \;\Leftrightarrow\; Ax''_f \vdash P = Q.$ (3)

Let $P$ and $Q$ be $\mathrm{CCSP_t}$ processes with guarded recursion. Then $P \leftrightarrow Q \;\Leftrightarrow\; Ax'' \vdash P = Q.$ (4)

Let $P$ and $Q$ be recursion-free $\mathrm{CCSP_t^\theta}$ processes. Then $P \leftrightarrow Q \;\Leftrightarrow\; Ax'_f \vdash P = Q.$ (5)

Let $P$ and $Q$ be $\mathrm{CCSP_t^\theta}$ processes with guarded recursion. Then $P \leftrightarrow Q \;\Leftrightarrow\; Ax' \vdash P = Q.$ (6)

Let $P$ and $Q$ be recursion-free $\mathrm{CCSP_t^\theta}$ processes. Then $P \leftrightarrow_r Q \;\Leftrightarrow\; Ax_f \vdash P = Q.$ (7)

Let $P$ and $Q$ be $\mathrm{CCSP_t^\theta}$ processes with guarded recursion. Then $P \leftrightarrow_r Q \;\Leftrightarrow\; Ax \vdash P = Q.$ (8)

In each of these cases "⇐" states the soundness of the axiomatisation and "⇒" completeness.

Section 10.1 recalls (4), which stems from [22], and (3), which is folklore. Then Section 10.2 extends the existing proofs of (4) and (3) to obtain (6) and (5). In Section 10.3 I move from strong bisimilarity to strong reactive bisimilarity; I discuss the merits of the axiom RA from Table 4, and establish its soundness, thereby obtaining direction "⇐" of (8) and (7). I prove the completeness of $Ax_f$ for recursion-free processes—direction "⇒" of (7)—in Section 10.4. Sections 10.5–10.7 deal with the completeness of *Ax* for guarded $\mathrm{CCSP_t^\theta}$—direction "⇒" of (8). Section 10.8 explains why I need the axiom of choice for the latter result.

## 10.1 A complete axiomatisation of strong bisimilarity on guarded $\mathrm{CCSP_t}$

The well-known axioms of Table 2 are *sound* for strong bisimilarity, meaning that writing $\leftrightarrow$ for $=$, and substituting arbitrary expressions for the free variables $x, y, z$, or the meta-variables $P_i$ and $Q_j$, turns them into true statements. In these axioms $\alpha, \beta$ range over $Act$ and $a, b$ over $A$. All axioms involving

Table 2: A complete axiomatisation of strong bisimilarity on guarded $\mathrm{CCSP_t}$

$$
\begin{array}{rclrcllrcl}
x+(y+z) &=& (x+y)+z & \tau_I(x+y) &=& \tau_I(x)+\tau_I(y) & & \mathcal{R}(x+y) &=& \mathcal{R}(x)+\mathcal{R}(y)\\
x+y &=& y+x & \tau_I(\alpha.x) &=& \alpha.\tau_I(x) & \text{if } \alpha\notin I & \mathcal{R}(\tau.x) &=& \tau.\mathcal{R}(x)\\
x+x &=& x & \tau_I(\alpha.x) &=& \tau.\tau_I(x) & \text{if } \alpha\in I & \mathcal{R}(\mathrm{t}.x) &=& \mathrm{t}.\mathcal{R}(x)\\
x+0 &=& 0 & \langle x|\mathcal{S}\rangle &=& \langle \mathcal{S}_x|\mathcal{S}\rangle & \text{(RDP)} & \mathcal{R}(a.x) &=& \sum_{\{b|(a,b)\in\mathcal{R}\}} b.\mathcal{R}(x)
\end{array}
$$

If $P=\sum_{i\in I}\alpha_i.P_i$ and $Q=\sum_{j\in J}\beta_j.Q_j$ then

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

Recursive Specification Principle (RSP) $\qquad \mathcal{S}\Rightarrow x=\langle x|\mathcal{S}\rangle \qquad$ ($\mathcal{S}$ guarded)

variables are equations. The axiom involving $P$ and $Q$ is a template that stands for a family of equations, one for each fitting choice of $P$ and $Q$. This is the $\mathrm{CCSP_t}$ version of the *expansion law* from [33]. The axiom RDP ($\langle x|\mathcal{S}\rangle = \langle \mathcal{S}_x|\mathcal{S}\rangle$) says that recursively defined processes $\langle x|\mathcal{S}\rangle$ satisfy their set of defining equations $\mathcal{S}$. As discussed in the previous section, this entails that each recursive specification has a solution. The axiom RSP [2, 10] is a conditional equation with the equations of a guarded recursive specification $\mathcal{S}$ as antecedents. It says that the $x$-component of any solution of $\mathcal{S}$—a vector of processes substituted for the variables $V_\mathcal{S}$—equals $\langle x|\mathcal{S}\rangle$. In other words, each solution of $\mathcal{S}$ equals the default solution. This is a compact way of saying that solutions of guarded recursive specifications are unique.

**Theorem 28** For $\mathrm{CCSP_t}$ processes $P,Q\in\mathbb{P}$ with guarded recursion, one has $P \mathrel{\underline{\leftrightarrow}} Q$, that is, $P$ and $Q$ are strongly bisimilar, iff $P=Q$ is derivable from the axioms of Table 2.

In this theorem, "if", the *soundness* of the axiomatisation of Table 2, is an immediate consequence of the soundness of the individual axioms. "Only if" states the *completeness* of the axiomatisation.

A crucial tool in its proof is the simple observation that the axioms from the first box of Table 2 allow any $\mathrm{CCSP_t}$ process with guarded recursion to be brought in the form $\sum_{i\in I}\alpha_i.P_i$—a *head normal form*. Using this, the rest of the proof is a standard argument employing RSP, independent of the choice of the specific process algebra. It can be found in [31, 33], [2], [10] and many other places. However, in the literature this completeness theorem was always stated and proved for a small fragment of the process algebra, allowing only guarded recursive specifications with a finite number of equations, and whose right-hand sides $\mathcal{S}_y$ involve only the basic operators inaction, action prefixing and choice. Since the set of true statements $P \mathrel{\underline{\leftrightarrow}} Q$, with $P$ and $Q$ processes in a process algebra like guarded $\mathrm{CCSP_t}$, is well-known to be undecidable, and even not recursively enumerable, it was widely believed that no sound and complete finitely presented axiomatisation of strong bisimilarity could exist. Only in March 2017, Kees Middelburg observed (in the setting of the process algebra ACP [2, 10]) that the standard proof applies almost verbatim to arbitrary processes with guarded recursion, although one has to be a bit careful in dealing with the infinite nature of recursive specifications. The argument has been carefully documented in [22], in the setting of the process algebra ACP. This result does not contradict the non-enumerability of the set of true statements $P \mathrel{\underline{\leftrightarrow}} Q$, due to the fact that RSP is a proof rule with infinitely many premises.

A well-known simplification of Theorem 28 and its proof also yields completeness without recursion:

**Theorem 29** For $\mathrm{CCSP_t}$ processes $P,Q\in\mathbb{P}$ without recursion, one has $P \mathrel{\underline{\leftrightarrow}} Q$ iff $P=Q$ is derivable from the axioms of Table 2 minus RDP and RSP.

Table 3: A complete axiomatisation of strong bisimilarity on guarded CCSP$_{\mathrm{t}}^{\theta}$

| | |
|---|---|
| $\theta_L^U(\sum_{i\in I}\alpha_i.x_i) = \sum_{i\in I}\alpha_i.x_i$ | $(\alpha_i \notin L\cup\{\tau\}$ for all $i\in I)$ |
| $\theta_L^U(x+\alpha.y+\beta.z) = \theta_L^U(x+\alpha.y)$ | $(\alpha\in L\cup\{\tau\}\wedge\beta\notin U\cup\{\tau\})$ |
| $\theta_L^U(x+\alpha.y+\beta.z) = \theta_L^U(x+\alpha.y)+\theta_L^U(\beta.z)$ | $(\alpha\in L\cup\{\tau\}\wedge\beta\in U\cup\{\tau\})$ |
| $\theta_L^U(\beta.x) = \beta.x$ | $(\beta\neq\tau)$ |
| $\theta_L^U(\tau.x) = \tau.\theta_L^U(x)$ | |
| $\psi_X(x+\alpha.z) = \psi_X(x)+\alpha.z$ | $(\alpha\notin X\cup\{\tau,\mathrm{t}\})$ |
| $\psi_X(x+\alpha.y+\mathrm{t}.z) = \psi_X(x+\alpha.y)$ | $(\alpha\in X\cup\{\tau\})$ |
| $\psi_X(x+\alpha.y+\beta.z) = \psi_X(x+\alpha.y)+\beta.z$ | $(\alpha,\beta\in X\cup\{\tau\})$ |
| $\psi_X(\alpha.x) = \alpha.x$ | $(\alpha\neq\mathrm{t})$ |
| $\psi_X(\sum_{j\in I}\mathrm{t}.y_i) = \sum_{j\in I}\mathrm{t}.\theta_X(y_j)$ | |

## 10.2 A complete axiomatisation of strong bisimilarity on guarded CCSP$_{\mathrm{t}}^{\theta}$

Table 3 extends Table 2 with axioms for the auxiliary operators $\theta_L^U$ and $\psi_X$. With Table 1 it is straightforward to check the soundness of these axioms. The fourth axiom, for instance, follows from the second or third rule for $\theta_L^U$ in Table 1, depending on whether $\beta\in L\cup\{\mathrm{t}\}$. Moreover, a straightforward induction shows that these axioms suffice to convert each CCSP$_{\mathrm{t}}^{\theta}$ process with guarded recursion into the form $\sum_{I\in I}\alpha_i.P_i$—a head normal form. The below proposition sharpens this observation by pointing out that one can take the processes $P_i$ for $i\in I$ to be exactly the ones that are reachable by one $\alpha_i$-labelled transition from $P$.

**Definition 30** Given a CCSP$_{\mathrm{t}}^{\theta}$ process $P\in\mathbb{P}$, let $\widehat{P} := \sum_{\{(\alpha,Q)\mid P\xrightarrow{\alpha}Q\}}\alpha.Q$.
By Proposition 12, $P$ is countably branching, so using Proposition 15 $\widehat{P}$ is a valid CCSP$_{\mathrm{t}}^{\theta}$ process. In case $P\in\mathbb{P}$ is a process with only guarded recursion, then $P$ is finitely branching by Proposition 13, so also $\widehat{P}$ is a valid CCSP$_{\mathrm{t}}^{\theta}$ process with only guarded recursion.

**Proposition 31** Let $P\in\mathbb{P}$ have guarded recursion only. Then $Ax'\vdash P=\widehat{P}$. The conditional equation RSP is not even needed here.

**Proof:** The proof is by induction on the measure $e(P)$, defined in the proof of Proposition 13.

Let $P=\langle x|\mathcal{S}\rangle$. Axiom RDP yields $Ax\vdash P=\langle x|\mathcal{S}\rangle=\langle\mathcal{S}_x|\mathcal{S}\rangle$. Moreover, $e(\langle\mathcal{S}_x|\mathcal{S}\rangle)<e(\langle x|\mathcal{S}\rangle)$. So by induction, $Ax\vdash\langle\mathcal{S}_x|\mathcal{S}\rangle=\widehat{\langle\mathcal{S}_x|\mathcal{S}\rangle}$. Moreover, $\{(\alpha,Q)\mid\langle\mathcal{S}_x|\mathcal{S}\rangle\xrightarrow{\alpha}Q\}=\{(\alpha,Q)\mid\langle x|\mathcal{S}\rangle\xrightarrow{\alpha}Q\}$, so $\widehat{\langle\mathcal{S}_x|\mathcal{S}\rangle}=\widehat{\langle x|\mathcal{S}\rangle}=\widehat{P}$. Thus $Ax\vdash P=\widehat{P}$.

Let $P=\theta_L^U(P')$. Using that $e(P')<e(P)$, by induction $Ax\vdash P'=\widehat{P'}$ so $Ax\vdash P=\theta_L^U(\widehat{P'})$. Let

$$\widehat{P'}=\sum_{h\in H}\tau.P_h+\sum_{i\in I}a_i.Q_i+\sum_{j\in J}b_j.R_j+\sum_{k\in K}\gamma_k.T_k\ ,$$

where $a_i\in L$ for all $i\in I$, $b_j\in U\backslash L$ for all $j\in J$, and $\gamma_k\notin U\cup\{\tau\}$ for all $k\in K$. (So $\gamma_k$ may be t.) In case $H\cup I=\emptyset$, one has $Ax\vdash P=\theta_L^U(\widehat{P'})=\widehat{P'}=\widehat{P}$, using the first axiom for $\theta_L^U$. Otherwise

$$Ax\vdash P=\sum_{h\in H}\tau.\theta_L^U(P_h)+\sum_{i\in I}a_i.Q_i+\sum_{j\in J}b_j.R_j$$

by the remaining four axioms for $\theta_L^U$. The right-hand side is $\widehat{P}$.

The cases for the remaining operators are equally straightforward. □

In the special case that $P$ is a recursion-free process, also the axiom RDP is not needed for this result.

Once we have head normalisation, the proofs of Theorems 28 and 29 are independent of the precise syntax of the process algebra in question. Using Proposition 31 we immediately obtain (6) and (5):

**Theorem 32** For CCSP$_{\rm t}^\theta$ processes $P,Q \in \mathbb{P}$ with guarded recursion, one has $P \leftrightarrow Q$ iff $P=Q$ is derivable from the axioms of Tables 2 and 3. □

**Theorem 33** For CCSP$_{\rm t}^\theta$ processes $P,Q \in \mathbb{P}$ without recursion, one has $P \leftrightarrow Q$ iff $P=Q$ is derivable from the axioms of Tables 2 and 3 minus RDP and RSP.

A law that turns out to be particularly useful in verifications modulo strong reactive bisimilarity is

| $\theta_K^V(\theta_L^U(x)) \leftrightarrow \theta_{K\cup L}^{V\cap U}(x)$ provided $U=V$ or $K=L$ or $K\subseteq L\subseteq U\subseteq V$ or $L\subseteq K\subseteq V\subseteq U$ | (L1) |
|---|---|

.

Note that the right-hand side only exists if $(K\cup L)\subseteq(V\cap U)$. This law is sound for strong bisimilarity, as demonstrated by the following proposition. Yet it is not needed to add it to Table 3, as all its closed instances are derivable. In fact, this is a consequence of the above completeness theorems.

**Proposition 34** $\theta_K^V(\theta_L^U(P)) \leftrightarrow \theta_{K\cup L}^{V\cap U}(P)$, provided $(K\cup L)\subseteq(V\cap U)$ and either $U=V$ or $K=L$ or $K\subseteq L\subseteq U\subseteq V$ or $L\subseteq K\subseteq V\subseteq U$.

**Proof:** For given $K,L,U,V\subseteq A$ with $(K\cup L)\subseteq(V\cap U)$ and either $U=V$ or $K=L$ or $K\subseteq L\subseteq U\subseteq V$ or $L\subseteq K\subseteq V\subseteq U$, let

$$\mathscr{B} := Id \cup \left\{ \left(\theta_K^V(\theta_L^U(P)), \theta_{K\cup L}^{V\cap U}(P)\right) \mid P\in\mathbb{P}\right\}.$$

It suffices to show that the symmetric closure $\widetilde{\mathscr{B}}$ of $\mathscr{B}$ is a strong bisimulation. So let $R\ \widetilde{\mathscr{B}}\ T$ and $R \stackrel{\alpha}{\longrightarrow} R'$ with $\alpha\in A\cup\{\tau,{\rm t}\}$. I have to find a $T'$ with $T\stackrel{\alpha}{\longrightarrow} T'$ and $R'\ \widetilde{\mathscr{B}}\ T'$.

