The Coarsest Precongruences Respecting Safety and Liveness Properties

Rob van Glabbeek[1,2]

[1] NICTA, Sydney, Australia
[2] School of Computer Sc. and Eng., Univ. of New South Wales, Sydney, Australia

Abstract. This paper characterises the coarsest refinement preorders on labelled transition systems that are precongruences for renaming and partially synchronous interleaving operators, and respect all safety, liveness, and conditional liveness properties, respectively.

1 Introduction

The goal of this paper is to define and characterise certain semantic equivalences $\equiv$ and refinement preorders $\sqsubseteq$ on processes. The idea is that $p \equiv q$ says, essentially, that for practical purposes processes p and q are equally suitable, i.e. one can be replaced for by the other without untoward side effects. Likewise, $p \sqsubseteq q$ says that for all practical purposes under consideration, q is at least as suitable as p, i.e. it will never harm to replace p by q. Thus, one should have that $p \equiv q$ iff both $p \sqsubseteq q$ and $q \sqsubseteq p$.

Naturally, the choice of $\equiv$ and $\sqsubseteq$ depends on how one models a process, and what range of practical purposes one considers. I this paper I restrict myself to one of the most basic process models: *labelled transition systems*. I study processes that merely perform actions a, b, c, ... which themselves are not subject to further investigations. These actions may be instantaneous or durational, but they may not last forever; moreover, in a finite amount of time only finitely many actions can be carried out. I distinguish between *visible* actions, that can be observed by the environment of a process, and whose occurrence can be influenced by this environment, and *invisible* actions, that cannot be observed of influenced. Since there is no need to distinguish different invisible actions, I can just as well consider all of them to be occurrences of the same invisible action, which is traditionally called τ. Furthermore, I abstract from real-time and probabilistic aspects of processes.

This choice of process model already rules out many practical purposes for which one process could be more suitable than another. I can for instance not compare processes on speed, since this is an issue that my process model has already abstracted from. In fact, the only aspects of processes that are captured by such a model and that may matter in practical applications, are the sequences of actions that a process may perform in a, possibly infinite, run, performed in, or in collaboration with, a certain environment. As the invisible action is by definition unobservable, it moreover suffices to consider sequences of *visible*

actions. A sequence of visible actions that a process p may perform is called a *trace* of p; it is a *complete trace* if it is performed during a maximal run of p, one that cannot be further extended. Obviously, the traces of p are completely determined by the complete traces of p, namely as their prefixes.

Based on the considerations above, it is tempting to postulate that the relevant behaviour of a process, as far as discernible in terms of labelled transition systems, is completely determined by its set of complete traces; hence two processes should be equivalent if they have the same competed traces. However, this argument bypasses the role of the environment in influencing the behaviour of a process. Often, one allows the actions a process performs to be synchronisations with the environment, and the environment can influence the course of action of a process by synchronising with some actions but not with others. Therefore, a safe over-approximation of the relevant behaviour of a process is not merely its set of complete traces, but rather its set of complete traces obtained as a function of the environment the process is running in.

In this paper I consider a neutral environment in which all courses of action are possible and the behaviour of a process is indeed determined by its complete traces. All other ways in which the environment may influence the behaviour of a process are given in terms of *contexts* build from other processes and certain composition operators. It could for instance be that in the neutral environment there is no way to tell the difference between processes p and q; maybe because they have the same set of complete traces. However, for a suitable parallel composition operator $\|$ and other process r it may be that there is a manifest practical difference between $p\|r$ and $q\|r$, so that one has $p\|r \not\equiv q\|r$. Now that fact alone is taken to be enough reason to postulate that $p \not\equiv q$. Namely the difference between p and q can be spotted by placing them in a context $\_\|r$. This context can be regarded as an environment in which the behaviours of p and q differ.

Following this programme, a suitable semantic equivalence on processes is defined in terms of two requirements. First of all the behaviour of processes is compared in the neutral environment. This entails isolating a class $\mathcal{C}$ of properties φ of processes that are deemed relevant in a given range of applications. One then requires for $p \equiv q$ to hold that p and q have the same properties from this class:

$$p \equiv q \quad \Rightarrow \quad \forall \varphi \in \mathcal{C}.\ (p \models \varphi \Leftrightarrow q \models \varphi) \tag{1}$$

where $p \models \varphi$ denotes that process p has the property φ. An equivalence $\equiv$ that satisfies this last requirement is said to *respect* or *preserve* the properties in $\mathcal{C}$. The second requirement entails selecting a class $\mathcal{O}$ of useful operators for combining processes. One then requires that for any context $C[\_]$ (such as $\_\|r$) built from operators from $\mathcal{O}$ and arbitrary processes, that

$$p \equiv q \quad \Rightarrow \quad C[p] \equiv C[q]. \tag{2}$$

An equivalence $\equiv$ that satisfies this last requirement is called a *congruence* for $\mathcal{O}$. For the sake of intuition it may help to consider the contrapositive formulation of these implications: if there exists a property φ in $\mathcal{C}$ that holds for p but not for q,

or vice versa, then p and q cannot be considered equivalent. Likewise $C[p] \not\equiv C[q]$ implies $p \not\equiv q$.

These two requirements merely insist that the desired equivalence $\equiv$ does not identify processes that in some context differ on their relevant properties. They are satisfied by many equivalence relations, including the identity relation, that distinguishes all processes. In order to characterise precisely when two systems have the same relevant properties in any relevant context, one takes the *coarsest* equivalence satisfying (1) and (2); the one making the most identifications. This equivalence is called *fully abstract* w.r.t. $\mathcal{C}$ and $\mathcal{O}$. It always exists, and, as is straightforward to check, is characterised by

$$p \equiv q \quad \Leftrightarrow \quad \forall \mathcal{O}\text{-context } C[\_].\, \forall \varphi \in \mathcal{C}.\, (C[p] \models \varphi \Leftrightarrow C[q] \models \varphi).$$

When, for a certain application, the choice of $\mathcal{C}$ and $\mathcal{O}$ is clear, the unique equivalence relation that is fully abstract w.r.t. $\mathcal{C}$ and $\mathcal{O}$ is the right semantic equivalence for that application. However, when the choice of $\mathcal{C}$ and $\mathcal{O}$ is not clear, or when proving results that may be re-used in future applications that may call for extending $\mathcal{C}$ or $\mathcal{O}$, it is better to err on the side of caution, and use equivalences that satisfy (1) and (2) but need not be fully abstract; instead the finest equivalence $\equiv_{\mathit{fine}}$ for which a result $p \equiv_{\mathit{fine}} q$ can be proved is often preferable, because this immediately entails that $p \equiv q$ for any coarser equivalence relation $\equiv$, in particular for an $\equiv$ that may turn out to be fully abstract for some future choice of $\mathcal{C}$ and $\mathcal{O}$. It is for this reason that much actual verification work employes the finest equivalences that lend themselves for verification purposes, such as the various variants of bisimulation equivalence [12], see e.g. [1]. Nevertheless, this paper is devoted to the characterisation of fully abstract equivalences, and preorders, for a few suitable choices of $\mathcal{C}$ and $\mathcal{O}$.