- The case that $R=T$ is trivial.
- Let $R=\theta_K^V(\theta_L^U(P))$ and $T=\theta_{K\cup L}^{V\cap U}(P)$.
  First assume $\alpha=\tau$. Then $P\stackrel{\tau}{\longrightarrow}P'$ for some $P'$ such that $R'=\theta_K^V(\theta_L^U(P'))$.
  Hence $T=\theta_{K\cup L}^{V\cap U}(P)\stackrel{\tau}{\longrightarrow}\theta_{K\cup L}^{V\cap U}(P') =: T'$, and $R'\ \mathscr{B}\ T'$.
  Now assume $\alpha\in A\cup\{{\rm t}\}$. Then $\theta_L^U(P)\stackrel{\alpha}{\longrightarrow}R'$ and either $\alpha\in V$ or $\theta_L^U(P)\stackrel{\beta}{\not\longrightarrow}$ for all $\beta\in K\cup\{\tau\}$. Using that $K\subseteq U$, this implies that either $\alpha\in V$ or $P\stackrel{\beta}{\not\longrightarrow}$ for all $\beta\in K\cup\{\tau\}$. Moreover, $P\stackrel{\alpha}{\longrightarrow}R'$ and either $\alpha\in U$ or $P\stackrel{\beta}{\not\longrightarrow}$ for all $\beta\in L\cup\{\tau\}$. It follows that either $\alpha\in V\cap U$ or $P\stackrel{\beta}{\not\longrightarrow}$ for all $\beta\in K\cup L\cup\{\tau\}$. (Here I use that either $U=V$ or $K=L$ or $K\subseteq L\subseteq U\subseteq V$ or $L\subseteq K\subseteq V\subseteq U$.) Consequently, $T=\theta_{K\cup L}^{V\cap U}(P)\stackrel{\alpha}{\longrightarrow}R'$.
- Let $R=\theta_{K\cup L}^{V\cap U}(P)$ and $T=\theta_K^V(\theta_L^U(P))$.
  First assume $\alpha=\tau$. Then $P\stackrel{\tau}{\longrightarrow}P'$ for some $P'$ such that $R'=\theta_{K\cup L}^{V\cap U}(P')$.
  Hence $T=\theta_K^V(\theta_L^U(P))\stackrel{\tau}{\longrightarrow}\theta_K^V(\theta_L^U(P')) =: T'$, and $R'\ \widetilde{\mathscr{B}}\ T'$.
  Now assume $\alpha\in A\cup\{{\rm t}\}$. Then $P\stackrel{\alpha}{\longrightarrow}R'$ and either $\alpha\in V\cap U$ or $P\stackrel{\beta}{\not\longrightarrow}$ for all $\beta\in K\cup L\cup\{\tau\}$. Consequently, $\theta_L^U(P)\stackrel{\alpha}{\longrightarrow}R'$ and thus $T=\theta_K^V(\theta_L^U(P))\stackrel{\alpha}{\longrightarrow}R'$. □

The side condition to L1 cannot be dropped, for $\theta_{\{c\}}^{\{a,c\}}\theta_\emptyset^{\{c\}}(a.0+c.0)\stackrel{a}{\longrightarrow}0$, yet $\theta_{\{c\}}^{\{c\}}(a.0+c.0)\stackrel{a}{\not\longrightarrow}$.

Table 4: A complete axiomatisation of strong reactive bisimilarity on guarded $\text{CCSP}_\text{t}^\theta$

$$\frac{\psi_X(x) = \psi_X(y) \text{ for all } X \subseteq A}{x = y} \qquad \text{(RA)}$$

### 10.3 A complete axiomatisation of strong reactive bisimilarity on guarded $\text{CCSP}_\text{t}^\theta$

To obtain a sound and complete axiomatisation of strong reactive bisimilarity for $\text{CCSP}_\text{t}^\theta$ with guarded recursion, one needs to combine the axioms of Tables 2, 3 and 4. These axioms are useful only in combination with the full congruence property of strong reactive bisimilarity, Theorem 25. This is what allows us to apply these axioms within subexpressions of a given expression. Since $\leftrightarroweq \subseteq \leftrightarroweq_r$, the soundness of all equational axioms for strong reactive bisimilarity follows from their soundness for strong bisimilarity. The soundness of RSP has been established as Theorem 27. The soundness of RA, the *reactive approximation axiom*, is contributed by the following proposition.

**Proposition 35** Let $P,Q \in \mathbb{P}$. If $\psi_X(P) \leftrightarroweq_r \psi_X(Q)$ for all $X \subseteq A$, then $P \leftrightarroweq_r Q$.

**Proof:** Given $P,Q \in \mathbb{P}$ with $\psi_X(P) \leftrightarroweq_r \psi_X(Q)$ for all $X \subseteq A$, I show that $\mathscr{B} := \leftrightarroweq_r \cup \{(P,Q),(Q,P)\}$ is a strong time-out bisimulation.

Let $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$. Take any $X \subseteq A$. Then $\psi_X(P) \xrightarrow{\alpha} P'$. Since $\psi_X(P) \leftrightarroweq_r \psi_X(Q)$, this implies $\psi_X(Q) \xrightarrow{\alpha} Q'$ for some $Q'$ with $P' \leftrightarroweq_r Q'$, and hence $Q \xrightarrow{\alpha} Q'$.

Let $P \xrightarrow{\text{t}} P'$ and $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$. Then $\psi_X(P) \xrightarrow{\text{t}} \theta_X(P')$ and $\mathcal{I}(\psi_X(P)) \cap (X \cup \{\tau\}) = \emptyset$. Since $\psi_X(P) \leftrightarroweq_r \psi_X(Q)$, this implies $\psi_X(Q) \xrightarrow{\text{t}} Q''$ for some $Q''$ with $\theta_X(\theta_X(P')) \leftrightarroweq_r \theta_X(Q'')$. It must be that $Q \xrightarrow{\text{t}} Q'$ for some $Q'$ with $Q'' = \theta_X(Q')$. By Proposition 34, $\theta_X(\theta_X(R)) \leftrightarroweq \theta_X(R)$ for all $R \in \mathbb{P}$. Thus $\theta_X(P') \leftrightarroweq \theta_X(\theta_X(P')) \leftrightarroweq_r \theta_X(\theta_X(Q')) \leftrightarroweq \theta_X(Q')$, which had to be shown. □

At first sight it appears that axiom RA is not very handy, as, in case the alphabet $A$ of visible actions is finite, the number of premises to verify is exponential in the size $A$. In case $A$ is infinite, there are even uncountably many premises. However, in practical verifications this is hardly an issue, as one uses a partition of the premises into a small number of equivalence classes, each of which requires only one common proof. This technique will be illustrated on three examples below. Furthermore, one could calculate the set of visible actions $\mathcal{J}(P)$ of a process $P$ that can be encountered as initial actions after one t-transition followed by a sequence of $\tau$-transitions. For large classes of processes, $\mathcal{J}(P)$ will be a finite set. Now axiom RA can be modified by changing $X \subseteq A$ into $X \subseteq \mathcal{J}(P) \cup \mathcal{J}(Q)$. This preserves the soundness of the axiom, because only the actions in $\mathcal{J}(P)$ play any rôle in evaluating $\psi_X(P)$.

A crucial property of strong reactive bisimilarity was mentioned in the introduction:

$$\tau.P + \text{t}.Q = \tau.P \qquad \text{(L2)}.$$

It is an immediate consequence of RA, since $\psi_X(\tau.P + \text{t}.Q) = \psi_X(\tau.P)$ for any $X \subseteq A$, by Table 3. Another useful law in verifications modulo strong reactive bisimilarity is

$$\textstyle\sum_{i\in I} a_i.x_i + \text{t}.y = \sum_{i\in I} a_i.x_i + \text{t}.\theta_\emptyset^{A\setminus In}(y), \text{ where } In = \{a_i \mid i \in I\}. \qquad \text{(L3)}$$

Its soundness is intuitively obvious: the t-transition to $y$ will be taken only in an environment $X$ with $X \cap In = \emptyset$. Hence one can just as well restrict the behaviour of $y$ to those transitions that are allowed in one such environment. This law was one of the prime reasons for extending the family of operators $\theta_X$ $(= \theta_X^X)$, which were needed to establish the key theorems of this paper, to the larger family $\theta_L^U$. Law L3 for finite $I$ is effortlessly derivable from its simple instance

$$a.x + \mathrm{t}.y = a.x + \mathrm{t}.\theta_\emptyset^{A\setminus\{a\}}(y). \qquad \text{(L3}'\text{)}$$

in combination with L1. I now show how to derive L3 from RA. For this proof I need to partition the set of premises of RA in only two equivalence classes.

First let $X \cap In \neq \emptyset$. Then $\psi_X(\sum_{i\in I} a_i.x_i + \mathrm{t}.y) = \sum_{i\in I} a_i.x_i = \psi_X(\sum_{i\in I} a_i.x_i + \mathrm{t}.\theta_\emptyset^{A\setminus In}(y))$.

Next let $X \cap In = \emptyset$. Then
$$\begin{aligned}\psi_X(\textstyle\sum_{i\in I} a_i.x_i + \mathrm{t}.y) &= \textstyle\sum_{i\in I} a_i.x_i + \mathrm{t}.\theta_X(y)\\ &= \textstyle\sum_{i\in I} a_i.x_i + \mathrm{t}.\theta_X(\theta_\emptyset^{A\setminus In}(y))\\ &= \psi_X(\textstyle\sum_{i\in I} a_i.x_i + \mathrm{t}.\theta_\emptyset^{A\setminus In}(y))\ ,\end{aligned}$$
where the second step is an application of L1.

As an application of L3$'$ one obtains the law from [18] that was justified in the introduction:

$$\begin{aligned}a.P + \mathrm{t}.(Q+\tau.R+a.S) &= a.P + \mathrm{t}.\theta_\emptyset^{A\setminus\{a\}}(Q+\tau.R+a.S)\\ &= a.P + \mathrm{t}.\theta_\emptyset^{A\setminus\{a\}}(Q+\tau.R)\\ &= a.P + \mathrm{t}.(Q+\tau.R)\ .\end{aligned}$$

As a third illustration of the use of RA I derive an equational law that does not follow from L1, L2 and L3, namely

$$b.P + \mathrm{t}.(a.Q + \tau.(b.R+a.S)) + \mathrm{t}.\tau.a.S = b.P + \mathrm{t}.(a.Q+\tau.a.S) + \mathrm{t}.\tau.(b.R+a.S)$$

These are the systems depicted in Figure 1. These systems are surely not strongly bisimilar. Moreover, L3 does not help in proving them equivalent, as applying $\theta_\emptyset^{A\setminus\{b\}}$ to any of the four targets of a t-transition does not kill any of the transitions of those processes. In particular, $\theta_\emptyset^{A\setminus\{b\}}(b.R+a.S) = b.R+a.S$. To derive this law from RA, I partition $\mathscr{P}(A)$ into three equivalence classes.

First let $b \in X$. Then
$$\begin{aligned}&\psi_X(b.P + \mathrm{t}.(a.Q + \tau.(b.R+a.S)) + \mathrm{t}.\tau.a.S)\\ &= b.P\\ &= \psi_X(b.P + \mathrm{t}.(a.Q+\tau.a.S) + \mathrm{t}.\tau.(b.R+a.S)).\end{aligned}$$

Next let $b \notin X$ and $a \in X$. Then

$$\begin{aligned}&\psi_X\big(b.P + \mathrm{t}.(a.Q + \tau.(b.R+a.S)) + \mathrm{t}.\tau.a.S\big)\\ &= b.P + \mathrm{t}.\theta_X\big(a.Q+\tau.(b.R+a.S)\big) + \mathrm{t}.\theta_X\big(\tau.a.S\big)\\ &= b.P + \mathrm{t}.(a.Q+\tau.\theta_X(b.R+a.S)) + \mathrm{t}.\tau.\theta_X(a.S)\\ &= b.P + \mathrm{t}.(a.Q+\tau.a.S) + \mathrm{t}.\tau.a.S\\ &= b.P + \mathrm{t}.(a.Q+\tau.\theta_X(a.S)) + \mathrm{t}.\tau.\theta_X\big(b.R+a.S\big)\\ &= b.P + \mathrm{t}.\theta_X\big(a.Q+\tau.a.S\big) + \mathrm{t}.\theta_X\big(\tau.(b.R+a.S)\big)\\ &= \psi_X\big(b.P + \mathrm{t}.(a.Q+\tau.a.S) + \mathrm{t}.\tau.(b.R+a.S)\big)\ .\end{aligned}$$

Finally let $a, b \notin X$. Then

$$\begin{aligned}&\psi_X\big(b.P + \mathrm{t}.(a.Q + \tau.(b.R+a.S)) + \mathrm{t}.\tau.a.S\big)\\ &= b.P + \mathrm{t}.\theta_X\big(a.Q+\tau.(b.R+a.S)\big) + \mathrm{t}.\theta_X\big(\tau.a.S\big)\\ &= b.P + \mathrm{t}.\tau.\theta_X(b.R+a.S) + \mathrm{t}.\tau.\theta_X(a.S)\\ &= b.P + \mathrm{t}.\tau.(b.R+a.S) + \mathrm{t}.\tau.a.S\\ &= b.P + \mathrm{t}.\tau.a.S + \mathrm{t}.\tau.(b.R+a.S)\\ &= b.P + \mathrm{t}.\tau.\theta_X(a.S) + \mathrm{t}.\tau.\theta_X\big(b.R+a.S\big)\\ &= b.P + \mathrm{t}.\theta_X\big(a.Q+\tau.a.S\big) + \mathrm{t}.\theta_X\big(\tau.(b.R+a.S)\big)\\ &= \psi_X\big(b.P + \mathrm{t}.(a.Q+\tau.a.S) + \mathrm{t}.\tau.(b.R+a.S)\big)\ .\end{aligned}$$

### 10.4 Completeness for finite processes

**Theorem 36** Let $P$ and $Q$ be closed recursion-free CCSP$_{\rm t}^\theta$ expressions. Then $P \leftrightarrows_r Q \Rightarrow Ax_f \vdash P = Q$.

**Proof:** Let the *length* of a path $P \xrightarrow{\alpha_1} P_1 \xrightarrow{\alpha_2} \dots \xrightarrow{\alpha_n} P_n$ of a processes $P$ be $n$. Let $d(P)$, the *depth* of $P$, be the length of its longest path; it is guaranteed to exists when $P$ is a closed recursion-free CCSP$_{\rm t}^\theta$ expression. I prove the theorem with induction on $\max(d(P), d(Q))$.

Suppose $P \leftrightarrows_r Q$. By Proposition 31 one has $Ax_f \vdash P = \widehat{P}$ and $Ax_f \vdash Q = \widehat{Q}$. I will show that $Ax_f \vdash \psi_X(\widehat{P}) = \psi_X(\widehat{Q})$ for all $X \subseteq A$. This will suffice, as then Axiom RA yields $Ax_f \vdash \widehat{P} = \widehat{Q}$ and thus $Ax_f \vdash P = Q$. So pick $X \subseteq A$. Let

$$\widehat{P} = \sum_{i\in I} \alpha_i.P'_i + \sum_{j\in J} {\rm t}.P''_j \qquad \text{and} \qquad \widehat{Q} = \sum_{k\in K} \beta_k.Q'_k + \sum_{h\in H} {\rm t}.Q''_h$$

with $\alpha_j, \beta_k \in A \cup \{\tau\}$ for all $i \in I$ and $k \in K$. The following two claims are the crucial part of the proof.

*Claim 1:* For each $i \in I$ there is a $k \in K$ with $\alpha_i = \beta_k$ and $Ax_f \vdash P'_i = Q'_k$.
*Claim 2:* If $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$, then for each $j \in J$ there is a $h \in H$ with $Ax_f \vdash \theta_X(P''_j) = \theta_X(Q''_h)$.

With these claims, the rest of the proof is straightforward. Since $P \leftrightarrows_r Q$, one has $\mathcal{I}(P) = \mathcal{I}(\widehat{P}) = \{\alpha_i \mid i \in I\} = \{\beta_k \mid k \in K\} = \mathcal{I}(\widehat{Q}) = \mathcal{I}(Q)$. First suppose that $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$. Then

$$\psi_X(\widehat{P}) = \sum_{i\in I} \alpha_i.P'_i + \sum_{j\in J} {\rm t}.\theta_X(P''_j) \qquad \text{and} \qquad \psi_X(\widehat{Q}) = \sum_{k\in K} \beta_k.Q'_k + \sum_{h\in H} {\rm t}.\theta_X(Q''_h)\ .$$

Claim 1 yields $Ax_f \vdash \psi_X(\widehat{Q}) = \psi_X(\widehat{Q}) + \alpha_i.P'_i$ for each $i \in I$. Likewise, Claim 2 yields $Ax_f \vdash \psi_X(\widehat{Q}) = \psi_X(\widehat{Q}) + {\rm t}.\theta_X(P''_j)$ for each $j \in J$. Together this yields $Ax_f \vdash \psi_X(\widehat{Q}) = \psi_X(\widehat{Q}) + \psi_X(\widehat{P})$. By symmetry one obtains $Ax_f \vdash \psi_X(\widehat{P}) = \psi_X(\widehat{P}) + \psi_X(\widehat{Q})$ and thus $Ax_f \vdash \psi_X(\widehat{P}) = \psi_X(\widehat{Q})$.

Next suppose $\mathcal{I}(P) \cap (X \cup \{\tau\}) \neq \emptyset$. Then $\psi_X(\widehat{P}) = \sum_{i\in I} \alpha_i.P'_i$ and $\psi_X(\widehat{Q}) = \sum_{k\in K} \beta_k.Q'_k$. The proof proceeds just as above, but without the need for Claim 2.

*Proof of Claim 1:* Pick $i \in I$. Then $\widehat{P} \xrightarrow{\alpha_i} P'_i$. So $\widehat{Q} \xrightarrow{\alpha_i} Q'$ for some $Q'$ with $P'_i \leftrightarrows_r Q'$. Hence there is a $k \in K$ with $\alpha_i = \beta_k$ and $Q' = Q'_k$. Using that $d(P'_i) < d(P)$ and $d(Q'_i) < d(Q)$, by induction $Ax_f \vdash P'_i = Q'_k$.

*Proof of Claim 2:* Pick $j \in J$. Then $\widehat{P} \xrightarrow{\rm t} P''_j$. Since $\mathcal{I}(\widehat{P}) \cap (X \cup \{\tau\}) = \emptyset$, there is a $Q''$ such that $\widehat{Q} \xrightarrow{\rm t} Q''$ and $\theta_X(P''_j) \leftrightarrows_r \theta_X(Q'')$. Hence there is a $h \in H$ with $Q'' = Q''_h$. Using that $d(\theta_X(P''_j)) \leq d(P''_j) < d(P)$ and $d(\theta_X(Q''_h)) \leq d(Q''_h) < d(Q)$, by induction $Ax_f \vdash \theta_X(P''_j) = \theta_X(Q''_h)$. □

### 10.5 The method of canonical representatives

The classic technique of proving completeness of axiomatisations for process algebras with recursion involves *merging guarded recursive equations* [31, 32, 40, 11, 30]. In essence it proves two bisimilar systems $P$ and $Q$ equivalent by equating both to an intermediate variant that is essentially a *product* of $P$ and $Q$. I tried so hard, and in vain, to apply this technique to obtain (8), that I came to believe that it fundamentally does not work for this axiomatisation. The problem is illustrated in Figure 3. Here, similar to the example of Figure 1, the processes 1 and 6 are strongly reactive bisimilar. The merging technique constructs a transition system whose states are pairs of states reachable from 1 and 6. There is a transition $(s,t) \xrightarrow{\alpha} (s',t')$ iff both $s \xrightarrow{\alpha} s'$ and $t \xrightarrow{\alpha} t'$. Normally, only those pairs $(s,t)$ satisfying

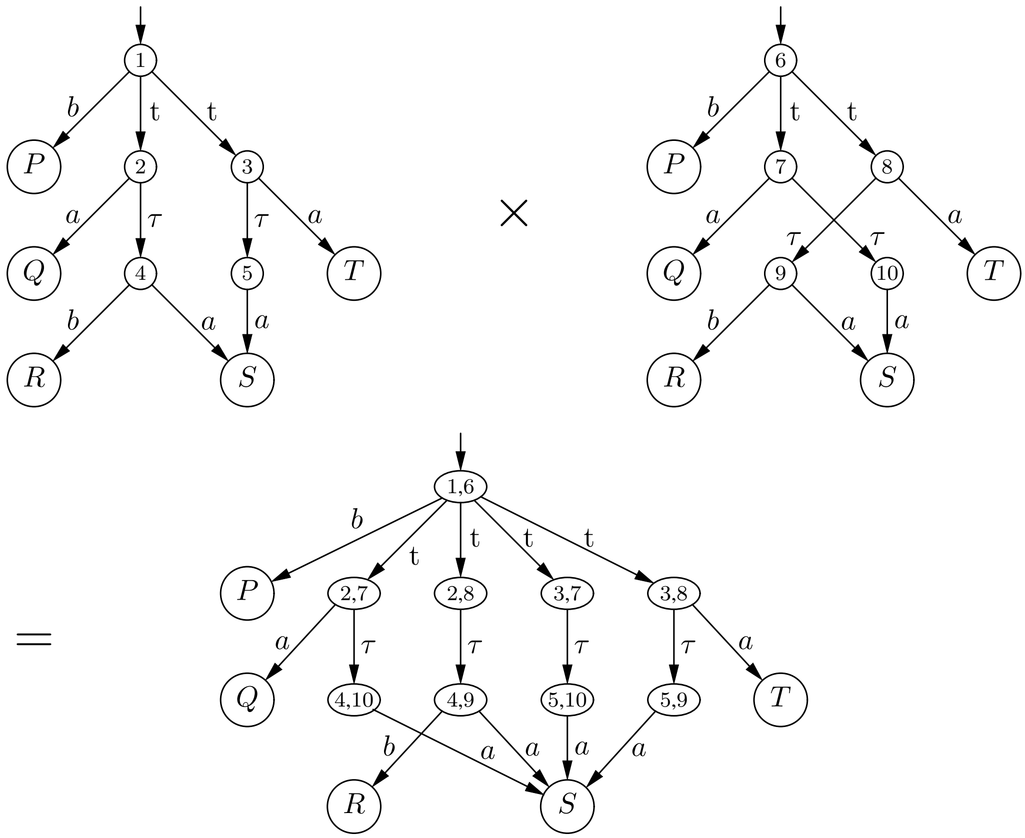


Figure 3: A failed product construction

$s \underline{\leftrightarrow} t$ are included. Here the requirement $s \underline{\leftrightarrow}_r t$ would be to strong. Namely, although $1 \underline{\leftrightarrow}_r 6$, one has neither $2 \underline{\leftrightarrow}_r 7$ nor $2 \underline{\leftrightarrow}_r 8$ nor $3 \underline{\leftrightarrow}_r 7$ nor $3 \underline{\leftrightarrow}_r 8$, so there would be no outgoing $\mathrm{t}$-transitions from $(1,6)$. Hence one has to include states $(s,t)$ with $s \underline{\leftrightarrow}_r^X t$ for some set $X$. Note that $2 \underline{\leftrightarrow}_r^X 7$ and $3 \underline{\leftrightarrow}_r^X 8$ when $a \in X$ and $b \notin X$, whereas $2 \underline{\leftrightarrow}_r^X 8$ and $3 \underline{\leftrightarrow}_r^X 7$ when $a \notin X$. This yields the product depicted in Figure 3.

In the reactive bisimulation game, the transition $1 \stackrel{\mathrm{t}}{\longrightarrow} 2$ will be matched by $6 \stackrel{\mathrm{t}}{\longrightarrow} 8$ only in an environment $X$ with $a \notin X$. Hence intuitively the state $(2,8)$ in the product should only be visited in such an environment. Yet, when aiming to show that $1 \underline{\leftrightarrow}_r (1,6) \underline{\leftrightarrow}_r 6$, one cannot prevent taking the transition $(1,6) \stackrel{\mathrm{t}}{\longrightarrow} (2,8)$ in an environment $X$ with $a \in X$ and $b \notin X$. However, since $(2,8) \stackrel{a}{\not\longrightarrow}$, this $\mathrm{t}$-transition cannot be simulated by process 2.

It may be possible to repair the construction, for instance by adding a transition $(2,8) \stackrel{a}{\longrightarrow} Q$ or $(2,8) \stackrel{a}{\longrightarrow} T$ after all, but not both. However, each such ad hoc repair that I tried gave raise to further problems, making the solution more and more complicated without sight on success.

Therefore, I here employ the novel method of *canonical solutions* [24, 29], which equates both $P$ and $Q$ to a canonical representative within the bisimulation equivalence class of $P$ and $Q$—one that has only one reachable state for each bisimulation equivalence class of states of $P$ and $Q$. Moreover, my proof employs the axiom of choice [41] in defining the transition relation on my canonical representative, in order to keep this process finitely branching.

To illustrate his technique on the example from Figure 3, the states 1 and 6, being strongly reactive bisimilar, form one new state $\{1,6\}$ of the canonical representative. Likewise, there will be states $\{4,9\}$

and $\{5,10\}$. However, the states 2, 3, 7 and 8 remain separate. Within the new state $\{1,6\}$ my construction chooses an arbitrary element, say 1. Based on this choice, the outgoing transitions of $\{1,6\}$ are dictated by 1, and thus go to $P$, $\{2\}$ and $\{3\}$. As a result, the canonical representative will look just like the left-hand process. It could however be the case that $S \leftrightarrow_r P$, in which case the initial states of these subprocesses are merged in the canonical representative, and again an element in the resulting equivalence class will be chosen that dictates its outgoing transitions.

### 10.6 The canonical representative

Let $\mathbb{P}^g$ denote the set of CCSP$^\theta_{\rm t}$ processes with guarded recursion. Let $[P] := \{Q \in \mathbb{P}^g \mid Q \leftrightarrow_r P\}$ be the strong reactive bisimulation equivalence class of a process $P \in \mathbb{P}^g$. Below, by "abstract process" I will mean such an equivalence class. Choose a function $\chi$ that selects an element out of each $\leftrightarrow_r$-equivalence class of CCSP$^\theta_{\rm t}$ processes with guarded recursion—this is possible by the axiom of choice [41]. Define the transition relations $\xrightarrow{\alpha}$, for $\alpha \in Act$, between abstract processes by

$$R \xrightarrow{\alpha} R' \;\;\Leftrightarrow\;\; \exists P' \in R'.\, \chi(R) \xrightarrow{\alpha} P' \,. \tag{9}$$

I will show that $P \leftrightarrow_r [P]$ for all $P \in \mathbb{P}^g$. Formally, $\leftrightarrow_r$ has been defined only between processes belonging to the same LTS $\mathbb{P}$, and here $[P] \notin \mathbb{P}$. However, this restriction is not material: two processes $P \in \mathbb{P}$ and $Q \in \mathbb{Q}$ from different LTSs can be compared by considering $\leftrightarrow_r$ on the disjoint union $\mathbb{P} \uplus \mathbb{Q}$.

**Lemma 7** Let $\alpha \in A \cup \{\tau\}$. Then $[P] \xrightarrow{\alpha} R'$ iff $P \xrightarrow{\alpha} P'$ for some $P'$ with $R' = [P']$.

**Proof:** Let $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$. Since $P \leftrightarrow_r \chi([P])$, by Definition 8 there is a $Q'$ such that $\chi([P]) \xrightarrow{\alpha} Q'$ and $P' \leftrightarrow_r Q'$. Hence $[P] \xrightarrow{\alpha} [Q']$ by (9). Moreover, $P' \in [Q']=[P']$.

Let $[P] \xrightarrow{\alpha} R'$ with $\alpha \in A \cup \{\tau\}$. Then $\chi([P]) \xrightarrow{\alpha} Q'$ for some $Q' \in R'$. Since $\chi([P]) \leftrightarrow_r P$, there is a $P'$ such that $P \xrightarrow{\alpha} P'$ and $Q' \leftrightarrow_r P'$. Hence $P' \in R'$ and thus $R' = [P']$. □

**Corollary 37** $\mathcal{I}([P]) = \mathcal{I}(P)$ for all $P \in \mathbb{P}^g$. □

**Lemma 8** If $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\rm t} P'$ then $[P] \xrightarrow{\rm t} [Q']$ for a $Q'$ with $\theta_X(P') \leftrightarrow_r \theta_X(Q')$. Moreover, if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $[P] \xrightarrow{\rm t} [Q']$ then $P \xrightarrow{\rm t} P'$ for a $P'$ with $\theta_X(Q') \leftrightarrow_r \theta_X(P')$.

**Proof:** Let $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\rm t} P'$. Since $P \leftrightarrow_r \chi([P])$, by Definition 8 there is a $Q'$ such that $\chi([P]) \xrightarrow{\rm t} Q'$ and $\theta_X(P') \leftrightarrow_r \theta_X(Q')$. Hence $[P] \xrightarrow{\rm t} [Q']$ by (9).

Let $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $[P] \xrightarrow{\rm t} [Q']$. Then $\chi([P]) \xrightarrow{\rm t} R'$ for some $R' \in [Q']$. Since $\chi([P]) \leftrightarrow_r P$ (so $\mathcal{I}(\chi([P])) = \mathcal{I}(P)$), there is a $P'$ such that $P \xrightarrow{\rm t} P'$ and $\theta_X(R') \leftrightarrow_r \theta_X(P')$. As $\leftrightarrow_r$ is a congruence for $\theta_X$, one has $\theta_X(Q') \leftrightarrow_r \theta_X(R')$, and thus $\theta_X(Q') \leftrightarrow_r \theta_X(P')$. □

**Definition 38** Let $\mathscr{B}^* := \{(R,T) \mid \exists n \geq 0.\, \exists R_0, \ldots, R_n.\ R = R_0 \mathrel{\mathscr{B}} R_1 \mathrel{\mathscr{B}} \cdots \mathrel{\mathscr{B}} R_n = T\}$ denote the reflexive and transitive closure of a binary relation $\mathscr{B}$. A *strong time-out bisimulation up to reflexivity and transitivity* is a symmetric relation $\mathscr{B} \subseteq \mathbb{P} \times \mathbb{P}$, such that, for $P \mathrel{\mathscr{B}} Q$,

- if $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$, then $\exists Q'$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \mathrel{\mathscr{B}^*} Q'$,
- if $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \xrightarrow{\rm t} P'$, then $\exists Q'$ with $Q \xrightarrow{\rm t} Q'$ and $\theta_X(P') \mathrel{\mathscr{B}^*} \theta_X(Q')$.

**Proposition 39** If $P \mathrel{\mathscr{B}} Q$ for a strong time-out bisimulation $\mathscr{B}$ up to reflexivity and transitivity, then $P \leftrightarrow_r Q$.

**Proof:** It suffices to show that $\mathscr{B}^*$ is a strong time-out bisimulation. Clearly this relation is symmetric.