The programme for refinement preorders proceeds along the same lines, but here it is important to distinguish between good and bad properties of processes. The counterpart of (1) is

$$p \sqsubseteq q \quad \Rightarrow \quad \forall \varphi \in \mathcal{G}.\, (p \models \varphi \Rightarrow q \models \varphi) \tag{3}$$

where $\mathcal{G}$ is the set of *good* properties within $\mathcal{C}$, those that for some applications may be required of a process. If this holds, $\sqsubseteq$ *respects* or *preserves* the properties in $\mathcal{G}$. When dealing with *bad* properties, those that in some applications should be avoided, the implication between $p \models \varphi$ and $q \models \varphi$ is oriented in the other direction. Since every bad property φ can be reformulated as a good property $\neg\varphi$, there is no specific need add a variant of (3) for the bad properties. The counterpart of (2) is simply

$$p \sqsubseteq q \quad \Rightarrow \quad C[p] \sqsubseteq C[q]. \tag{4}$$

and a preorder $\sqsubseteq$ that satisfies this requirement is called a *precongruence* for $\mathcal{O}$. Now the preorder that is *fully abstract* w.r.t. $\mathcal{G}$ and $\mathcal{O}$ always exists, and is characterised by

$$p \sqsubseteq q \quad \Leftrightarrow \quad \forall \mathcal{O}\text{-context } C[\_].\, \forall \varphi \in \mathcal{G}.\, (C[p] \models \varphi \Rightarrow C[q] \models \varphi).$$

It is the coarsest precongruence for $\mathcal{O}$ that respects the properties in $\mathcal{G}$. A characterisation of the preorder $\sqsubseteq$ that is fully abstract w.r.t. a certain $\mathcal{G}$ and $\mathcal{O}$ automatically yields a characterisation of the equivalence $\equiv$ that is fully abstract w.r.t. $\mathcal{G}$ and $\mathcal{O}$, as one has $p \equiv q$ iff both $p \sqsubseteq q \wedge q \sqsubseteq p$.

In this paper I will propose three main candidates for the set $\mathcal{G}$ of good properties: *safety properties* in Section 3, *liveness properties* in Section 4 and *conditional liveness properties* in Section 5. For the sake of theoretical completeness I moreover address general *linear time properties* in Section 6.

In Section 2 I will define my model of labelled transition systems and propose a class of $\mathcal{C}$ of operators that appear useful in applications to combine processes. My favourite selection contains

- the *partially synchronous interleaving operator* of CSP [15],
- *abstraction* or *concealment* [3,15]
- and the *state operator* [2],

or any other basis that is equally expressive. With each of the four choices for $\mathcal{G}$ this set of operators determines a fully abstract preorder, which will be characterised in Sections 3, 4, 5 and 6. It turns out that the resulting preorders are somewhat robust under the precise choice of operators for which one imposes a precongruence requirement: the same ones are obtained already without using concealment, and using merely injective renaming instead of the more general state operator. In the other direction, I could just as well have used all operators of CSP.

2 Labelled Transition Systems and a Selection of Composition Operators

Let Σ^* denote the set of finite sequences over a given set Σ, and Σ^∞ the set of infinite ones; $\Sigma^\omega := \Sigma^* \cup \Sigma^\infty$. Write ϵ for the empty sequence, $\sigma\rho$ for the concatenation of sequences $\sigma \in \Sigma^*$ and $\rho \in \Sigma^\omega$, and a for the sequence consisting of the single symbol $a \in \Sigma$. Write $\sigma \leq \rho$ for "σ is a prefix of ρ", i.e. "$\rho = \sigma \vee \exists \nu \in \Sigma^*.\sigma\nu = \rho$", and $\rho < \sigma$ for "$\sigma \leq \rho$ and $\sigma \neq \rho$".

I presuppose an infinite action alphabet A, not containing the *silent* action τ, and set $A_\tau = A \cup \{\tau\}$.

Definition 1 A *labelled transition system* (LTS) is a pair $(\mathbb{P}, \rightarrow)$, where $\mathbb{P}$ is a class of *processes* or *states* and $\rightarrow \subseteq \mathbb{P} \times A_\tau \times \mathbb{P}$ is a set of *transitions*, such that for each $p \in \mathbb{P}$ and $\alpha \in A_\tau$ the class $\{q \in \mathbb{P} \mid (p, \alpha, q) \in \rightarrow\}$ is a set.

Assuming a fixed transition system $(\mathbb{P}, \rightarrow)$, I write $p \xrightarrow{\alpha} q$ for $(p, \alpha, q) \in \rightarrow$; this means that process p can evolve into process q, while performing the action α. The ternary relation $\stackrel{=}{\Longrightarrow} \subseteq \mathbb{P} \times A^* \times \mathbb{P}$ is the least relation satisfying

$$p \stackrel{\epsilon}{\Longrightarrow} p \ , \qquad \frac{p \xrightarrow{\tau} q}{p \stackrel{\epsilon}{\Longrightarrow} q} \ , \qquad \frac{p \xrightarrow{a} q,\ a \neq \tau}{p \stackrel{a}{\Longrightarrow} q} \qquad \text{and} \qquad \frac{p \stackrel{\sigma}{\Longrightarrow} q \stackrel{\rho}{\Longrightarrow} r}{p \stackrel{\sigma\rho}{\Longrightarrow} r} \ .$$

This enables a formalisation of the concepts of traces and complete traces from the introduction.

Definition 2 Let $p \in \mathbb{P}$.

- p is *deterministic* if, for any $\sigma \in A^*$, $p \stackrel{\sigma}{\Longrightarrow} q_1$ and $p \stackrel{\sigma}{\Longrightarrow} q_2$ implies that $q_1 = q_2$ and $q_1 \stackrel{\tau}{\not\longrightarrow} r$.
- p *deadlocks*, notation $p \not\rightarrow$, if there are no $\alpha \in A_\tau$ and $q \in \mathbb{P}$ with $p \stackrel{\alpha}{\longrightarrow} q$.
- p is *locked* if it can never do a visible action, i.e. if $p \stackrel{a}{\Longrightarrow} q$ for no $a \in A$ and $q \in \mathbb{P}$.
- p *diverges*, notation $p\Uparrow$, if there are $p_i \in \mathbb{P}$ for all $i > 0$ such that $p \stackrel{\tau}{\longrightarrow} p_1 \stackrel{\tau}{\longrightarrow} p_2 \stackrel{\tau}{\longrightarrow} \cdots$.
- $a_1a_2a_3 \cdots \in A^\infty$ is an *infinite trace* of p if there are $p_1, p_2, \ldots \in \mathbb{P}$ such that $p \stackrel{a_1}{\Longrightarrow} p_1 \stackrel{a_2}{\Longrightarrow} p_2 \stackrel{a_3}{\Longrightarrow} \cdots$.
- $\mathit{inf}(p)$ denotes the set of infinite traces of p.
- $\mathit{ptraces}(p) := \{\sigma \in A^* \mid \exists q.\ p \stackrel{\sigma}{\Longrightarrow} q\}$ is the set of *partial traces* of p.
- $\mathit{traces}(p) := \mathit{inf}(p) \cup \mathit{ptraces}(p)$ is the set of *traces* of p.
- $\mathit{deadlocks}(p) := \{\sigma \in A^* \mid \exists q.\ p \stackrel{\sigma}{\Longrightarrow} q \not\rightarrow\}$ is the set of *deadlock traces* of p.
- $\mathit{diverg}(p) := \{\sigma \in A^* \mid \exists q.\ p \stackrel{\sigma}{\Longrightarrow} q\Uparrow\}$ is the set of *divergence traces* of p.
- $CT(p) := \mathit{inf}(p) \cup \mathit{diverg}(p) \cup \mathit{deadlocks}(p)$ is the set of *complete traces* of p.

Note that $\mathit{traces}(p) = \{\sigma \in A^\omega \mid \exists \rho \in CT(p).\ \sigma \leq \rho\}$.