- Suppose $R_0 \mathscr{B} R_1 \mathscr{B} \cdots \mathscr{B} R_n$ for some $n \geq 0$ and $R_0 \xrightarrow{\alpha} R_0'$ with $\alpha \in A \cup \{\tau\}$. I have to find an $R_n'$ such that $R_n \xrightarrow{\alpha} R_n'$ and $R_0' \mathscr{B}^* R_n'$. I proceed with induction on $n$. The case $n = 0$ is trivial. Fixing an $n > 0$, by Definition 38 there is an $R_1'$ such that $R_1 \xrightarrow{\alpha} R_1'$ and $R_0' \mathscr{B}^* R_1'$. Now by induction there is an $R_n'$ such that $R_n \xrightarrow{\alpha} R_n'$ and $R_1' \mathscr{B}^* R_n'$. Hence $R_0' \mathscr{B}^* R_n'$.
- Suppose $R_0 \mathscr{B} R_1 \mathscr{B} \cdots \mathscr{B} R_n$ for some $n \geq 0$, $\mathcal{I}(R_0) \cap (X \cup \{\tau\}) = \emptyset$ and $R_0 \xrightarrow{\mathrm{t}} R_0'$. By Definition 38 $\mathcal{I}(R_0) = \mathcal{I}(R_1) = \cdots = \mathcal{I}(R_n)$. I have to find an $R_n'$ such that $R_n \xrightarrow{\mathrm{t}} R_n'$ and $\theta_X(R_0') \mathscr{B}^* \theta_X(R_n')$. This proceeds exactly as for the case above. □

**Lemma 9** $\theta_X([P]) \leftrightarrow_r [\theta_X(P)]$ for all $P \in \mathbb{P}^g$ and $X \subseteq A$.

**Proof:** I show that the symmetric closure of $\mathscr{B} := \{(\theta_X([P]), [\theta_X(P)]) \mid P \in \mathbb{P}^g \wedge X \subseteq A\}$ is a strong time-out bisimulation up to reflexivity and transitivity.

- Let $\theta_X([P]) \xrightarrow{\tau} R'$. Then $[P] \xrightarrow{\tau} Q'$ for some $Q'$ with $R' = \theta_X(Q')$. By Lemma 7, $P \xrightarrow{\tau} P'$ for some $P'$ with $Q' = [P']$. Hence $\theta_X(P) \xrightarrow{\tau} \theta_X(P')$ and thus $[\theta_X(P)] \xrightarrow{\tau} [\theta_X(P')]$ by Lemma 7. Moreover, $R' = \theta_X([P']) \mathscr{B} [\theta_X(P')]$.
- Let $[\theta_X(P)] \xrightarrow{\tau} R'$. By Lemma 7, $\theta_X(P) \xrightarrow{\tau} Q'$ for some $Q'$ with $R' = [Q']$. Thus $P \xrightarrow{\tau} P'$ for some $P'$ with $Q' = \theta_X(P')$. Now $[P] \xrightarrow{\tau} [P']$ by Lemma 7, and thus $\theta_X([P]) \xrightarrow{\tau} \theta_X([P'])$. Moreover, $R' = [\theta_X(P')] \mathscr{B}^{-1} \theta_X([P'])$.
- Let $\theta_X([P]) \xrightarrow{a} R'$ with $a \in A$. Then $[P] \xrightarrow{a} R'$ and either $a \in \mathcal{I}([P])$ or $[P] \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Thus either $a \in \mathcal{I}(P)$ or $P \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$, using Corollary 37. By Lemma 7, $P \xrightarrow{a} P'$ for some $P'$ with $R' = [P']$. Hence $\theta_X(P) \xrightarrow{a} P'$ and thus $[\theta_X(P)] \xrightarrow{a} [P'] = R'$.
- Let $[\theta_X(P)] \xrightarrow{a} R'$ with $a \in A$. By Lemma 7, $\theta_X(P) \xrightarrow{a} P'$ for some $P'$ with $R' = [P']$. Thus $P \xrightarrow{a} P'$ and either $a \in \mathcal{I}(P)$ or $P \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Therefore either $a \in \mathcal{I}([P])$ or $[P] \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$, using Corollary 37. Moreover, $[P] \xrightarrow{a} [P']$ by Lemma 7. It follows that $\theta_X([P]) \xrightarrow{a} [P'] = R'$.
- Let $\mathcal{I}(\theta_X([P])) \cap (X \cup \{\tau\}) = \emptyset$ and $\theta_X([P]) \xrightarrow{\mathrm{t}} R' = [Q']$. Then $[P] \xrightarrow{\mathrm{t}} [Q']$ and $[P] \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Thus $P \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$, using Corollary 37, so $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $\mathcal{I}(\theta_X(P)) \cap (X \cup \{\tau\}) = \emptyset$. By Lemma 8, $P \xrightarrow{\mathrm{t}} P'$ for some $P'$ with $\theta_X(Q') \leftrightarrow_r \theta_X(P')$. Hence $\theta_X(P) \xrightarrow{\mathrm{t}} P'$ and thus, again applying Lemma 8, $[\theta_X(P)] \xrightarrow{\mathrm{t}} [T']$ for some $T'$ with $\theta_X(P') \leftrightarrow_r \theta_X(T')$. Moreover, $\theta_X(R') = \theta_X([Q']) \mathscr{B} [\theta_X(Q')] = [\theta_X(T')] \mathscr{B}^{-1} \theta_X([T'])$.
- Let $\mathcal{I}([\theta_X(P)]) \cap (X \cup \{\tau\}) = \emptyset$ and $[\theta_X(P)] \xrightarrow{\mathrm{t}} [Q']$. By Lemma 8, $\theta_X(P) \xrightarrow{\mathrm{t}} P'$ for a $P'$ with $\theta_X(Q') \leftrightarrow_r \theta_X(P')$. Hence $P \xrightarrow{\mathrm{t}} P'$ and $P \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$, so $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$. Hence, by Lemma 8, $[P] \xrightarrow{\mathrm{t}} [T']$ for a $T'$ with $\theta_X(P') \leftrightarrow_r \theta_X(T')$. By Corollary 37, $[P] \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. So $\theta_X([P]) \xrightarrow{\mathrm{t}} [T']$. Moreover, $\theta_X([Q']) \mathscr{B} [\theta_X(Q')] = [\theta_X(T')] \mathscr{B}^{-1} \theta_X([T'])$. □

**Proposition 40** $P \leftrightarrow_r [P]$ for all $P \in \mathbb{P}^g$.

**Proof:** Using Proposition 24, I show that the symmetric closure of the relation $\mathscr{B} := \{(P, [P]) \mid P \in \mathbb{P}^g\}$ is a strong time-out bisimulation up to $\leftrightarrow_r$. Here the right-hand side processes come from an LTS that is closed under $\theta$ and contains the processes $[P]$ for $P \in \mathbb{P}^g$.

- Let $P \xrightarrow{\alpha} P'$ with $\alpha \in A \cup \{\tau\}$. Then $[P] \xrightarrow{\alpha} [P']$ by Lemma 7, and $P' \mathscr{B} [P']$.
- Let $[P] \xrightarrow{\alpha} R'$ with $\alpha \in A \cup \{\tau\}$. Then, by Lemma 7, $P \xrightarrow{\alpha} P'$ for some $P'$ with $R' = [P']$. Moreover, $R' \mathscr{B}^{-1} P'$.

- Let $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$ and $P \stackrel{\mathrm{t}}{\longrightarrow} P'$. By Lemma 8, $[P] \stackrel{\mathrm{t}}{\longrightarrow} [Q']$ for some $Q'$ such that $\theta_X(P') \leftrightarrow_r \theta_X(Q')$. Moreover, using Lemma 9, $\theta_X(P') \mathscr{B} [\theta_X(P')] = [\theta_X(Q')] \leftrightarrow_r \theta_X([Q'])$.
- Let $\mathcal{I}([P]) \cap (X \cup \{\tau\}) = \emptyset$ and $[P] \stackrel{\mathrm{t}}{\longrightarrow} [Q']$. Then, by Lemma 8, $P \stackrel{\mathrm{t}}{\longrightarrow} P'$ for some $P'$ with $\theta_X(Q') \leftrightarrow_r \theta_X(P')$. By Lemma 9, $\theta_X([Q']) \leftrightarrow_r [\theta_X(Q')] = [\theta_X(P')] \mathscr{B}^{-1} \theta_X(P')$. □

By Proposition 13 each $P \in \mathbb{P}^g$ is finitely branching. By construction, so is $[P]$.

No two states reachable from $[P]$ are strongly reactive bisimilar. Hence the process $[P]$ with its above-generated transition relation can be seen as a version of $P$ were each equivalence class of reachable states is collapsed into a single state—a kind of minimisation. But it is not exactly a minimisation, as not all states reachable from $[P]$ need be strongly reactive bisimilar with reachable states of $P$. This is illustrated by Process 6 of Figure 3, when $\chi(\{1,6\}) = 1$. Now $\{2\}$ and $\{3\}$ are reachable from $[P]$, but not strongly reactive bisimilar with reachable states of 6.

### 10.7 Completeness for finitely branching processes

I will now give a syntactic representation of each process $[P]$, for $P \in \mathbb{P}^g$, as a CCSP$_\mathrm{t}^\theta$ process with guarded recursion. Take a different variable $x_R$ for each $\leftrightarrow_r$-equivalence class $R$ of CCSP$_\mathrm{t}^\theta$ processes with guarded recursion. Let $V_\mathcal{S}$ be the set of all those variables, and define the recursive specification $\mathcal{S}$ by

$$x_R = \sum_{R \stackrel{\alpha}{\longrightarrow} R'} \alpha.x_{R'} \ .$$

By construction, $R \leftrightarrow \langle x_R | \mathcal{S}\rangle$, that is, the process $\mathbf{[}P\mathbf{]} := \langle x_{[P]} | \mathcal{S}\rangle \in \mathbb{P}^g$ is strongly bisimilar to $[P]$. In fact, the symmetric closure of the relation $\{([P], \mathbf{[}P\mathbf{]}) \mid P \in \mathbb{P}^g\}$ is a strong bisimulation. Thus, $\mathbf{[}P\mathbf{]}$ serves as a normal form within the $\leftrightarrow_r$-equivalence class of $P \in \mathbb{P}^g$.

The above construction will not work when there are not as many variables as equivalences classes of CCSP$_\mathrm{t}^\theta$ processes with guarded recursion. Note that each real number in the interval $[0,1)$ can be represented as an infinite sequence of 0s and 1s, and thus as a CCSP$_\mathrm{t}^\theta$ processes with guarded recursion employing the finite alphabet $A = \{0,1\}$. Hence there are uncountably many equivalences classes of CCSP$_\mathrm{t}^\theta$ processes with guarded recursion.

To solve this problem, one starts here already with the proof of (8), and fixes two processes $P_0$ and $Q_0 \in \mathbb{P}^g$ with $P_0 \leftrightarrow_r Q_0$. The task is to prove $\mathit{Ax} \vdash P_0 = Q_0$. Now call an equivalence class $R$ of CCSP$_\mathrm{t}^\theta$ processes with guarded recursion *relevant* if either $R$ is reachable from $[P_0] = [Q_0]$, or a member of $R$ is reachable from $P_0$ or $Q_0$. There are only countably many relevant equivalence classes. It suffices to take a variable $x_R$ only for relevant $R$. Below, I will call a process $P \in \mathbb{P}^g$ *relevant* if it is a member of a relevant equivalence class; in case we had enough variables to start with, all processes $P \in \mathbb{P}^g$ may be called relevant.

**Lemma 10** Let $P, Q \in \mathbb{P}^g$ be relevant. Then $\mathbf{[}P\mathbf{]} \leftrightarrow_r \mathbf{[}Q\mathbf{]} \Rightarrow \mathbf{[}P\mathbf{]} = \mathbf{[}Q\mathbf{]}$.

**Proof:** Suppose $\mathbf{[}P\mathbf{]} \leftrightarrow_r \mathbf{[}Q\mathbf{]}$. Then $P \leftrightarrow_r \mathbf{[}P\mathbf{]} \leftrightarrow_r \mathbf{[}Q\mathbf{]} \leftrightarrow_r Q$, so $[P] = [Q]$, and hence $\mathbf{[}P\mathbf{]} = \mathbf{[}Q\mathbf{]}$. □

**Lemma 11** Let $P, Q \in \mathbb{P}^g$ be relevant. Then $\theta_X(\mathbf{[}P\mathbf{]}) \leftrightarrow_r \theta_X(\mathbf{[}Q\mathbf{]}) \Rightarrow \theta_X(\mathbf{[}P\mathbf{]}) \leftrightarrow \theta_X(\mathbf{[}Q\mathbf{]})$.

**Proof:** I show that $\mathscr{B} := \mathit{Id} \cup \{(\theta_X(\mathbf{[}P\mathbf{]}), \theta_X(\mathbf{[}Q\mathbf{]})) \mid \theta_X(\mathbf{[}P\mathbf{]}) \leftrightarrow_r \theta_X(\mathbf{[}Q\mathbf{]})\}$ is a strong bisimulation.

Suppose $\theta_X(\mathbf{[}P\mathbf{]}) \leftrightarrow_r \theta_X(\mathbf{[}Q\mathbf{]})$. Then $\mathbf{[}P\mathbf{]} \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$ iff $\mathbf{[}Q\mathbf{]} \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$, since $\mathcal{I}(\mathbf{[}P\mathbf{]}) \cap (X \cup \{\tau\}) = \mathcal{I}(\theta_X(\mathbf{[}P\mathbf{]})) \cap (X \cup \{\tau\}) = \mathcal{I}(\theta_X(\mathbf{[}Q\mathbf{]})) \cap (X \cup \{\tau\}) = \mathcal{I}(\mathbf{[}Q\mathbf{]}) \cap (X \cup \{\tau\})$.

First consider the case that $\mathbf{[}P\mathbf{]} \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Then $\theta_X(\mathbf{[}P\mathbf{]}) \leftrightarrow \mathbf{[}P\mathbf{]}$ and $\theta_X(\mathbf{[}Q\mathbf{]}) \leftrightarrow \mathbf{[}Q\mathbf{]}$. Hence $\mathbf{[}P\mathbf{]} \leftrightarrow \theta_X(\mathbf{[}P\mathbf{]}) \leftrightarrow_r \theta_X(\mathbf{[}Q\mathbf{]}) \leftrightarrow \mathbf{[}Q\mathbf{]}$. So by Lemma 10, $\mathbf{[}P\mathbf{]} = \mathbf{[}Q\mathbf{]}$, and thus $\theta_X(\mathbf{[}P\mathbf{]}) \ \mathit{Id}\ \theta_X(\mathbf{[}Q\mathbf{]})$.

Henceforth I suppose that $\mathbf{[}P\mathbf{]} \stackrel{\beta}{\longrightarrow}$ for some $\beta \in X \cup \{\tau\}$. So $\mathbf{[}P\mathbf{]} \stackrel{\mathrm{t}}{\not\longrightarrow}$ and $\mathbf{[}Q\mathbf{]} \stackrel{\mathrm{t}}{\not\longrightarrow}$.

- Let $\theta_X([P]) \stackrel{a}{\longrightarrow} P''$ with $a \in A$. Then $\theta_X([Q]) \stackrel{a}{\longrightarrow} Q''$ for some $Q''$ with $P'' \leftrightarrow_r Q''$. One has $[P] \stackrel{a}{\longrightarrow} P''$ and $[Q] \stackrel{a}{\longrightarrow} Q''$. The process $P''$ must have the form $[P']$, and likewise $Q'' = [Q']$. Since $[P'] \leftrightarrow_r [Q']$, Lemma 10 yields $[P'] = [Q']$.
- Let $\theta_X([P]) \stackrel{\tau}{\longrightarrow} P''$. Then $\theta_X([Q]) \stackrel{\tau}{\longrightarrow} Q''$ for some $Q''$ with $P'' \leftrightarrow_r Q''$. The process $P''$ must have the form $\theta_X([P'])$, and likewise $Q'' = \theta_X([Q'])$. Hence $P'' \mathscr{B} Q''$. □

**Definition 41** Given a relevant CCSP$^\theta_{\rm t}$ process $P \in \mathbb{P}^g$, let $\widetilde{P} := \sum_{\{(\alpha,Q) \mid P \stackrel{\alpha}{\longrightarrow} Q\}} \alpha.[Q]$.

Thus, $\widetilde{P}$ is defined like the head-normal form $\widehat{P}$ of $P \in \mathbb{P}^g$, except that all processes $Q$ reachable from $P$ by performing one transition are replaced by the normal form within their $\leftrightarrow_r$-equivalence class. So $P \leftrightarrow \widehat{P} \leftrightarrow_r \widetilde{P}$. Note that $[P] = \widetilde{\chi([P])}$ is provable through a single application of the axiom RDP.

The following step is the only one where the reactive approximation axiom (RA) is used.

**Proposition 42** Let $P, Q \in \mathbb{P}^g$ be relevant. Then $P \leftrightarrow_r Q \Rightarrow Ax \vdash \widetilde{P} = \widetilde{Q}$.

**Proof:** Suppose $P \leftrightarrow_r Q$. Then $\widetilde{P} \leftrightarrow_r P \leftrightarrow_r Q \leftrightarrow_r \widetilde{Q}$. With Axiom RA it suffices to show that $Ax_f \vdash \psi_X(\widetilde{P}) = \psi_X(\widetilde{Q})$ for all $X \subseteq A$. So pick $X \subseteq A$. Let

$$\widetilde{P} = \sum_{i \in I} \alpha_i.P'_i + \sum_{j \in J} {\rm t}.P''_j \qquad \text{and} \qquad \widetilde{Q} = \sum_{k \in K} \beta_k.Q'_k + \sum_{h \in H} {\rm t}.Q''_h$$

with $\alpha_j, \beta_k \in A \cup \{\tau\}$ for all $i \in I$ and $k \in K$. As for Theorem 36, the following two claims are crucial.

*Claim 1:* For each $i \in I$ there is a $k \in K$ with $\alpha_i = \beta_k$ and $Ax \vdash P'_i = Q'_k$.
*Claim 2:* If $\mathcal{I}(P) \cap (X \cup \{\tau\}) = \emptyset$, then for each $j \in J$ there is a $h \in H$ with $Ax \vdash \theta_X(P''_j) = \theta_X(Q''_h)$.

With these claims the proof proceeds exactly as the one of Theorem 36.

*Proof of Claim 1:* Pick $i \in I$. Then $\widetilde{P} \stackrel{\alpha_i}{\longrightarrow} P'_i$. So $\widetilde{Q} \stackrel{\alpha_i}{\longrightarrow} Q'$ for some $Q'$ with $P'_i \leftrightarrow_r Q'$. Hence there is a $k \in K$ with $\alpha_i = \beta_k$ and $Q' = Q'_k$. The processes $P'_i$ and $Q'_k$ must have the form $[P']$ and $[Q']$ for some $P', Q' \in \mathbb{P}^g$. Hence, by Lemma 10, $P'_i = Q'_k$, and thus certainly $Ax \vdash P'_i = Q'_k$.

*Proof of Claim 2:* Pick $j \in J$. Then $\widetilde{P} \stackrel{\rm t}{\longrightarrow} P''_j$. Since $\mathcal{I}(\widetilde{P}) \cap (X \cup \{\tau\}) = \emptyset$, there is a $Q''$ such that $\widetilde{Q} \stackrel{\rm t}{\longrightarrow} Q''$ and $\theta_X(P''_j) \leftrightarrow_r \theta_X(Q'')$. Hence there is a $h \in H$ with $Q'' = Q''_h$. The processes $P''_j$ and $Q''_h$ have the form $[P'']$ and $[Q'']$ for some $P'', Q'' \in \mathbb{P}^g$. So by Lemma 11, $\theta_X(P''_j) \leftrightarrow \theta_X(Q'')$. The completeness of $Ax$ for strong bisimilarity (Theorem 32) now yields $Ax \vdash \theta_X(P''_j) = \theta_X(Q'')$. □

**Theorem 43** Let $P \in \mathbb{P}^g$ be relevant. Then $Ax \vdash P = [P]$.

**Proof:** Let *reach*$(P)$ be the set of processes reachable from $P$. Take a different variable $z_R$ for each $R \in$ *reach*$(P)$, and define the recursive specification $\mathcal{S}'$ by $V_{\mathcal{S}'} := \{z_R \mid R \in \textit{reach}(P)\}$ and

$$z_R = \sum_{R \stackrel{\alpha}{\longrightarrow} R'} \alpha.z_{R'} \; .$$

By construction, $R \leftrightarrow \langle x_R | \mathcal{S} \rangle$. In fact, the symmetric closure of $\{(R, \langle x_R|\mathcal{S}\rangle) \mid R \in \textit{reach}(P)\}$ is a strong bisimulation. To establish Theorem 43 through an application of RSP, I show that both $P$ and $[P]$ are $x_P$-components of solutions of $\mathcal{S}'$. So I show

$$Ax \vdash R = \sum_{R \stackrel{\alpha}{\longrightarrow} R'} \alpha.R' \qquad \text{and} \qquad Ax \vdash [R] = \sum_{R \stackrel{\alpha}{\longrightarrow} R'} \alpha.[R']$$

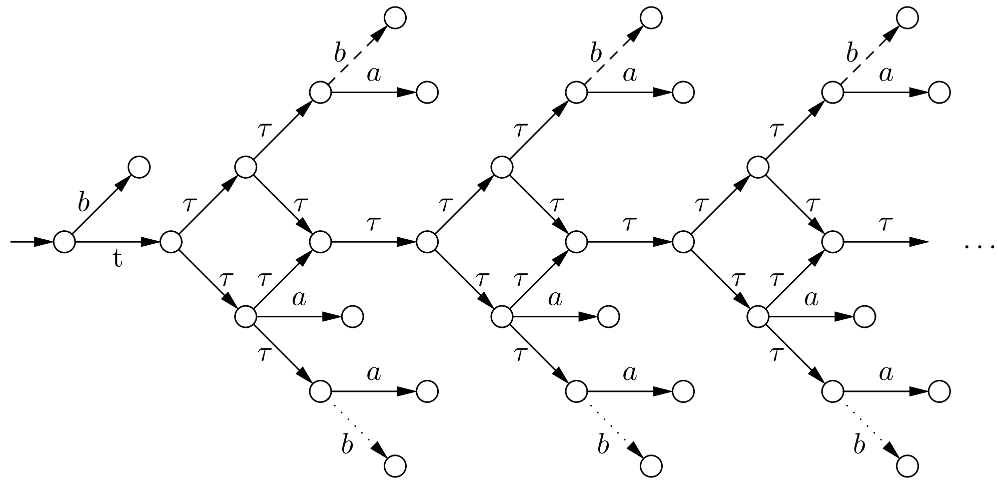


Figure 4: An uncountable variety of strongly reactive bisimilar processes

for all $R \in \textit{reach}(P)$. The first of these statements is a direct application of Proposition 31. The second statement can be reformulated as *Ax* $\vdash$ **[**$R$**]** $= \widetilde{R}$. As remarked above, *Ax* $\vdash$ **[**$R$**]** $= \widetilde{\chi([R])}$ through a single application of RDP. Hence I need to show that *Ax* $\vdash \widetilde{\chi([R])} = \widetilde{R}$. Considering that $\chi([R]) \leftrightarrow_r R$, this is a consequence of Proposition 42. □

**Corollary 44** Let $P, Q \in \mathbb{P}^g$ be relevant. Then $P \leftrightarrow_r Q \Rightarrow$ *Ax* $\vdash P = Q$.

**Proof:** Let $P \leftrightarrow_r Q$. Then **[**$P$**]** = **[**$Q$**]** by Lemma 10, so *Ax* $\vdash P =$ **[**$P$**]** = **[**$Q$**]** $= Q$. □

## 10.8 Necessity of the axiom of choice

At first glance it may look like the above proof can be simplified so as to avoid using the axiom of choice, namely by changing (9) into

$$R \xrightarrow{\alpha} R' \;\;\Leftrightarrow\;\; \exists P \in R, P' \in R'.\, P \xrightarrow{\alpha} P' \;.$$

However, this would make some processes $[P]$ infinitely branching, even when $P$ is finitely branching. Figure 4 shows an uncountable collection of strongly reactive bisimilar finitely branching processes. Here each pair of a dashed $b$-transition and the dotted one right below it constitutes a design choice: either the dashed or the dotted $b$-transition is present, but not both. Since there is this binary choice for infinitely many pairs of $b$-transitions, this figure represents an uncountable collection of processes. All of them are strongly reactive bisimilar, because the $\mathrm{t}$-transition will only be taken in an environment that blocks $b$. In case $a$ is blocked as well, all the $a$-transitions from a state with an outgoing $\tau$-transition can be dropped, and the difference between these processes disappears. In case $a$ is allowed by the environment, all $b$ transitions can be dropped, and again the difference between these processes disappears. Hence the above alternative definition would yield uncountably many outgoing $\mathrm{t}$-transitions from the equivalence class of all these processes. This would make it impossible to represent such a "minimised" process in $\mathrm{CCSP}_{\mathrm{t}}^{\theta}$.

## 11 Concluding remarks

This paper laid the foundations of the proper analogue of strong bisimulation semantics for a process algebra with time-outs. This makes it possible to specify systems in this setting and verify their correctness properties. The addition of time-outs comes with considerable gains in expressive power. An illustration of this is mutual exclusion.

As shown in [20], it is fundamentally impossible to correctly specify mutual exclusion protocols in standard process algebras, such as CCS [33], CSP [6, 28], ACP [2, 10] or CCSP, unless the correctness of the specified protocol hinges on a fairness assumption. The latter, in the view of [20], does not provide an adequate solution, as fairness assumptions are in many situations unwarranted and lead to false conclusions. In [9] a correct process-algebraic rendering of mutual exclusion is given, but only after making two important modifications to standard process algebra. The first involves making a justness assumption. Here *justness* [21] is an alternative to fairness, in some sense a much weaker form of fairness—meaning weaker than weak fairness. Unlike (strong or weak) fairness, its use typically is warranted and does not lead to false conclusions. The second modification is the addition of a new construct—*signals*—to CCS, or any other standard process algebra. Interestingly, both modifications are necessary; just using justness, or just adding signals, is insufficient. Bouwman [4, 5] points out that since the justness requirement was fairly new, and needed to be carefully defined to describe its interaction with signals anyway, it is possible to specify mutual exclusion without adding signals to the language at all, instead reformulating the justness requirement in such a way that it effectively turns some actions into signals. Yet justness is essential in all these approaches. This may be seen as problematic, because large parts of the foundations of process algebra are incompatible with justness, and hence need to be thoroughly reformulated in a justness-friendly way. This is pointed out in [17].

The addition of time-outs to standard process algebra makes it possible to specify mutual exclusion without assuming justness! Instead, one should make the assumption called *progress* in [21], which is weaker than justness, uncontroversial, unproblematic, and made (explicitly or implicitly) in virtually all papers dealing with issues like mutual exclusion. This claim is substantiated in [19].

Besides applications to protocol verification, future work includes adapting the work done here to a form of reactive bisimilarity that abstracts from hidden actions, that is, to provide a counterpart for process algebras with time-outs of, for instance, branching bisimilarity [23], weak bisimilarity [33] or coupled similarity [36, 12, 3]. Other topics worth exploring are the extension to probabilistic processes, and especially the relations with timed process algebras. Davies & Schneider in [7], for instance, added a construct with a quantified time-out to the process algebra CSP [6, 28], elaborating the timed model of CSP presented by Reed & Roscoe in [38].

**Acknowledgement.** My thanks to the CONCUR'20 and Acta Informatica referees for helpful feedback.

## References

[1] J.C.M. Baeten, J.A. Bergstra & J.W. Klop (1986): *Syntax and defining equations for an interrupt mechanism in process algebra*. Fundamenta Informaticae IX(2), pp. 127–168, doi:10.3233/FI-1986-9202.

[2] J.C.M. Baeten & W.P. Weijland (1990): *Process Algebra*. Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press, doi:10.1017/CBO9780511624193.

[3] B. Bisping, U. Nestmann & K. Peters (2020): *Coupled similarity: the first 32 years*. Acta Informatica 57(3-5), pp. 439–463, doi:10.1007/s00236-019-00356-4.

[4] M.S. Bouwman (2018): *Liveness analysis in process algebra: simpler techniques to model mutex algorithms*. Technical Report, Eindhoven University of Technology. Available at `http://www.win.tue.nl/~timw/downloads/bouwman_seminar.pdf`.

[5] M.S. Bouwman, B. Luttik & T.A.C. Willemse (2020): *Off-the-shelf automated analysis of liveness properties for just paths*. Acta Informatica 57(3-5), pp. 551–590, doi:10.1007/s00236-020-00371-w.

[6] S.D. Brookes, C.A.R. Hoare & A.W. Roscoe (1984): *A theory of communicating sequential processes*. Journal of the ACM 31(3), pp. 560–599, doi:10.1145/828.833.

[7] J. Davies & S. Schneider (1993): *Recursion Induction for Real-Time Processes*. Formal Aspects of Computing 5(6), pp. 530–553, doi:10.1007/BF01211248.

[8] R. De Nicola & M. Hennessy (1984): *Testing equivalences for processes*. Theoretical Computer Science 34, pp. 83–133, doi:10.1016/0304-3975(84)90113-0.

[9] V. Dyseryn, R.J. van Glabbeek & P. Höfner (2017): *Analysing Mutual Exclusion using Process Algebra with Signals*. In K. Peters & S. Tini, editors: Proceedings Combined 24th International Workshop on Expressiveness in Concurrency and 14th Workshop on Structural Operational Semantics, Electronic Proceedings in Theoretical Computer Science 255, Open Publishing Association, pp. 18–34, doi:10.4204/EPTCS.255.2.

[10] W. J. Fokkink (2000): *Introduction to Process Algebra*. Texts in Theoretical Computer Science, An EATCS Series, Springer, doi:10.1007/978-3-662-04293-9.

[11] R.J. van Glabbeek (1993): *A complete axiomatization for branching bisimulation congruence of finite-state behaviours*. In A.M. Borzyszkowski & S. Sokołowski, editors: Proceedings $18^{th}$ International Symposium on Mathematical Foundations of Computer Science, MFCS '93, LNCS 711, Springer, pp. 473–484, doi:10.1007/3-540-57182-5_39.

[12] R.J. van Glabbeek (1993): *The Linear Time – Branching Time Spectrum II; The semantics of sequential systems with silent moves*. In E. Best, editor: Proceedings $4^{th}$ International Conference on Concurrency Theory, CONCUR'93, LNCS 715, Springer, pp. 66–81, doi:10.1007/3-540-57208-2_6.

[13] R.J. van Glabbeek (1994): *On the expressiveness of ACP (extended abstract)*. In A. Ponse, C. Verhoef & S.F.M. van Vlijmen, editors: Proceedings First Workshop on the Algebra of Communicating Processes, ACP'94, Workshops in Computing, Springer, pp. 188–217, doi:10.1007/978-1-4471-2120-6_8.

[14] R.J. van Glabbeek (2001): *The Linear Time – Branching Time Spectrum I; The Semantics of Concrete, Sequential Processes*. In J.A. Bergstra, A. Ponse & S.A. Smolka, editors: Handbook of Process Algebra, chapter 1, Elsevier, pp. 3–99, doi:10.1016/B978-044482830-9/50019-9.

[15] R.J. van Glabbeek (2004): *The Meaning of Negative Premises in Transition System Specifications II*. Journal of Logic and Algebraic Programming 60–61, pp. 229–258, doi:10.1016/j.jlap.2004.03.007.

[16] R.J. van Glabbeek (2017): *Lean and Full Congruence Formats for Recursion*. In: Proceedings $32^{nd}$ Annual ACM/IEEE Symposium on Logic in Computer Science, LICS'17, IEEE Computer Society Press, doi:10.1109/LICS.2017.8005142. Available at `https://arxiv.org/abs/1704.03160`.

[17] R.J. van Glabbeek (2019): *Ensuring liveness properties of distributed systems: Open problems*. Journal of Logical and Algebraic Methods in Programming 109:100480, doi:10.1016/j.jlamp.2019.100480. Available at `http://arxiv.org/abs/1912.05616`.