To justify that $CT(p)$ is indeed a correct formalisation of the set of complete traces of p, I postulate that in a neutral environment, if a process $p \in \mathbb{P}$ has any outgoing transition $p \stackrel{\alpha}{\longrightarrow} q$, then within a finite amount of time it will do one its outgoing transitions. This is called a *progress property*; it says that a process will continue to make progress if possible.

As explained in the introduction, whether a fully abstract equivalence identifies processes p and q may depend on the existence of a third process r such that $p\|r$ can be distinguished from $q\|r$. When restricting attention to a particular labelled transition system $(\mathbb{P}, \rightarrow)$ it might happen that a perfectly reasonable candidate r happens not to be a member of $\mathbb{P}$, and thus that the conclusion $p \equiv q$ is arrived at solely as a result underpopulation of $\mathbb{P}$. To obtain the most robust notions of equivalence, I therefore assume my LTS to be *universal*, in the sense that up to isomorphism it contains *any* process one can imagine.

Definition 3 An LTS $(\mathbb{P}, \rightarrow_{\mathbb{P}})$ is *universal* if for any other LTS $(\mathbb{Q}, \rightarrow_{\mathbb{Q}})$ there exists an injective mapping $f : \mathbb{Q} \rightarrow \mathbb{P}$, called an *embedding*, such that, for any $q \in \mathbb{Q}$ and $p' \in \mathbb{P}$ one has $f(q) \stackrel{\alpha}{\longrightarrow_{\mathbb{P}}} p'$ iff $p' \in \mathbb{P}$ has the form $f(q')$ for some $q' \in Q$ with $q \stackrel{\alpha}{\longrightarrow_{\mathbb{Q}}} q'$.

The existence of a universal LTS has been established in [7]. Here one needs $\mathbb{P}$ to be a proper class. All preorders $\sqsubseteq$ that I consider in this paper are defined on arbitrary LTSs and have the property that $q \sqsubseteq q' \Leftrightarrow f(p) \sqsubseteq f(q')$, for any embedding f. This means that they are precongruences for isomorphism, and only take into account the future behaviour of processes, i.e. in determining whether $p \sqsubseteq q$ transitions leading to p or q play no rôle. Thus, a definition of such a preorder on a universal LTS, implicitly also defines it on any other LTS.

I will now do a proposal for the set $\mathcal{O}$ that will be my default choice in this paper. It consists of three operators for combining processes that appear useful in practical applications.

$$\frac{p \xrightarrow{\alpha} p'}{p\|_S q \xrightarrow{\alpha} p'\|_S q}\ (\alpha\notin S) \qquad \frac{q \xrightarrow{\alpha} q'}{p\|_S q \xrightarrow{\alpha} p\|_S q'}\ (\alpha\notin S) \qquad \frac{p \xrightarrow{a} p' \quad q \xrightarrow{a} q'}{p\|_S q \xrightarrow{a} p\|_S q'}\ (a\in S)$$

$$\frac{p \xrightarrow{\alpha} p'}{\tau_I(p) \xrightarrow{\alpha} \tau_I(p')}\ (\alpha\notin I) \qquad \frac{p \xrightarrow{a} p'}{\tau_I(p) \xrightarrow{\tau} \tau_I(p')}\ (a\in I)$$

$$\frac{p \xrightarrow{\tau} p'}{\lambda_s^m(p) \xrightarrow{\tau} \lambda_s^m(p')} \qquad \frac{p \xrightarrow{a} p'}{\lambda_s^m(p) \xrightarrow{a(m,s)} \lambda_{s(m,a)}^m(p')}$$

Table 1. *Partially synchronous interleaving, abstraction, and the state operator*

The first is the *partially synchronous interleaving operator* of CSP [15]. It is parametrised with a set $S \subseteq A$ of visible actions on which it synchronises: the composition $p\|_S q$ can perform an action from S only when both p and q perform it. All other actions from p and q are interleaved: whenever one of the two components can perform such an action, so can the composition, while the other component doesn't change its state. Formally, for any choice of $S \subseteq A$, $\|_S : \mathbb{P} \times \mathbb{P} \to \mathbb{P}$ is a binary operator on $\mathbb{P}$ such that a process $p\|_S q$ can make an α-transition iff this can be inferred by the first three rules of Table 1 from the transitions that p and q can make. Here a ranges over A and α over A_τ.

A context $\_\|_S r$ is widely regarded as a plausible way of modelling an environment that partially synchronises with processes under investigation. It is for this reason I include it in $\mathcal{O}$. This argument does not hold for many other process algebraic operators, such as the choice operator $+$ of CCS [12]. This is an example of an operator that is useful for *describing* particular processes, but a context $\_+r$ does not really model a reasonable environment in which one wants to run processes under investigation. For reasons of algebraic convenience, being a precongruence for the $+$ is an optional desideratum of refinement preorders, but it is not such an overriding requirement as being a precongruence for $\|_S$.

The second operator nominated for membership of $\mathcal{O}$ is the unary *abstraction operator* τ_I of ACP_τ [3], also known as the *concealment* operator of CSP [4,15]. This operator models a change in the level of abstraction at which processes are regarded, by reclassifying visible actions as hidden ones. It is parametrised with the set $I \subseteq A$ of visible actions that one chooses to abstract from, and formally defined by the next two rules of Table 1. Abstraction from internal actions by such a mechanism is an essential part of most work on verification in a process algebraic setting, and a context $\tau_I(\_)$ represents a reasonable environment in which to evaluate processes.

My final nominee for the set $\mathcal{O}$ of useful composition operators is the *state operator* λ_s^m of [2]. This unary operator formalises an interface between a process and its environment that is able to rename actions: if its argument process performs an action b, the interface $\lambda_s^m(\_)$ may pass on this action to the environment as c, thereby opening up the possibility of synchronisation with another occurrence of c when using a composed context $\lambda_s^m(\_)\|_S r$. Furthermore, the in-

terface may remember the actions that have been performed to far, and make its renaming behaviour dependent on this history. For instance, if its argument p performs two a-actions in a row, $\lambda_s^m(p)$ may pass these on to the environment as a_1 and a_2, respectively.

The state operator λ_s^m is parametrised with an *interface specification* $m = (\mathcal{S}, \text{ACTION}, \text{EFFECT})$, consisting of set $\mathcal{S}$ of *internal states*, and functions $\text{ACTION} : \mathcal{S} \times A \to A$ and $\text{EFFECT} : \mathcal{S} \times A \to \mathcal{S}$, as well as a *current state* $s \in \mathcal{S}$. Here ACTION is a function that renames actions performed by an argument process p into actions performed by the interface $\lambda_s^m(p)$; the renaming depends on the internal state of the state operator, and thus is of type $\mathcal{S} \times A \to A$. EFFECT specifies the transformation of one internal state of the state operator into another, as triggered by the the encounter of an action of its argument process; it thus is of type $\mathcal{S} \times A \to \mathcal{S}$. Traditionally, one writes $a(m, s)$ for $\text{ACTION}(s, a)$ and $s(m, a)$ for $\text{EFFECT}(s, a)$. So $a(m, s)$ denotes the action a, as modified by the interface m in state s, whereas $s(m, a)$ denotes the internal state s, as modified by the occurrence of action a of the argument process within the scope of the interface m. With this notation, the formal definition of the state operator is given by the last two rules of Table 1.