[18] R.J. van Glabbeek (2021): *Failure Trace Semantics for a Process Algebra with Time-outs*. Logical Methods in Computer Science 17(2):11, doi:10.23638/LMCS-17(2:11)2021.

[19] R.J. van Glabbeek (2021): *Modelling Mutual Exclusion in a Process Algebra with Time-outs*. Available at `https://arxiv.org/abs/2106.12785`.

[20] R.J. van Glabbeek & P. Höfner (2015): *CCS: It's not fair! Fair schedulers cannot be implemented in CCS-like languages even under progress and certain fairness assumptions*. Acta Informatica 52(2-3), pp. 175–205, doi:10.1007/s00236-015-0221-6. Available at `http://arxiv.org/abs/1505.05964`.

[21] R.J. van Glabbeek & P. Höfner (2019): *Progress, Justness and Fairness*. ACM Computing Surveys 52(4):69, doi:10.1145/3329125. Available at `https://arxiv.org/abs/1810.07414`.

[22] R.J. van Glabbeek & C.A. Middelburg (2020): *On Infinite Guarded Recursive Specifications in Process Algebra*. Available at `http://arxiv.org/abs/2005.00746`.

[23] R.J. van Glabbeek & W.P. Weijland (1996): *Branching Time and Abstraction in Bisimulation Semantics*. Journal of the ACM 43(3), pp. 555–600, doi:10.1145/233551.233556.

[24] C. Grabmayer & W.J. Fokkink (2020): *A Complete Proof System for 1-Free Regular Expressions Modulo Bisimilarity*. In H. Hermanns, L. Zhang, N. Kobayashi & D. Miller, editors: Proc. 35th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS'20, ACM, pp. 465–478, doi:10.1145/3373718.3394744.

[25] J.F. Groote (1993): *Transition System Specifications with Negative Premises*. Theoretical Computer Science 118, pp. 263–299, doi:10.1016/0304-3975(93)90111-6.

[26] J.F. Groote & F.W. Vaandrager (1992): *Structured Operational Semantics and Bisimulation as a Congruence*. Information and Computation 100(2), pp. 202–260, doi:10.1016/0890-5401(92)90013-6.

[27] M. Hennessy & R. Milner (1985): *Algebraic laws for nondeterminism and concurrency*. Journal of the ACM 32(1), pp. 137–161, doi:10.1145/2455.2460.

[28] C.A.R. Hoare (1985): *Communicating Sequential Processes*. Prentice Hall, Englewood Cliffs.

[29] X. Liu & T.Yu (2020): *Canonical Solutions to Recursive Equations and Completeness of Equational Axiomatisations*. In I. Konnov & L. Kovacs, editors: Proceedings 31st International Conference on Concurrency Theory (CONCUR 2020), Leibniz International Proceedings in Informatics (LIPIcs) 171, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, doi:10.4230/LIPIcs.CONCUR.2020.35.

[30] M. Lohrey, P.R. D'Argenio & H. Hermanns (2005): *Axiomatising divergence*. Information and Computation 203(2), pp. 115–144, doi:10.1016/j.ic.2005.05.007.

[31] R. Milner (1984): *A complete inference system for a class of regular behaviours*. Journal of Computer and System Sciences 28, pp. 439–466, doi:10.1016/0022-0000(84)90023-0.

[32] R. Milner (1989): *A Complete Axiomatisation for Observational Congruence of Finite-State Behaviors*. Information and Computation 81(2), pp. 227–247, doi:10.1016/0890-5401(89)90070-9.

[33] R. Milner (1990): *Operational and algebraic semantics of concurrent processes*. In J. van Leeuwen, editor: Handbook of Theoretical Computer Science, chapter 19, Elsevier Science Publishers B.V. (North-Holland), pp. 1201–1242. 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.

[34] E.-R. Olderog (1987): *Operational Petri net semantics for CCSP*. In G. Rozenberg, editor: Advances in Petri Nets 1987, LNCS 266, Springer, pp. 196–223, doi:10.1007/3-540-18086-9_27.

[35] E.-R. Olderog & C.A.R. Hoare (1986): *Specification-oriented semantics for communicating processes*. Acta Informatica 23, pp. 9–66, doi:10.1007/BF00268075.

[36] J. Parrow & P. Sjödin (1992): *Multiway synchronization verified with coupled simulation*. In W.R. Cleaveland, editor: Proceedings CONCUR 92, Stony Brook, NY, USA, LNCS 630, Springer, pp. 518–533, doi:10.1007/BFb0084813.

[37] M. Pohlmann (2021): *Reducing Strong Reactive Bisimilarity to Strong Bisimilarity*. Bachelor's thesis, TU Berlin. Available at https://maxpohlmann.github.io/Reducing-Reactive-to-Strong-Bisimilarity/thesis.pdf.

[38] G.M. Reed & A.W. Roscoe (1988): *A Timed Model for Communicating Sequential Processes*. Theoretical Computer Science 58, pp. 249–261, doi:10.1016/0304-3975(88)90030-8.

[39] F.W. Vaandrager (1993): *Expressiveness Results for Process Algebras*. In J.W. de Bakker, W.P. de Roever & G. Rozenberg, editors: Proceedings REX Workshop on Semantics: Foundations and Applications, Beekbergen, The Netherlands, 1992, LNCS 666, Springer, pp. 609–638, doi:10.1007/3-540-56596-5_49.

[40] D.J. Walker (1990): *Bisimulation and divergence*. Information and Computation 85(2), pp. 202–241, doi:10.1016/0890-5401(90)90048-M.

[41] Ernst Zermelo (1908): *Untersuchungen über die Grundlagen der Mengenlehre I*. Mathematische Annalen 65(2), pp. 261–281, doi:10.1007/bf01449999.

# A Initials congruence

This appendix contains the proofs of two facts about initials equivalence I need in this paper, namely that it is a full congruence for $\text{CCSP}_\text{t}^\theta$, and that it is not affected by which processes are substituted for variables whose free occurrences are guarded.

**Theorem 18** Initials equivalence is a full congruence for $\text{CCSP}_\text{t}^\theta$.

**Proof:** Let $\mathscr{B} \subseteq \mathbb{P} \times \mathbb{P}$ be the smallest relation satisfying

- if $\mathcal{S}$ and $\mathcal{S}'$ are recursive specifications with $x \in V_\mathcal{S} = V_{\mathcal{S}'}$ and $\langle x|\mathcal{S}\rangle, \langle x|\mathcal{S}'\rangle \in \mathbb{P}$, such that $\mathcal{S}_y =_I \mathcal{S}'_y$ for all $y \in V_\mathcal{S}$, then $\langle x|\mathcal{S}\rangle \mathscr{B} \langle x|\mathcal{S}'\rangle$,
- if $P =_\mathcal{I} Q$, then $P \mathscr{B} Q$,
- if $P \mathscr{B} Q$ and $\alpha \in A \cup \{\tau, \text{t}\}$, then $\alpha.P \mathscr{B} \alpha.Q$,
- if $P_1 \mathscr{B} Q_1$ and $P_2 \mathscr{B} Q_2$, then $P_1 + P_2 \mathscr{B} Q_1 + Q_2$,
- if $P_1 \mathscr{B} Q_1$, $P_2 \mathscr{B} Q_2$ and $S \subseteq A$, then $P_1 \|_S P_2 \mathscr{B} Q_1 \|_S Q_2$,
- if $P \mathscr{B} Q$ and $I \subseteq A$, then $\tau_I(P) \mathscr{B} \tau_I(Q)$,
- if $P \mathscr{B} Q$ and $\mathcal{R} \subseteq A \times A$, then $\mathcal{R}(P) \mathscr{B} \mathcal{R}(Q)$,
- if $P \mathscr{B} Q$, $L \subseteq U \subseteq A$ and $X \subseteq A$, then $\theta_L^U(P) \mathscr{B} \theta_L^U(Q)$ and $\psi_X(P) \mathscr{B} \psi_X(Q)$,
- if $\mathcal{S}$ is a recursive specification with $z \in V_\mathcal{S}$, and $\rho, \nu : \mathit{Var} \setminus V_\mathcal{S} \to \mathbb{P}$ are substitutions satisfying $\rho(x) \mathscr{B} \nu(x)$ for all $x \in \mathit{Var} \setminus V_\mathcal{S}$, then $\langle z|\mathcal{S}\rangle[\rho] \mathscr{B} \langle z|\mathcal{S}\rangle[\nu]$.

A trivial structural induction on $E \in \mathbb{E}$ (not using the first two clauses) shows that

$$\text{if } \rho, \nu : \mathit{Var} \to \mathbb{P} \text{ satisfy } \rho(x) \mathscr{B} \nu(x) \text{ for all } x \in \mathit{Var}, \text{ then } E[\rho] \mathscr{B} E[\nu]. \quad (*)$$

For $\mathcal{S}$ a recursive specification and $\rho : \mathit{Var} \setminus V_\mathcal{S} \to \mathbb{P}$, let $\rho_\mathcal{S} : \mathit{Var} \to \mathbb{P}$ be the closed substitution given by $\rho_\mathcal{S}(x) := \langle x|\mathcal{S}\rangle[\rho]$ if $x \in V_\mathcal{S}$ and $\rho_\mathcal{S}(x) := \rho(x)$ otherwise. Then $\langle E|\mathcal{S}\rangle[\rho] = E[\rho_\mathcal{S}]$ for all $E \in \mathbb{E}$. Hence an application of (*) with $\rho_\mathcal{S}$ and $\nu_\mathcal{S}$ yields that under the conditions of the last clause for $\mathscr{B}$ above one even has $\langle E|\mathcal{S}\rangle[\rho] \mathscr{B} \langle E|\mathcal{S}\rangle[\nu]$ for all expressions $E \in \mathbb{E}$, ($)
and likewise, in the first clause, $\langle E|\mathcal{S}\rangle \mathscr{B} \langle E|\mathcal{S}'\rangle$ for all $E \in \mathbb{E}$ with variables from $V_\mathcal{S}$. (#)

It suffices to show that $P \mathscr{B} Q \Rightarrow P =_\mathcal{I} Q$, because then $\mathscr{B} = =_\mathcal{I}$, and (*) implies that $\mathscr{B}$ is a lean congruence. Moreover, the clauses for $\mathscr{B}$ (not needing the last) then imply that $=_\mathcal{I}$ is a full congruence. This I will do by induction on the *stratum* $(\lambda_R, \kappa_R)$ of processes $R \in \mathbb{P}$, as defined in Section 5. So pick a stratum $(\lambda, \kappa)$ and assume that $P' \mathscr{B} Q' \Rightarrow P' =_\mathcal{I} Q'$ for all $P', Q' \in \mathbb{P}$ with $(\lambda_P, \kappa_P) < (\lambda, \kappa)$ and $(\lambda_Q, \kappa_Q) < (\lambda, \kappa)$. I need to show that $P \mathscr{B} Q \Rightarrow P =_\mathcal{I} Q$ for all $P, Q \in \mathbb{P}$ with $(\lambda_P, \kappa_P) \leq (\lambda, \kappa)$ and $(\lambda_Q, \kappa_Q) \leq (\lambda, \kappa)$.

Because $=_I$ is symmetric, so is $\mathscr{B}$. Hence, it suffices to show that $P \mathscr{B} Q \wedge P \xrightarrow{\alpha} \Rightarrow Q \xrightarrow{\alpha}$ for all $P, Q \in \mathbb{P}$ with $(\lambda_P, \kappa_P), (\lambda_Q, \kappa_Q) \leq (\lambda, \kappa)$ and all $\alpha \in A \cup \{\tau\}$. This I will do by structural induction on the proof $\pi$ of $P \xrightarrow{\alpha}$ from the rules of Table 1. I make a case distinction based on the derivation of $P \mathscr{B} Q$. So assume $P \mathscr{B} Q$, $(\lambda_P, \kappa_P), (\lambda_Q, \kappa_Q) \leq (\lambda, \kappa)$, and $P \xrightarrow{\alpha}$ with $\alpha \in A \cup \{\tau\}$.

- Let $P = \langle x|\mathcal{S}\rangle \in \mathbb{P}$ and $Q = \langle x|\mathcal{S}'\rangle \in \mathbb{P}$ where $\mathcal{S}$ and $\mathcal{S}'$ are recursive specifications with $x \in V_\mathcal{S} = V_{\mathcal{S}'}$, such that $\mathcal{S}_y =_\mathcal{I} \mathcal{S}'_y$ for all $y \in V_\mathcal{S}$, meaning that for all $y \in V_\mathcal{S}$ and $\sigma : V_\mathcal{S} \to \mathbb{P}$ one has $\mathcal{S}_y[\sigma] =_\mathcal{I} \mathcal{S}'_y[\sigma]$.
  By Table 1 the transition $\langle \mathcal{S}_x|\mathcal{S}\rangle \xrightarrow{\alpha}$ is provable by means of a strict subproof of $\pi$. By (#) above one has $\langle \mathcal{S}_x|\mathcal{S}\rangle \mathscr{B} \langle \mathcal{S}_x|\mathcal{S}'\rangle$. So by induction $\langle \mathcal{S}_x|\mathcal{S}'\rangle \xrightarrow{\alpha}$. Since $\langle \_|\mathcal{S}'\rangle$ is the application of a substitution of the form $\sigma : V_{\mathcal{S}'} \to \mathbb{P}$, one has $\langle \mathcal{S}_x|\mathcal{S}'\rangle =_\mathcal{I} \langle \mathcal{S}'_x|\mathcal{S}'\rangle$. Hence $\langle \mathcal{S}'_x|\mathcal{S}'\rangle \xrightarrow{\alpha}$. By Table 1, $Q = \langle x|\mathcal{S}'\rangle \xrightarrow{\alpha}$.

- The case $P =_\mathcal{I} Q$ is trivial.
- Let $P = \beta.P^\dagger$ and $Q = \beta.Q^\dagger$ with $\beta \in A \cup \{\tau, \mathrm{t}\}$ and $P^\dagger \mathrel{\mathscr{B}} Q^\dagger$. Then $\alpha = \beta$ and $Q \xrightarrow{\alpha}$.
- Let $P = P_1 + P_2$ and $Q = Q_1 + Q_2$ with $P_1 \mathrel{\mathscr{B}} Q_1$ and $P_2 \mathrel{\mathscr{B}} Q_2$. I consider the first rule from Table 1 that could have been responsible for the derivation of $P \xrightarrow{\alpha}$; the other proceeds symmetrically. So suppose that $P_1 \xrightarrow{\alpha}$. Then by induction $Q_1 \xrightarrow{\alpha}$. By the same rule, $Q \xrightarrow{\alpha}$.
- Let $P = P_1 \|_S P_2$ and $Q = Q_1 \|_S Q_2$ with $P_1 \mathrel{\mathscr{B}} Q_1$ and $P_2 \mathrel{\mathscr{B}} Q_2$. I consider the three rules from Table 1 that could have been responsible for the derivation of $P \xrightarrow{\alpha}$.

  First suppose that $\alpha \notin S$, and $P_1 \xrightarrow{\alpha}$. By induction, $Q_1 \xrightarrow{\alpha}$. Consequently, $Q_1 \|_S Q_2 \xrightarrow{\alpha}$.

  Next suppose that $\alpha \in S$, $P_1 \xrightarrow{\alpha}$ and $P_2 \xrightarrow{\alpha}$. By induction, $Q_1 \xrightarrow{\alpha}$ and $Q_2 \xrightarrow{\alpha}$. So $Q_1 \|_S Q_2 \xrightarrow{\alpha}$.