The special case of a state operator with a singleton set of internal states is known as a *renaming operator*. Renaming operators occur in the languages CCS [12] and CSP [4,15]. Here I denote a renaming operator as λ^m, where the redundant subscript s is omitted, and m trivialises to a function $\text{ACTION} : A \to A$. I speak of an *injective* renaming operator if $\text{ACTION}(a) = \text{ACTION}(b)$ implies $a = b$. For any injective renaming operator λ^m there exists an inverse renaming operator λ^{-m} (not necessarily injective) such that for all $p \in \mathbb{P}$, the process $\lambda^{-m}(\lambda^m(p))$ behaves exactly the same as p—they are equivalent under all notions of equivalence considered in this paper.

3 Safety Properties

A *safety property* [9] is a property that says that

> *something bad will never happen.*

To formulate a canonical safety property, assume that my alphabet of visible actions contains one specific action b, whose occurrence is *bad*. The canonical safety property now says that b **will never happen**.

Definition 4 A process p satisfies the *canonical safety property*, notation $p \models \mathit{safety}(b)$, if no trace of p contains the action b.

To arrive at a general concept of safety property for labelled transition systems, assume that some notion of *bad* is defined. Now, to judge whether a process p satisfies this safety property, one should judge whether p can reach a state in which one would say that something bad had happened. But all observable behaviour of p that is recorded in a labelled transition system until one comes to such a verdict, is the sequence of visible actions performed until that point.

Thus the safety property is completely determined by the set sequences of visible actions that, when performed by p, lead to such a judgement. Therefore one can just as well define the concept of a safety property in terms of such a set.

Definition 5 A *safety property* of processes in an LTS is given by a set $B \subseteq A^*$. A process p *satisfies* this safety property, $p \models \mathit{safety}(B)$, when $\mathit{ptraces}(p) \cap B = \emptyset$.

This formalisation of safety properties is essentially the same as the one in [9] and all subsequent work on safety properties; the only, non-essential, difference is that I work with transition systems in which the transitions are labelled, whereas [9] and most related work deals with state-labelled transition systems.

A property is called *trivial* if it either always holds or always fails. Trivial properties are respected by any equivalence. The sets $B := \emptyset$ and $B := \{\epsilon\}$ specify trivial safety properties.

Theorem 1 A precongruence for the state operator respects every safety property iff it respects the canonical safety property.

Proof: "*Only if*" follows because the canonical safety property is in fact a safety property, namely the one with B being the set of those sequences that contain the action b.
"*If*": I use here a state operator that remembers exactly what sequence of actions has occurred so far. Thus the set of internal states of its interface specification m is A^*, and furthermore $\sigma(m,a) := \sigma a$ for all $\sigma \in A^*$ and $a \in A$. Now given a safety property $B \subseteq A^*$, let $b \in A$ be the special "bad" action, and $d \in A$ be a different "neutral" action. Define $a(m,\sigma) := \begin{cases} b \text{ if } \sigma a \in B \\ d \text{ otherwise.} \end{cases}$
Then $\lambda^m_\epsilon(p) \models \mathit{safety}(b)$ iff $p \models \mathit{safety}(B)$. Thus, if $p \sqsubseteq q$ and $p \models \mathit{safety}(B)$, then $\lambda^m_\epsilon(p) \sqsubseteq \lambda^m_\epsilon(q)$ and $\lambda^m_\epsilon(p) \models \mathit{safety}(b)$. Hence $\lambda^m_\epsilon(q) \models \mathit{safety}(b)$, so $q \models \mathit{safety}(B)$. □

Being locked (see Definition 2) is a safely property, namely with B the set of all sequences over A^* of length 1. It can be understood this way by regarding any occurrence of an action as bad.

Theorem 2 A precongruence for abstraction that respects the property of being locked, respects the canonical safety property.

Proof: Let $\sqsubseteq$ be a precongruence for abstraction that respects the property of being locked, and suppose that $p \sqsubseteq q$. Let $I := A \setminus \{b\}$. Then τ_I is an operator that renames all actions other than b into τ; thus if a process of the form $\tau_I(r)$ ever performs a visible action, it must be b. Now $p \models \mathit{safety}(b) \Leftrightarrow \tau_I(p) \models \mathit{safety}(b) \Leftrightarrow \tau_I(p)$ is locked $\Rightarrow \tau_I(q)$ is locked $\Leftrightarrow \tau_I(q) \models \mathit{safety}(b) \Leftrightarrow q \models \mathit{safety}(b)$. □

By combining Theorems 1 and 2 one obtains:

Corollary 1 A precongruence for abstraction and for the state operator that respects the property of being locked, respects all safety properties.

Theorem 3 Any precongruence for $\mathcal{O}$ that respects a single nontrivial safety property, respects every safety property.

Proof: Let $\sqsubseteq$ be a precongruence for $\mathcal{O}$ that respects $\mathit{safety}(B)$, where $B \subseteq A^*$, $B \neq \emptyset$ and $\epsilon \notin B$. Let $\sigma \in A^*$ and $a \in A$ be such that $\sigma a \in B$, and no prefix $\rho \leq \sigma$ of σ is in B. Let $\mathit{safety}(a)$ be the canonical safety property, but with a playing the role of b. Naturally, Theorem 1 holds for this renamed canonical safety property as well. Hence it suffices to show that $\sqsubseteq$ respects the property $\mathit{safety}(a)$. Let $I := A \setminus \{a\}$. Then τ_I is an operator that renames all actions other than a into τ; thus if a process of the form $\tau_I(s)$ ever performs a visible action, it must be a. Let r_σ be a process with $CT(r) = \{\sigma\}$ and $r_{\sigma a}$ be a process with $CT(r) = \{\sigma a\}$. Then, for any choice of $s \in \mathbb{P}$, $(\tau_I(s)\|_\emptyset r_\sigma)\|_A r_{\sigma a}$ is a process all of whose traces are prefixes of σa, with $\sigma a \in \mathit{ptraces}((\tau_I(s)\|_\emptyset r_\sigma)\|_A r_{\sigma a})$ iff $a \in \mathit{ptraces}(\tau_I(s))$, which is the case iff $s \not\models \mathit{safety}(a)$. Suppose $p \sqsubseteq q$. Then $(\tau_I(p)\|_\emptyset r_\sigma)\|_A r_{\sigma a} \sqsubseteq (\tau_I(q)\|_\emptyset r_\sigma)\|_A r_{\sigma a}$ and

$$p \models \mathit{safety}(a) \Leftrightarrow (\tau_I(p)\|_\emptyset r_\sigma)\|_A r_{\sigma a} \models \mathit{safety}(B) \Rightarrow$$
$$(\tau_I(q)\|_\emptyset r_\sigma)\|_A r_{\sigma a} \models \mathit{safety}(B) \Leftrightarrow q \models \mathit{safety}(a). \qquad \square$$

Let $\sqsubseteq_{\mathit{safety}}$ denote the preorder that is fully abstract w.r.t. the class of safety properties and $\mathcal{O}$. The following, well-known, theorem characterises this preorder as *reverse partial trace inclusion*.

Theorem 4 $p \sqsubseteq_{\mathit{safety}} q \Leftrightarrow \mathit{ptraces}(p) \supseteq \mathit{ptraces}(q)$.

Proof: Define reverse partial trace inclusion, $\sqsubseteq_T^{-1}$, by $p \sqsubseteq_T^{-1} q$ iff $\mathit{ptraces}(p) \supseteq \mathit{ptraces}(q)$.
"$\Leftarrow$": It suffices to establish that $\sqsubseteq_T^{-1}$ is a precongruence for $\mathcal{O}$ that respects all safety properties.

That $\sqsubseteq_T^{-1}$ is a precongruence for $\mathcal{O}$ follows immediately from the following observations:

$$\begin{aligned}
\mathit{ptraces}(p\|_S q) &= \{\sigma \in \nu\|_S \xi \mid \nu \in \mathit{ptraces}(p) \wedge \xi \in \mathit{ptraces}(q)\} \\
\mathit{ptraces}(\tau_I(p)) &= \{\tau_I(\sigma) \mid \sigma \in \mathit{ptraces}(p)\} \\
\mathit{ptraces}(\lambda_s^m(p)) &= \{\lambda_s^m(\sigma) \mid \sigma \in \mathit{ptraces}(p)\}.
\end{aligned}$$

Here $\nu\|_S\xi$ denotes the set of sequences of actions for which is it possible to mark each action occurrence as *left*, *right* or both, obeying the restriction that an occurrence of action a is marked both *left* and *right* iff $a \in S$, such that the subsequence of all *left*-labelled action occurrences is ν and the subsequence of all *right*-labelled action occurrences is ξ. Furthermore, the operators τ_I and λ_s^m on A^* are uniquely determined by

$$\begin{aligned}
\tau_I(\epsilon) &= \epsilon \qquad & \tau_I(a\sigma) &= \begin{cases} \tau_I(\sigma) & \text{if } a \in I \\ a\tau_I(\sigma) & \text{otherwise} \end{cases} \\
\lambda_s^m(\epsilon) &= \epsilon \qquad & \lambda_s^m(a\sigma) &= a(m,s)\lambda_{s(m,a)}^m(\sigma).
\end{aligned}$$

To show that $\sqsubseteq_T^{-1}$ respects all safety properties, let $B \subseteq A^*$, $p \sqsubseteq_T^{-1} q$, and suppose $p \models \mathit{safety}(B)$. Then $\mathit{ptraces}(q) \subseteq \mathit{ptraces}(p)$ and $\mathit{ptraces}(p) \cap B = \emptyset$. Thus $\mathit{ptraces}(q) \cap B = \emptyset$, i.e. $q \models \mathit{safety}(B)$, which had to be shown.

"⇒": Let $\sqsubseteq$ be any precongruence for $\mathcal{O}$ that respects all safety properties, and suppose $p \sqsubseteq q$. I have to establish that $p \sqsubseteq_T^{-1} q$. Let $B := A^* \setminus \mathit{ptraces}(p)$. Then $p \models \mathit{safety}(B)$. Thus $q \models \mathit{safety}(B)$, i.e. $\mathit{ptraces}(q) \cap (A^* \setminus \mathit{ptraces}(p)) = \emptyset$. This yields $\mathit{ptraces}(q) \subseteq \mathit{ptraces}(p)$. □

The above characterisation as reverse partial trace inclusion of the coarsest congruence for $\mathcal{O}$ that respects all safety properties, is rather robust under the choice of $\mathcal{O}$. It holds already for the empty class of operators, and it remains true when adding in all operators of CSP [4], CCS [12] or ACP_τ [3], as $\sqsubseteq_T^{-1}$ is known to be a precongruence for all of them.

By Theorem 3, the characterisation also remains valid when requiring respect for one arbitrary safety property only, instead of all of them, but to this end all three operators of $\mathcal{O}$ are needed. If we just retain the state operator, by Theorem 1 it suffices to require respect for the canonical safety property only.

4 Liveness Properties

A *liveness property* [9] is a property that says that

something good will eventually happen.

To formulate a canonical liveness property, assume that the alphabet A contains one specific action g, whose occurrence is *good*. The canonical liveness property now says that g **will eventually happen**.

Definition 6 A process p satisfies the *canonical liveness property*, notation $p \models \mathit{liveness}(g)$, if every complete trace of p contains the action g.

To arrive at a general concept of liveness property for labelled transition systems, assume that some notion of *good* is defined. Now, to judge whether a process p satisfies this liveness property, one should judge whether p can reach a state in which one would say that something good had happened. But all observable behaviour of p that is recorded in a labelled transition system until one comes to such a verdict, is the sequence of visible actions performed until that point. Thus the liveness property is completely determined by the set sequences of visible actions that, when performed by p, lead to such a judgement. Therefore one can just as well define the concept of a liveness property in terms of such a set.

Definition 7 A *liveness property* of processes in an LTS is given by a set $G \subseteq A^*$. A process p *satisfies* this liveness property, notation $p \models \mathit{liveness}(G)$, when each complete trace of p has a prefix in G.

This formalisation of liveness properties is essentially different from the one in [9] and most subsequent work on liveness properties; this point is discussed in Section 6.

Just as for safety properties, the sets $G := \emptyset$ and $G := \{\epsilon\}$ specify trivial liveness properties.

Theorem 5 A precongruence for the state operator respects every liveness property iff it respects the canonical liveness property.

Proof: Just like the proof of Theorem 1. □

A process p has the *initial progress* property if it cannot immediately diverge or deadlock, i.e. if $\epsilon \notin \mathit{diverg}(p) \cup \mathit{deadlocks}(p)$. This is a liveness property, namely with G the set of all sequences over A^* of length 1. It can be understood this way by regarding any occurrence of an action as good.

Theorem 6 A precongruence for abstraction that respects the initial progress property, respects the canonical liveness property.

Proof: Just like the proof of Theorem 2. □

By combining Theorems 5 and 6 one obtains:

Corollary 2 A precongruence for abstraction and for the state operator that respects the initial progress property, respects all liveness properties.

Conjecture 1 Any precongruence for $\mathcal{O}$ that respects a single nontrivial liveness property, respects every liveness property.

Let $\sqsubseteq_{\mathit{liveness}}$ denote the preorder that is fully abstract w.r.t. the class of liveness properties and $\mathcal{O}$. I will proceed to characterise $\sqsubseteq_{\mathit{liveness}}$ as the preorder $\sqsubseteq^{\perp}_{\mathit{FDI}}$ based on failures, divergences and infinite traces that is also used in the work on CSP [15]. *Failures* of a process p are defined below; they are pairs $\langle \sigma, X \rangle$ such that p can perform the sequence of visible actions σ and then reach a state in which no further progress can be made in case the environment allows only those visible actions to occur that are listed in X. The preorder $\sqsubseteq^{\perp}_{\mathit{FDI}}$ does not take into account any information about the behaviour of processes that can be thought of taking place after a divergence. One of the ways to erase this information from the set of failures, divergences and infinite traces of a process is by means of *flooding*. Flooded sets of failures, divergences and infinite traces are indicated by the subscript $_{\perp}$.

Definition 8 Let $p \in \mathbb{P}$.
- $\mathit{initials}(p) := \{\alpha \in A_\tau \mid \exists q.\ p \xrightarrow{\alpha} q\}$.
- $\mathit{failures}(p) := \{\langle \sigma, X \rangle \in A^* \times \mathcal{P}(A) \mid \exists q.\ p \stackrel{\sigma}{\Longrightarrow} q \wedge \mathit{initials}(q) \cap (X \cup \{\tau\}) = \emptyset\}$.
- $\mathit{diverg}_{\perp}(p) := \{\sigma\rho \mid \sigma \in \mathit{diverg}(p) \wedge \rho \in A^*\}$.
- $\mathit{inf}_{\perp}(p) := \{\sigma\rho \mid \sigma \in \mathit{diverg}(p) \wedge \rho \in A^\infty\}$.
- $\mathit{failures}_{\perp}(p) := \{\langle \sigma\rho, X \rangle \mid \sigma \in \mathit{diverg}(p) \wedge \rho \in A^* \wedge X \subseteq A\}$.