  The remaining case proceeds symmetrically to the first.
- Let $P = \tau_I(P^\dagger)$ and $Q = \tau_I(Q^\dagger)$ with $I \subseteq A$ and $P^\dagger \mathrel{\mathscr{B}} Q^\dagger$. Then $P^\dagger \xrightarrow{\beta}$ and either $\beta = \alpha \notin I$, or $\beta \in I$ and $\alpha = \tau$. By induction, $Q^\dagger \xrightarrow{\beta}$. Consequently, $Q = \tau_I(Q^\dagger) \xrightarrow{\alpha}$.
- Let $P = \mathcal{R}(P^\dagger)$ and $Q = \mathcal{R}(Q^\dagger)$ with $\mathcal{R} \subseteq A \times A$ and $P^\dagger \mathrel{\mathscr{B}} Q^\dagger$. Then $P^\dagger \xrightarrow{\beta}$ and either $(\beta, \alpha) \in \mathcal{R}$ or $\beta = \alpha = \tau$. By induction, $Q^\dagger \xrightarrow{\beta}$. Consequently, $Q = \mathcal{R}(Q^\dagger) \xrightarrow{\alpha}$.
- Let $P = \theta_L^U(P^\dagger)$, $Q = \theta_L^U(Q^\dagger)$ and $P^\dagger \mathrel{\mathscr{B}} Q^\dagger$. Then $(\lambda_{P^\dagger}, \kappa_{P^\dagger}) < (\lambda, \kappa)$ and $(\lambda_{Q^\dagger}, \kappa_{Q^\dagger}) < (\lambda, \kappa)$, as remarked in Section 5. So by induction $P^\dagger =_\mathcal{I} Q^\dagger$. (This is the only use of stratum induction.) Since $\theta_L^U(P^\dagger) \xrightarrow{\alpha}$, it must be that $P^\dagger \xrightarrow{\alpha}$ and either $\alpha \in U \cup \{\tau\}$ or $P^\dagger \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in L \cup \{\tau\}$. In the latter case, $Q^\dagger \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in L \cup \{\tau\}$. Moreover, $Q^\dagger \xrightarrow{\alpha}$. So, in both cases, $Q = \theta_L^U(Q^\dagger) \xrightarrow{\alpha}$.
- Let $P = \psi_X(P^\dagger)$, $Q = \psi_X(Q^\dagger)$ and $P^\dagger \mathrel{\mathscr{B}} Q^\dagger$. Since $\psi_X(P^\dagger) \xrightarrow{\alpha}$, one has $P^\dagger \xrightarrow{\alpha}$. By induction $Q^\dagger \xrightarrow{\alpha}$. So $Q = \psi_X(Q^\dagger) \xrightarrow{\alpha}$.
- Let $P = \langle z|\mathcal{S}\rangle[\rho] = \langle z|\mathcal{S}[\rho]\rangle$ and $Q = \langle z|\mathcal{S}\rangle[\nu] = \langle z|\mathcal{S}[\nu]\rangle$ where $\mathcal{S}$ is a recursive specification with $z \in V_\mathcal{S}$, and $\rho, \nu : \mathit{Var} \setminus V_\mathcal{S} \rightarrow \mathbb{P}$ satisfy $\rho(x) \mathrel{\mathscr{B}} \nu(x)$ for all $x \in \mathit{Var} \setminus V_\mathcal{S}$. By Table 1 the transition $\langle \mathcal{S}_z|\mathcal{S}[\rho]\rangle \xrightarrow{\alpha}$ is provable by means of a strict subproof of the proof $\pi$ of $\langle z|\mathcal{S}\rangle[\rho] \xrightarrow{\alpha}$. By ($) above one has $\langle \mathcal{S}_z|\mathcal{S}[\rho]\rangle \mathrel{\mathscr{B}} \langle \mathcal{S}_z|\mathcal{S}[\nu]\rangle$. So by induction, $\langle \mathcal{S}_z|\mathcal{S}[\nu]\rangle \xrightarrow{a}$. By Table 1, $Q = \langle z|\mathcal{S}[\nu]\rangle \xrightarrow{\alpha}$. □

**Lemma 5** Let $H \in \mathbb{E}$ be guarded and have free variables from $W \subseteq \mathit{Var}$ only, and let $\vec{P}, \vec{Q} \in \mathbb{P}^W$. Then $\mathcal{I}(H[\vec{P}]) = \mathcal{I}(H[\vec{Q}])$.

**Proof:** Lemma 5 can be strengthened as follows.

> Let $H \in \mathbb{E}$ be such that all free occurrences of variables from $W \subseteq \mathit{Var}$ in $H$ are guarded, and let $\vec{P}, \vec{Q} \in \mathbb{P}^W$. Then $H[\vec{P}] =_\mathcal{I} H[\vec{Q}]$.

The proof proceeds with structural induction on $H$.

- Let $H = \langle x|\mathcal{S}\rangle$, so that $H[\vec{P}] = \langle x|\mathcal{S}[\vec{P}^\dagger]\rangle$, where $\vec{P}^\dagger$ is the $W \setminus V_\mathcal{S}$-tuple that is left of $\vec{P}$ after deleting the $y$-components, for $y \in V_\mathcal{S}$, and $H[\vec{Q}] = \langle x|\mathcal{S}[\vec{Q}^\dagger]\rangle$. For each $y \in V_\mathcal{S}$, all free occurrences of variables from $W \setminus V_\mathcal{S}$ in $\mathcal{S}_y$ are guarded. Thus, by induction, $\mathcal{S}_y[\vec{P}^\dagger] =_\mathcal{I} \mathcal{S}_y[\vec{Q}^\dagger]$. Since $=_\mathcal{I}$ is a full congruence for CCSP$_\mathrm{t}^\theta$, it follows that $H[\vec{P}] = \langle x|\mathcal{S}[\vec{P}^\dagger]\rangle =_\mathcal{I} \langle x|\mathcal{S}[\vec{Q}^\dagger]\rangle = H[\vec{Q}]$.
- Let $H = \alpha.H'$ for some $\alpha \in \mathit{Act}$. Then $\mathcal{I}(H[\vec{P}]) = \mathcal{I}(H[\vec{Q}])$ (namely $\emptyset$ if $\alpha = \mathrm{t}$ and $\{\alpha\}$ otherwise).
- Let $H = H_1 \|_S H_2$. Since all free occurrences of variables from $W \subseteq \mathit{Var}$ in $H$ are guarded, so are those in $H_1$ and $H_2$. Thus, by induction, $H_1[\vec{P}] =_\mathcal{I} H_1[\vec{Q}]$ and $H_2[\vec{P}] =_\mathcal{I} H_2[\vec{Q}]$. Since $=_\mathcal{I}$ is a full congruence for $S$ it follows that $H[\vec{P}] =_\mathcal{I} H[\vec{Q}]$.
- The cases for all other operators go exactly like the case for $\|_S$. □

# B Proofs of lemmas on $\theta_X$ and strong bisimilarity from Section 7.2

The following lemmas on the relation between $\theta_X$ and the other operators of $\text{CCSP}_\text{t}^\theta$ deal with strong bisimilarity, but are needed in the congruence proof for strong reactive bisimilarity.

**Lemma 12** If $\mathcal{I}(Q) \cap (Y \cup \{\tau\}) = \emptyset$ then $\theta_Y(Q) \leftrightarrows Q$.

**Proof:** This follows immediately from the operational rules for $\theta_Y$. □

**Lemma 2** If $P \stackrel{\tau}{\not\longrightarrow}$, $\mathcal{I}(P) \cap X \subseteq S$ and $Y = X \setminus (S \setminus \mathcal{I}(P))$, then $\theta_X(P \|_S Q) \leftrightarrows \theta_X(P \|_S \theta_Y(Q))$.

**Proof:** Let $P \in \mathbb{P}$ and $S, X, Y \subseteq A$ be as indicated in the lemma. Let

$$\mathscr{B} := \leftrightarrows \cup \{(\theta_X(P \|_S Q), \theta_X(P \|_S \theta_Y(Q))) \mid Q \in \mathbb{P}\}$$

It suffices to show that the symmetric closure $\widetilde{\mathscr{B}}$ of $\mathscr{B}$ is a strong bisimulation.
So let $R \, \widetilde{\mathscr{B}} \, T$ and $R \stackrel{\alpha}{\longrightarrow} R'$ with $\alpha \in A \cup \{\tau, \text{t}\}$. I have to find a $T'$ with $T \stackrel{\alpha}{\longrightarrow} T'$ and $R' \, \widetilde{\mathscr{B}} \, T'$.

- The case that $R \leftrightarrows T$ is trivial.
- Let $R = \theta_X(P \|_S Q)$ and $T = \theta_X(P \|_S \theta_Y(Q))$, for some $Q \in \mathbb{P}$.
  First assume $\alpha = \tau$. Then $Q \stackrel{\tau}{\longrightarrow} Q'$ for some $Q'$ with $R' = \theta_X(P \|_S Q')$. Consequently, $T = \theta_X(P \|_S \theta_Y(Q)) \stackrel{\tau}{\longrightarrow} \theta_X(P \|_S \theta_Y(Q')) =: T'$ and $R' \, \mathscr{B} \, T'$.
  Now assume $\alpha \in A \cup \{\text{t}\}$. Then $P \|_S Q \stackrel{\alpha}{\longrightarrow} R'$. I first deal with the case that $\alpha \in X$, and consider the three rules from Table 1 that could have derived $P \|_S Q \stackrel{\alpha}{\longrightarrow} R'$.
  - The case that $\alpha \notin S$ and $P \stackrel{\alpha}{\longrightarrow} P'$ cannot occur, because $\mathcal{I}(P) \cap X \subseteq S$.
  - Let $\alpha \in S$, $P \stackrel{\alpha}{\longrightarrow} P'$, $Q \stackrel{\alpha}{\longrightarrow} Q'$ and $R' = P' \|_S Q'$. Then $\alpha \in \mathcal{I}(P)$, so $\alpha \notin S \setminus \mathcal{I}(P)$ and thus $\alpha \in Y$. Hence $\theta_Y(Q) \stackrel{\alpha}{\longrightarrow} Q'$. Now $T = \theta_X(P \|_S \theta_Y(Q)) \stackrel{\alpha}{\longrightarrow} P' \|_S Q' = R'$.
  - Let $\alpha \notin S$, $Q \stackrel{\alpha}{\longrightarrow} Q'$ and $R' = P \|_S Q'$. Then $\alpha \in Y$, so $\theta_Y(Q) \stackrel{\alpha}{\longrightarrow} Q'$. Therefore, $P \|_S \theta_Y(Q) \stackrel{\alpha}{\longrightarrow} P \|_S Q'$ and thus $T = \theta_X(P \|_S \theta_Y(Q)) \stackrel{\alpha}{\longrightarrow} P \|_S Q' = R'$.

  Finally, assume $\alpha \in (A \cup \{\text{t}\}) \setminus X$. In that case $P \|_S Q \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Therefore, $Q \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in (X \setminus S) \cup \{\tau\}$, and for all $\beta \in X \cap S \cap \mathcal{I}(P)$, and thus for all $\beta \in Y \cup \{\tau\}$. By Lemma 12, $\theta_Y(Q) =_\mathcal{I} Q$, and hence $P \|_S \theta_Y(Q) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Again, I consider the three rules from Table 1 that could have derived $P \|_S Q \stackrel{\alpha}{\longrightarrow} R'$.
  - Let $\alpha \notin S$, $P \stackrel{\alpha}{\longrightarrow} P'$ and $R' = P' \|_S Q$. Then $P \|_S \theta_Y(Q) \stackrel{\alpha}{\longrightarrow} P' \|_S \theta_Y(Q)$ and thus $T = \theta_X(P \|_S \theta_Y(Q)) \stackrel{\alpha}{\longrightarrow} P' \|_S \theta_Y(Q) =: T'$. By Lemma 12, $\theta_Y(Q) \leftrightarrows Q$. Since $\leftrightarrows$ is a congruence for $\|_S$, it follows that $R' = P' \|_S Q \leftrightarrows P' \|_S \theta_Y(Q) = T'$.
  - Let $\alpha \in S$, $P \stackrel{\alpha}{\longrightarrow} P'$, $Q \stackrel{\alpha}{\longrightarrow} Q'$ and $R' = P' \|_S Q'$. Then $\theta_Y(Q) \stackrel{\alpha}{\longrightarrow} Q'$ and therefore $P \|_S \theta_Y(Q) \stackrel{\alpha}{\longrightarrow} P' \|_S Q'$ and $T = \theta_X(P \|_S \theta_Y(Q)) \stackrel{\alpha}{\longrightarrow} P' \|_S Q' = R'$.
  - Let $\alpha \notin S$, $Q \stackrel{\alpha}{\longrightarrow} Q'$ and $R' = P \|_S Q'$. Then $\theta_Y(Q) \stackrel{\alpha}{\longrightarrow} Q'$, so $P \|_S \theta_Y(Q) \stackrel{\alpha}{\longrightarrow} P \|_S Q'$ and thus $T = \theta_X(P \|_S \theta_Y(Q)) \stackrel{\alpha}{\longrightarrow} P \|_S Q' = R'$.
- Let $R = \theta_X(P \|_S \theta_Y(Q))$ and $T = \theta_X(P \|_S Q)$, for some $Q \in \mathbb{P}$.
  First assume $\alpha = \tau$. Then $Q \stackrel{\tau}{\longrightarrow} Q'$ for some $Q'$ with $R' = \theta_X(P \|_S \theta_Y(Q'))$. Consequently, $T = \theta_X(P \|_S Q) \stackrel{\tau}{\longrightarrow} \theta_X(P \|_S Q') =: T'$ and $R' \, \widetilde{\mathscr{B}} \, T'$.
  Now assume $\alpha \in A \cup \{\text{t}\}$. Then $P \|_S \theta_Y(Q) \stackrel{\alpha}{\longrightarrow} R'$ and either $\alpha \in X$ or $P \|_S \theta_Y(Q) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. In the latter case one obtains $\theta_Y(Q) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in Y \cup \{\tau\}$ (as above), and thus $Q \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in Y \cup \{\tau\}$, that is, $\mathcal{I}(Q) \cap (Y \cup \{\tau\}) = \emptyset$. Furthermore, this implies that $P \|_S Q \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$.
  I consider the three rules from Table 1 that could have derived $P \|_S Q \stackrel{\alpha}{\longrightarrow} R'$.

- Let $\alpha \notin S$, $P \xrightarrow{\alpha} P'$ and $R' = P' \|_S \theta_Y(Q)$. Then $a \notin X$, because $\mathcal{I}(P) \cap X \subseteq S$. Hence $P \|_S \theta_Y(Q) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$, so $\mathcal{I}(Q) \cap (Y \cup \{\tau\}) = \emptyset$. Now $T = \theta_X(P \|_S Q) \xrightarrow{\alpha} P' \|_S Q =: T'$ and $R' \leftrightarrow T'$, using Lemma 12.
- Let $\alpha \in S$, $P \xrightarrow{\alpha} P'$, $\theta_Y(Q) \xrightarrow{\alpha} Q'$ and $R' = P' \|_S Q'$. Then $Q \xrightarrow{\alpha} Q'$. Hence $P \|_S Q \xrightarrow{\alpha} P' \|_S Q'$ and thus $T = \theta_X(P \|_S Q) \xrightarrow{\alpha} P' \|_S Q' = R'$.
- Let $\alpha \notin S$, $\theta_Y(Q) \xrightarrow{\alpha} Q'$ and $R' = P \|_S Q'$. Then $Q \xrightarrow{\alpha} Q'$. Consequently, $P \|_S Q \xrightarrow{\alpha} P \|_S Q'$ and thus $T = \theta_X(P \|_S Q) \xrightarrow{\alpha} P \|_S Q' = R'$. □

**Lemma 3** $\theta_X(\tau_I(P)) \leftrightarrow \theta_X(\tau_I(\theta_{X\cup I}(P)))$.

**Proof:** For given $X$ and $I$, let $\mathscr{B} := Id \cup \{(\theta_X(\tau_I(P)), \theta_X(\tau_I(\theta_{X\cup I}(P)))) \mid P \in \mathbb{P}\}$. It suffices to show that the symmetric closure $\widetilde{\mathscr{B}}$ of $\mathscr{B}$ is a strong bisimulation. So let $R \; \widetilde{\mathscr{B}} \; T$ and $R \xrightarrow{\alpha} R'$ with $\alpha \in A \cup \{\tau, \mathrm{t}\}$. I have to find a $T'$ with $T \xrightarrow{\alpha} T'$ and $R' \; \widetilde{\mathscr{B}} \; T'$.