So $\mathit{deadlocks}(p) = \{\sigma \mid \langle \sigma, A \rangle \in \mathit{failures}(p)\}$
and $\mathit{ptraces}(p) = \mathit{diverg}(p) \cup \{\sigma \mid \langle \sigma, \emptyset \rangle \in \mathit{failures}(p)\}$.

Theorem 7 $p \sqsubseteq_{\mathit{liveness}} q \quad \Leftrightarrow \quad \begin{array}{rcl} \mathit{diverg}_{\perp}(p) & \supseteq & \mathit{diverg}_{\perp}(q) \wedge \\ \mathit{inf}_{\perp}(p) & \supseteq & \mathit{inf}_{\perp}(q) \wedge \\ \mathit{failures}_{\perp}(p) & \supseteq & \mathit{failures}_{\perp}(q). \end{array}$

Proof: Let $\sqsubseteq^{\perp}_{FDI}$ be the preorder defined by: $p \sqsubseteq^{\perp}_{FDI} q$ iff the right-hand side of Theorem 7 holds.
"$\Leftarrow$": It suffices to establish that $\sqsubseteq^{\perp}_{FDI}$ is a liveness respecting precongruence.

To show that $\sqsubseteq^{\perp}_{FDI}$ respects liveness, let $G \subseteq A^*$, $p \sqsubseteq^{\perp}_{FDI} q$, and suppose $p \models \mathit{liveness}(G)$. I need to show that $q \models \mathit{liveness}(G)$. So suppose $\sigma \in CT(q)$. Then either $\sigma \in \mathit{diverg}(g) \subseteq \mathit{diverg}_\perp(g) \subseteq \mathit{diverg}_\perp(p)$ or $\sigma \in \mathit{inf}(q) \subseteq \mathit{inf}_\perp(q) \subseteq \mathit{inf}_\perp(p)$ or $\langle \sigma, A\rangle \in \mathit{failures}(q) \subseteq \mathit{failures}_\perp(q) \subseteq \mathit{failures}_\perp(q)$. In the first case $\rho \in \mathit{diverg}(p) \subseteq CT(p)$ for some $\rho \leq \sigma$; in the second case either $\sigma \in \mathit{inf}(p) \subseteq CT(p)$ or $\rho \in \mathit{diverg}(p) \subseteq CT(p)$ for some $\rho < \sigma$; and in the third case either $\langle \sigma, A\rangle \in \mathit{failures}(p)$ or $\rho \in \mathit{diverg}(p) \subseteq CT(p)$ for some $\rho \leq \sigma$. In all three cases $\rho \in CT(p)$ for some $\rho \leq \sigma$. Since $p \models \mathit{liveness}(G)$, there must be a $\nu \leq \rho$ with $\nu \in G$. As $\nu \leq \sigma$ it follows that $q \models \mathit{liveness}(G)$.

That $\sqsubseteq^{\perp}_{FDI}$ is a precongruence for $\|_S$ and τ_I has been established in [15] by means of the following observations:

$$\begin{aligned}
\mathit{diverg}_\perp(p\|_S q) &= \{\sigma\rho \mid \exists \langle \nu, X\rangle \in \mathit{failures}_\perp(p), \xi \in \mathit{diverg}_\perp(q).\\
&\qquad\qquad \sigma \in \nu\|_S\xi \wedge \rho \in A^*\}\\
&\quad \cup \{\sigma\rho \mid \exists \nu \in \mathit{diverg}_\perp(p), \langle \xi, X\rangle \in \mathit{failures}_\perp(q).\\
&\qquad\qquad \sigma \in \nu\|_S\xi \wedge \rho \in A^*\}\\
\mathit{inf}_\perp(p\|_S q) &= \{\sigma \mid \exists \nu \in \mathit{inf}_\perp(p), \xi \in \mathit{inf}_\perp(q).\ \sigma \in \nu\|_S\xi\} \cup\\
&\quad \{\sigma \mid \exists \langle \nu, X\rangle \in \mathit{failures}_\perp(p), \xi \in \mathit{inf}_\perp(q).\ \sigma \in \nu\|_S\xi\} \cup\\
&\quad \{\sigma \mid \exists \nu \in \mathit{inf}_\perp(p), \langle \xi, X\rangle \in \mathit{failures}_\perp(q).\ \sigma \in \nu\|_S\xi\} \cup\\
&\quad \{\sigma\rho \mid \sigma \in \mathit{diverg}_\perp(p\|_S q) \wedge \rho \in A^\infty\}\\
\mathit{failures}_\perp(p\|_S q) &= \{\langle \sigma, X \cup Y\rangle \mid \exists \langle \nu, X\rangle \in \mathit{failures}_\perp(p), \langle \xi, Y\rangle \in \mathit{failures}_\perp(q).\\
&\qquad\qquad X \setminus S = Y \setminus S \wedge \sigma \in \nu\|_S\xi\}\\
&\quad \cup \{\langle \sigma, X\rangle \mid \sigma \in \mathit{diverg}_\perp(p\|_S q) \wedge X \subseteq A\}.\\
\mathit{diverg}_\perp(\tau_I(p)) &= \{\tau_I(\sigma)\rho \mid \tau_I(\sigma), \rho \in A^* \wedge \sigma \in \mathit{inf}_\perp(p) \cup \mathit{diverg}_\perp(p)\}\\
\mathit{inf}_\perp(\tau_I(p)) &= \{\tau_I(\sigma) \mid \tau_I(\sigma) \in A^\infty \wedge \sigma \in \mathit{inf}_\perp(p)\}\\
&\quad \cup \{\sigma\rho \mid \sigma \in \mathit{diverg}_\perp(\tau_I(p)) \wedge \rho \in A^\infty\}\\
\mathit{failures}_\perp(\tau_I(p)) &= \{\langle \tau_I(\sigma), X\rangle \mid \langle \sigma, X \cup I\rangle \in \mathit{failures}_\perp(p)\}\\
&\quad \cup \{\langle \sigma, X\rangle \mid \sigma \in \mathit{diverg}_\perp(\tau_I(p)) \wedge X \subseteq A\}.
\end{aligned}$$

Here $\tau_I(\sigma)$ for $\sigma \in A^\infty$ is the supremum, w.r.t. the prefix order $\leq$ on A^ω, of the set $\{\tau_I(\rho) \mid \rho < \sigma\}$.

Likewise, $\sqsubseteq^{\perp}_{FDI}$ is a congruence for λ^m_s:

$$\begin{aligned}
\mathit{diverg}_\perp(\lambda^m_s(p)) &= \{\lambda^m_s(\sigma)\rho \mid \sigma \in \mathit{diverg}_\perp(p) \wedge \rho \in A^*\}\\
\mathit{inf}_\perp(\lambda^m_s(p)) &= \{\lambda^m_s(\sigma) \mid \sigma \in \mathit{inf}_\perp(p)\}\\
&\quad \cup \{\sigma\rho \mid \sigma \in \mathit{diverg}_\perp(\lambda^m_s(p)) \wedge \rho \in A^\infty\}\\
\mathit{failures}_\perp(\lambda^m_s(p)) &= \{\langle \lambda^m_s(\sigma), X\rangle \mid \langle \sigma, \lambda^{-m}_s(X)\rangle \in \mathit{failures}_\perp(p)\}\\
&\quad \cup \{\langle \sigma, X\rangle \mid \sigma \in \mathit{diverg}_\perp(\lambda^m_s(p)) \wedge X \subseteq A\}.
\end{aligned}$$

Here $\lambda^{-m}_s(X) := \{a \in A \mid a(m,s) \in X\}$.

"$\Rightarrow$": Let $\sqsubseteq$ be any liveness respecting precongruence, and suppose $p \sqsubseteq q$. I have to establish that $p \sqsubseteq^{\perp}_{FDI} q$. W.l.o.g. I may assume that neither p nor q has any trace containing the action g. For let λ^m be an injective renaming operator such that g is not in the image of λ^m. Then $\lambda^m(p) \sqsubseteq \lambda^m(q)$. Suppose one can

establish $\lambda^m(p) \sqsubseteq^{\perp}_{FDI} \lambda^m(q)$. Since $\sqsubseteq^{\perp}_{FDI}$ is a precongruence for renaming, this yields $p \equiv^{\perp}_{FDI} \lambda^{-m}(\lambda^m(p)) \sqsubseteq^{\perp}_{FDI} \lambda^{-m}(\lambda^m(p)) \equiv^{\perp}_{FDI} q$.

Suppose $\mathit{diverg}_{\perp}(p) \not\supseteq \mathit{diverg}_{\perp}(q)$; say $\sigma \in \mathit{diverg}_{\perp}(q) \setminus \mathit{diverg}_{\perp}(p)$. So there is no $\rho \leq \sigma$ with $\rho \in \mathit{diverg}(p)$. Let r be a deterministic process such that $CT(r) = \{\rho g \mid \rho \leq \sigma\}$. Then each complete trace of $p \|^g r$ contains g. Here I write $\|^g$ for $\|_{A \setminus \{g\}}$, the interleaving operator that synchronises on all visible actions except g. As $\sqsubseteq$ is a precongruence, $p \sqsubseteq q$ implies $p\|^g r \sqsubseteq q\|^g r$, and since $\sqsubseteq$ respects the canonical liveness property, I obtain that each complete trace of $q\|^g r$ must contain g. However, $\rho \in \mathit{diverg}(q)$ for some $\rho \leq \sigma$. So $\rho \in CT(q\|^g r)$, although ρ does not contain g.

Suppose $\mathit{inf}_{\perp}(p) \not\supseteq \mathit{inf}_{\perp}(q)$; say $\sigma \in \mathit{inf}_{\perp}(q) \setminus \mathit{inf}_{\perp}(p)$. So $\sigma \notin \mathit{inf}(p)$ and there is no $\rho < \sigma$ with $\rho \in \mathit{diverg}(p)$. Let r be a deterministic process such that $CT(r) = \{\rho g \mid \rho < \sigma\} \cup \{\sigma\}$. Then each complete trace of $p\|^g r$ contains g. As $\sqsubseteq$ is a precongruence, $p \sqsubseteq q$ implies $p\|^g r \sqsubseteq q\|^g r$, and since $\sqsubseteq$ respects the canonical liveness property, I obtain that each complete trace of $q\|^g r$ must contain g. However, either $\sigma \in \mathit{inf}(q)$ or $\rho \in \mathit{diverg}(q)$ for some $\rho < \sigma$. So either $\sigma \in CT(q\|^g r)$ or $\rho \in CT(q\|^g r)$, and neither σ nor ρ contains g.

Suppose $\mathit{failures}_{\perp}(p) \not\supseteq \mathit{failures}_{\perp}(q)$; say $\langle \sigma, X\rangle \in \mathit{failures}_{\perp}(q) \setminus \mathit{failures}_{\perp}(p)$. So $\langle \sigma, X\rangle \notin \mathit{failures}(p)$ and there is no $\rho \leq \sigma$ with $\rho \in \mathit{diverg}(p)$. Let r be a deterministic process with $CT(r) = \{\rho g \mid \rho < \sigma\} \cup \{\sigma a \mid a \in X\}$, and consider the liveness property given by $G := \{\rho g \mid \rho < \sigma\} \cup \{\sigma a \mid a \in X\}$. Then $p\|^g r \models \mathit{liveness}(G)$. As $\sqsubseteq$ is a precongruence, $p \sqsubseteq q$ implies $p\|^g r \sqsubseteq q\|^g r$, and since $\sqsubseteq$ respects liveness properties, also $q\|^g r \models \mathit{liveness}(G)$. However, either $\langle \sigma, X\rangle \in \mathit{failures}(q)$ or there is an $\rho \leq \sigma$ with $\rho \in \mathit{diverg}(q)$. So either $\sigma \in CT(q\|^g r)$ or $\rho \in CT(q\|^g r)$ for some $\rho \leq \sigma$, contradicting that $q\|^g r \models \mathit{liveness}(G)$. □

The standard refinement preorder used in CSP is in fact the *failures-divergences* preorder $\sqsubseteq_{FD}$, defined exactly like $\sqsubseteq^{\perp}_{FDI}$, but abstracting from the infinite traces. As remarked in [15], this can be done because in CSP one normally restricts attention to processes p with the property that for any $\sigma \in A^*$ either $\sigma \in \mathit{diverg}_{\perp}(p)$ or there are only finitely many processes q with $p \stackrel{\sigma}{\Longrightarrow} q$. For such processes the set $\mathit{inf}_{\perp}(p)$ is, with Königs Lemma, completely determined by $\mathit{failures}_{\perp}(p)$ and $\mathit{diverg}_{\perp}(p)$, and thus need not be explicitly recorded. When extending CSP to processes not having this property, the component $\mathit{inf}_{\perp}$ should be added to the semantics of processes [15]. In fact, $\sqsubseteq^{\perp}_{FDI}$ is the coarsest precongruence for $\mathcal{O}$ contained in $\sqsubseteq_{FD}$: if p, q and r are the processes used in the $\mathit{inf}_{\perp}$-case of the above proof, and $I := A \setminus \{g\}$, then $\epsilon \in \mathit{diverg}_{\perp}(\tau_I(q\|^g r)) \setminus \mathit{diverg}_{\perp}(\tau_I(p\|^g r))$.

The above characterisation as $\sqsubseteq^{\perp}_{FDI}$ of the coarsest congruence for $\mathcal{O}$ that respects all liveness properties, is somewhat robust under the choice of $\mathcal{O}$. It holds already for with just $\|^g$ and injective renaming (for these are the only two operators that are just in the proof), and it remains true when adding in all operators of CSP [4], as $\sqsubseteq^{\perp}_{FDI}$ is known to be a precongruence for all of them [15].

By Corollary 2, the above characterisation also remains valid when requiring respect for the initial progress property only, but to this end all three operators of $\mathcal{O}$ are needed. This result has in essence been obtained already by Bill Roscoe

in [15]. The state operator does not feature in [15]; its rôle in this full abstraction result is taken over by a renaming operator that allows renaming an action a into a choice between two actions b and c. When ignoring this difference in syntax, Theorem 7 can be obtained as an immediate corollary of Corollary 2 and that result. The main reason for using the above proof instead is to show that the concealment or abstraction operator is not needed here.