- The case that $R = T$ is trivial.
- Let $R = \theta_X(\tau_I(P))$ and $T = \theta_X(\tau_I(\theta_{X\cup I}(P)))$, for some $P \in \mathbb{P}$.

  First assume $\alpha = \tau$. Then $\tau_I(P) \xrightarrow{\tau} R''$ for some $R''$ such that $R' = \theta_X(R'')$. Therefore, $P \xrightarrow{\beta} P'$ for some $\beta \in I \cup \{\tau\}$ and some $P'$ with $R'' = \tau_I(P')$. In case $\beta = \tau$, it turns out that $T = \theta_X(\tau_I(\theta_{X\cup I}(P))) \xrightarrow{\tau} \theta_X(\tau_I(\theta_{X\cup I}(P'))) =: T'$. Moreover, $R' \; \mathscr{B} \; T'$. In case $\beta \in I$, $\theta_{X\cup I}(P) \xrightarrow{\beta} P'$, so $\tau_I(\theta_{X\cup I}(P)) \xrightarrow{\tau} \tau_I(P')$ and $T = \theta_X(\tau_I(\theta_{X\cup I}(P))) \xrightarrow{\tau} \theta_X(\tau_I(P')) = R'$.

  Now assume $\alpha \in A \cup \{\mathrm{t}\}$. Then $\tau_I(P) \xrightarrow{\alpha} R'$ and either $\alpha \in X$ or $\tau_I(P) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. It follows that $\alpha \notin I$ and $P \xrightarrow{\alpha} P'$ for some $P'$ with $R' = \tau_I(P')$. Moreover, in case $\alpha \notin X$ one has $P \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup I \cup \{\tau\}$, and hence also $\theta_{X\cup I}(P) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup I \cup \{\tau\}$, and thus $\tau_I(\theta_{X\cup I}(P)) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Now $\theta_{X\cup I}(P) \xrightarrow{\alpha} P'$, so $\tau_I(\theta_{X\cup I}(P)) \xrightarrow{\alpha} \tau_I(P')$ and thus $T = \theta_X(\tau_I(\theta_{X\cup I}(P))) \xrightarrow{\alpha} \tau_I(P') = R'$.
- Let $R = \theta_X(\tau_I(\theta_{X\cup I}(P)))$ and $T = \theta_X(\tau_I(P))$, for some $P \in \mathbb{P}$.

  First assume $\alpha = \tau$. Then $\tau_I(\theta_{X\cup I}(P)) \xrightarrow{\tau} R''$ for some $R''$ such that $R' = \theta_X(R'')$. Therefore, $\theta_{X\cup I}(P) \xrightarrow{\beta} P'$ for some $\beta \in I \cup \{\tau\}$ and some $P'$ with $R'' = \tau_I(P')$. In case $\beta = \tau$, it turns out that $P \xrightarrow{\tau} P''$ for some $P''$ such that $P' = \theta_{X\cup I}(P'')$. So $T = \theta_X(\tau_I(P)) \xrightarrow{\tau} \theta_X(\tau_I(P'')) =: T'$, and $R' \; \widetilde{\mathscr{B}} \; T'$. In case $\beta \in I$, one has $P \xrightarrow{\beta} P'$, so $\tau_I(P) \xrightarrow{\tau} \tau_I(P')$ and $T = \theta_X(\tau_I(P)) \xrightarrow{\tau} \theta_X(\tau_I(P')) = R'$.

  Now assume $\alpha \in A \cup \{\mathrm{t}\}$. Then $\tau_I(\theta_{X\cup I}(P)) \xrightarrow{\alpha} R'$, so $\alpha \notin I$ and $\theta_{X\cup I}(P) \xrightarrow{\alpha} P'$ for some $P'$ such that $R' = \tau_I(P')$. Thus $P \xrightarrow{\alpha} P'$ and either $\alpha \in X$ or $P \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup I \cup \{\tau\}$. In the latter case $\tau_I(P) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Now $\tau_I(P) \xrightarrow{\alpha} \tau_I(P')$ and consequently $T = \theta_X(\tau_I(P)) \xrightarrow{\alpha} \tau_I(P') = R'$. □

**Lemma 4** $\theta_X(\mathcal{R}(P)) \leftrightarrow \theta_X(\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P)))$.

**Proof:** For given $X \subseteq A$ and $\mathcal{R} \subseteq A \times A$, let $\mathscr{B} := Id \cup \{(\theta_X(\mathcal{R}(P)), \theta_X(\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P)))) \mid P \in \mathbb{P}\}$. It suffices to show that the symmetric closure $\widetilde{\mathscr{B}}$ of $\mathscr{B}$ is a strong bisimulation. So let $R \; \widetilde{\mathscr{B}} \; T$ and $R \xrightarrow{\alpha} R'$ with $\alpha \in A \cup \{\tau, \mathrm{t}\}$. I have to find a $T'$ with $T \xrightarrow{\alpha} T'$ and $R' \; \widetilde{\mathscr{B}} \; T'$.

- The case that $R = T$ is trivial.
- Let $R = \theta_X(\mathcal{R}(P))$ and $T = \theta_X(\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P)))$, for some $P \in \mathbb{P}$.

  First assume $\alpha = \tau$. Then $P \xrightarrow{\tau} P'$ for some $P'$ such that $R' = \theta_X(\mathcal{R}(P'))$. Hence $T = \theta_X(\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P))) \xrightarrow{\tau} \theta_X(\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P'))) =: T'$, and $R' \; \mathscr{B} \; T'$.

Now assume $\alpha \in A \cup \{\mathrm{t}\}$. Then $\mathcal{R}(P) \xrightarrow{\alpha} R'$, and either $\alpha \in X$ or $\mathcal{R}(P) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. In the latter case, $P \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in \mathcal{R}^{-1}(X) \cup \{\tau\}$. Moreover, $P \xrightarrow{\gamma} P'$, for some $\gamma$ with $\gamma = \mathrm{t} = \alpha$ or $(\gamma, \alpha) \in \mathcal{R}$, and some $P'$ with $R' = \mathcal{R}(P')$. In case $\alpha \in X$, one has $\gamma \in \mathcal{R}^{-1}(X)$. Therefore, $\theta_{\mathcal{R}^{-1}(X)}(P) \xrightarrow{\gamma} P'$, and thus $\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P)) \xrightarrow{\alpha} \mathcal{R}(P')$.

Either $\alpha \in X$ or $\theta_{\mathcal{R}^{-1}(X)}(P) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in \mathcal{R}^{-1}(X) \cup \{\tau\}$, in which case $\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P)) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Consequently, $T = \theta_X(\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P))) \xrightarrow{\alpha} \mathcal{R}(P') = R'$.

- Let $R = \theta_X(\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P)))$ and $T = \theta_X(\mathcal{R}(P))$, for some $P \in \mathbb{P}$.

  First assume $\alpha = \tau$. Then $P \xrightarrow{\tau} P'$ for some $P'$ such that $R' = \theta_X(\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P')))$. Hence $T = \theta_X(\mathcal{R}(P)) \xrightarrow{\tau} \theta_X(\mathcal{R}(P')) =: T'$, and $R' \widetilde{\mathscr{B}} T'$.

  Now assume $\alpha \in A \cup \{\mathrm{t}\}$. Then $\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P)) \xrightarrow{\alpha} R'$ and either $\alpha \in X$ or $\mathcal{R}(\theta_{\mathcal{R}^{-1}(X)}(P)) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Therefore, $\theta_{\mathcal{R}^{-1}(X)}(P) \xrightarrow{\gamma} P'$ for some $\gamma$ with $\gamma = \mathrm{t} = \alpha$ or $(\gamma, \alpha) \in \mathcal{R}$, and some $P'$ such that $R' = \mathcal{R}(P')$. Hence $P \xrightarrow{\gamma} P'$, and thus $\mathcal{R}(P) \xrightarrow{\alpha} \mathcal{R}(P')$. In case $\alpha \notin X$, one has $\theta_{\mathcal{R}^{-1}(X)}(P) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in \mathcal{R}^{-1}(X) \cup \{\tau\}$, and thus $P \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in \mathcal{R}^{-1}(X) \cup \{\tau\}$, so $\mathcal{R}(P) \stackrel{\beta}{\not\longrightarrow}$ for all $\beta \in X \cup \{\tau\}$. Hence $T = \theta_X(\mathcal{R}(P)) \xrightarrow{\alpha} \mathcal{R}(P') = R'$. □

## C Reducing Strong Reactive Bisimilarity to Strong Bisimilarity

Pohlmann [37] introduces unary operators $\vartheta$ and $\vartheta_X$ 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. Although inspired by my operators $\theta_X$ from Section 4,[1] their semantics is different, and given by the following structural operational rules (for all $X \subseteq A$).

$$\frac{x \xrightarrow{\tau} y}{\vartheta(x) \xrightarrow{\tau} \vartheta(y)} \qquad \frac{}{\vartheta(x) \xrightarrow{\varepsilon_X} \vartheta_X(x)}$$

$$\frac{x \xrightarrow{a} y}{\vartheta_X(x) \xrightarrow{a} \vartheta(y)}\ (a \in X) \qquad \frac{x \xrightarrow{\tau} y}{\vartheta_X(x) \xrightarrow{\tau} \vartheta_X(y)}$$

$$\frac{x \stackrel{\alpha}{\not\longrightarrow} \text{ for all } \alpha \in X \cup \{\tau\}}{\vartheta_X(x) \xrightarrow{\mathrm{t}_\varepsilon} \vartheta(y)} \qquad \frac{x \xrightarrow{\mathrm{t}} y \quad x \stackrel{\alpha}{\not\longrightarrow} \text{ for all } \alpha \in X \cup \{\tau\}}{\vartheta_X(x) \xrightarrow{\mathrm{t}} \vartheta_X(y)}$$

Here 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 $x$. 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.

These rules mirror the clauses of Definition 1 of a strong reactive bisimulation.

- $\tau$-transitions can be performed regardless of the environment,
- triggered environments can stabilise into arbitrary stable environments $X$ for $X \subseteq A$,
- allowed visible transitions can be performed and can trigger a change in the environment,
- $\tau$-transitions cannot be observed by the environment and hence cannot trigger a change,
- if the underlying system is idle, the environment may time-out and become triggered to change,
- if the underlying system is idle, it can perform a t-transition, not observed by the environment.

The main result from [37] reduces strong reactive bisimilarity to strong bisimilarity:

[1] Pohlmann [37] follows the original, 2020, version of this paper; this appendix was added in September 2021.

**Theorem 45** Let $P,Q \in \mathbb{P}$, $X \subseteq A$. Then $P \leftrightarrow_r Q$ iff $\vartheta(P) \leftrightarrow \vartheta(Q)$, and $P \leftrightarrow_r^X Q$ iff $\vartheta_X(P) \leftrightarrow \vartheta_X(Q)$.

**Proof:** If $\mathscr{R}$ is a strong reactive bisimulation, then

$$\mathscr{B} := \{(\vartheta(P), \vartheta(Q)) \mid (P,Q) \in \mathscr{R}\} \cup \{(\vartheta(P), \vartheta(Q)) \mid (P,X,Q) \in \mathscr{R}\}$$

is a strong bisimulation. Moreover,

$$\mathscr{R} := \{(P,Q) \mid \vartheta(P) \leftrightarrow \vartheta(Q)\} \cup \{(P,X,Q) \mid \vartheta_X(P) \leftrightarrow \vartheta_X(Q)\}$$

is a strong reactive bisimulation. Both statements follows directly from the definitions, and they imply the theorem. This proof stems from [37], where it is formalised in Isabelle. □

Another notable result from [37] is a function $\varsigma$ that turns any formula $\varphi$ from my extension of the Hennessy-Milner logic into a formula $\varsigma(\varphi)$ in the regular Hennessy-Milner logic, such that $P \models \varphi$ iff $\vartheta(P) \models \varsigma(\varphi)$ and $P \models_X \varphi$ iff $\vartheta_X(P) \models \varsigma(\varphi)$.

Interestingly, the operators $\vartheta$ and $\vartheta_X$ from [37] can be expressed in terms of (fairly) standard process algebra operators. Define the universal environment $\mathcal{E}$ as the recursive specification

$$\{U = \sum_{X \subseteq A} \varepsilon_X.X\} \;\cup\; \{X = \mathrm{t}_\varepsilon.U + \sum_{a \in X} a.U \mid X \subseteq A\}.$$

In case $A$ is infinite, this requires an infinite choice operator $\sum$, which was not included in the syntax of CCSP$_\mathrm{t}$ used in Section 5. Here $V_\mathcal{E} = \{U\} \cup \{X \mid X \subseteq A\}$ are the bound variables of $\mathcal{E}$. The process $\langle U|\mathcal{E}\rangle$ denotes an environment that is triggered to change, and $\langle X|\mathcal{E}\rangle$ one that allows exactly the actions in $X$. The only actions that $\langle U|\mathcal{E}\rangle$ can do are stabilising into any $\langle X|\mathcal{E}\rangle$. The process $\langle X|\mathcal{E}\rangle$ can either synchronise on any action $a \in X$ or perform a time-out, in both cases returning to the state $\langle U|\mathcal{E}\rangle$.

If we now drop the negative premises from the structural operational rules of the operators $\vartheta_X$, and add a rule $\frac{x \xrightarrow{\mathrm{t}} y}{\vartheta(x) \xrightarrow{\mathrm{t}} \vartheta(y)}$, then $\vartheta(P) \leftrightarrow \langle U|\mathcal{E}\rangle \|_A P$ and $\vartheta_X(P) \leftrightarrow \langle X|\mathcal{E}\rangle \|_A P$. Here the operator $\|_A$ enforces synchronisation on all visible actions $a \in A$, although actions $\varepsilon_X$ and $\mathrm{t}_\varepsilon$ can occur when the environment is ready do do them, and actions $\tau$ and $\mathrm{t}$ can be triggered by just the process $P$. Checking strong bisimilarity between $\vartheta(P)$ and $\langle U|\mathcal{E}\rangle\|_A P$, and between $\vartheta_X(P)$ and $\langle X|\mathcal{E}\rangle\|_A P$, is straightforward.

To obtain the real process $\vartheta(P)$ from $\langle U|\mathcal{E}\rangle \|_A P$, or $\vartheta_X(P)$ from $\langle X|\mathcal{E}\rangle \|_A P$, all one has to do is to inhibit any $\mathrm{t}$- or $\mathrm{t}_\varepsilon$-transition when a transition with a label in $A \cup \{\tau\} \cup \{\varepsilon_X \mid X \subseteq A\}$ is possible. This can be achieved with the priority operator of Baeten, Bergstra & Klop [1]. This unary operator $\Theta$ is parametrised by a partial order $<$ on the set of actions, the *priority* order, and passes through a transition of its argument process only if no transition with a higher priority is possible. Its operational semantic is given by

$$\frac{x \xrightarrow{\alpha} y \quad x \stackrel{\beta}{\not\longrightarrow} \text{ for all } \beta > \alpha}{\Theta(x) \xrightarrow{\alpha} \Theta(y)}\ .$$

For the present application I take $< := \{(\mathrm{t},\alpha), (\mathrm{t}_\varepsilon,\alpha) \mid \alpha \in Act\backslash\{\mathrm{t},\mathrm{t}_\varepsilon\}\}$, thus giving $\mathrm{t}$ and $\mathrm{t}_\varepsilon$ a lower priority than all other actions. This yields the desired properties

$$\vartheta(P) \leftrightarrow \Theta(\langle U|\mathcal{E}\rangle \|_A P) \qquad \text{and} \qquad \vartheta_X(P) \leftrightarrow \Theta(\langle X|\mathcal{E}\rangle \|_A P)\ .$$