By Theorem 5, $\sqsubseteq^{\perp}_{FDI}$ is even fully abstract w.r.t. the partially synchronous interleaving and state operators, and the canonical liveness property. This result, like the full abstraction result of [15], does not hold without the state operator, or something equally powerful, even if renaming and abstraction is allowed to be used. Namely, as pointed out by Antti Puhakka [13], one would fail to distinguish the following two processes:

5 Conditional Liveness Properties

Figure 1 presents two processes that have the same liveness properties in any

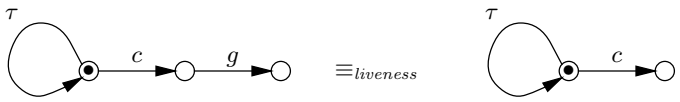

Fig. 1. *Two processes with the same liveness properties but different conditional liveness properties*

CSP-context. The fact that only the left-hand process *can* do something good doesn't matter here, as neither of the two processes is *guaranteed* to do something good: they may never proceed beyond their initial τ-loops. Nevertheless, from a practical point of view, the difference between these two processes may be enormous. It could be that the action c comes with a huge cost, that is only worth making when something good happens afterwards. Only the right-hand side process is able to incur the cost without any benefits, and for this reason it lacks an important property that the left-hand process has. I call such properties *conditional liveness properties* [6,11]. A *conditional liveness property* is a property that says that

under certain conditions something good will eventually happen.

To formulate a canonical conditional liveness property, assume that the alphabet A contains two specific action c and g, where the occurrence of c is the condition, and the occurrence of g is *good*. The canonical conditional liveness property now says that **if** c **occurs then** g **will eventually happen**.

Definition 9 A process p satisfies the *canonical conditional liveness property*, notation $p \models \mathit{liveness}_c(g)$, if every complete trace of p that contains the action c also contains the action g.

To arrive at a general concept of conditional liveness property for labelled transition systems, assume that some condition, and some notion of *good* is defined. Now, to judge whether a process p satisfies this conditional liveness property, one should judge first of all in which states the condition is fulfilled. All observable behaviour of p that is recorded in a labelled transition system until one comes to such a verdict, is the sequence of visible actions performed until that point. Thus the condition is completely determined by the set sequences of visible actions that, when performed by p, lead to such a judgement. Next one should judge whether p can reach a state in which one would say that something good had happened. Again, this judgement can be expressed in terms of the sequences of visible actions that lead to such a state.

Definition 10 A *conditional liveness property* of processes in an LTS is given by two sets $C, G \subseteq A^*$. A process p *satisfies* this conditional liveness property, notation $p \models \mathit{liveness}_C(G)$, when each complete trace of p that has a prefix in C, also has prefix in G.

For the sake of added generality, one could make the notion of success dependent on the particular sequence of actions that fulfilled the condition. This would make G a function from C to $\mathcal{P}(A^*)$ and the requirement would be that each complete trace of p that has a prefix $\sigma \in C$, also has prefix in $G(\sigma)$. However, such a generalised conditional liveness property can be expressed as a conjunction of standard ones, and a preorder that respects a given collection of properties also respects their conjunction.

Theorem 8 A precongruence for the state operator respects every conditional liveness property iff it respects the canonical conditional liveness property.

Proof: "*Only if*" follows because the canonical conditional liveness property is in fact a conditional liveness property, namely the one with C being the set of those sequences that contain the action c, and G the set of those sequences that contain the action g.
"*If*": Again I use a state operator that remembers exactly what sequence of actions has occurred so far. Thus the set of internal states of its interface specification m is A^*, and $\sigma(m, a) := \sigma a$ for all $\sigma \in A^*$ and $a \in A$. Note that the properties $\mathit{liveness}_C(G)$ and $\mathit{liveness}_{C \setminus G}(G)$ are satisfied by the same processes, so w.l.o.g. I may restrict attention to properties $\mathit{liveness}_C(G)$ with $C \cap G = \emptyset$. Given such a property, define

$$a(m, \sigma) := \begin{cases} c \text{ if } \sigma a \in C \\ g \text{ if } \sigma a \in G \\ d \text{ otherwise.} \end{cases}$$

Then $\lambda_\epsilon^m(p) \models \mathit{liveness}_c(g)$ iff $p \models \mathit{liveness}_C(G)$. Thus, if $p \sqsubseteq q$ and $p \models \mathit{liveness}_C(G)$, then $\lambda_\epsilon^m(p) \sqsubseteq \lambda_\epsilon^m(q)$ and $\lambda_\epsilon^m(p) \models \mathit{liveness}_c(g)$. Hence $\lambda_\epsilon^m(q) \models \mathit{liveness}_c(g)$, so $q \models \mathit{liveness}_C(G)$. □

An element $\sigma \notin \mathit{diverg}(p) \cup \mathit{deadlocks}(p)$ is called a *deadlock/divergence trace* of a process p. For any $\sigma \in A^*$, not having a deadlock/divergence trace σ is a conditional liveness property, namely with $C := \{\sigma\}$ and $G := \{\sigma a \mid a \in A\}$. Using similar techniques as for Corollary 1, one can establish:

Corollary 3 A precongruence for abstraction and for the state operator that respects the property of having no deadlock/divergence trace c, respects all liveness properties.

Let $\sqsubseteq_{\mathit{cond.\ liveness}}$ denote the preorder that is fully abstract w.r.t. the class of conditional liveness properties and $\mathcal{O}$. Furthermore, write $\sqsubseteq_{d/d}$ for the coarsest precongruence for $\mathcal{O}$ such that $q \sqsubseteq_{d/d} p$ implies $\mathit{deadlocks}(q) \cup \mathit{diverg}(q) \subseteq \mathit{deadlocks}(p) \cup \mathit{diverg}(p)$.

Corollary 4 $p \sqsubseteq_{\mathit{cond.\ liveness}} q$ iff $q \sqsubseteq_{d/d} p$.

Proof: "*If*" follows immediately from Corollary 3. "*Only if*" follows from the observation that the absence of any deadlock/divergence trace σ is a conditional liveness property. □

Antti Puhakka [13] has given a characterisation of the coarsest congruence that preserves deadlock/divergence traces, $\equiv_{d/d}$. His arguments easily extend to a characterisation of $\sqsubseteq_{d/d}$ and hence, using Corollary 4, of $\sqsubseteq_{\mathit{cond.\ liveness}}$. Below I will give a direct proof of the same result. It shows that this characterisation is already valid when merely requiring the precongruence property for $\|_S$ and injective renaming.

As for $\sqsubseteq_{\mathit{liveness}}$, the characterisation of $\sqsubseteq_{\mathit{cond.\ liveness}}$ is in terms of failures, divergences and infinite traces, and again some information needs to be erased, but less than in the case of $\sqsubseteq_{\mathit{liveness}}$. This time we need to forget about failures $\langle \sigma, X\rangle \in \mathit{failures}(p)$ such that $\sigma \in \mathit{diverg}(p)$, and about infinite traces of p that have arbitrary long prefixes in $\mathit{diverg}(p)$. In [13] this is achieved by removal of such failures and infinite traces; here, in order to stress the similarity with the refinement preorder of CSP, I equivalently apply the method of flooding.

Definition 11 Let $p \in \mathbb{P}$.
- $\mathit{inf}_d(p) := \mathit{inf}(p) \cup \{\sigma \in A^\infty \mid \forall \rho < \sigma \exists \nu \in \mathit{diverg}(p).\ \rho \leq \nu < \sigma\}$.
- $\mathit{failures}_d(p) := \mathit{failures}(p) \cup \{\langle \sigma, X\rangle \mid \sigma \in \mathit{diverg}(p) \wedge X \subseteq A\}$.

Theorem 9 $p \sqsubseteq_{\mathit{cond.\ liveness}} q \;\Leftrightarrow\; \begin{array}{rcl} \mathit{diverg}(p) & \supseteq & \mathit{diverg}(q) \wedge \\ \mathit{inf}_d(p) & \supseteq & \mathit{inf}_d(q) \wedge \\ \mathit{failures}_d(p) & \supseteq & \mathit{failures}_d(q). \end{array}$

Proof: Let $\sqsubseteq^d_{FDI}$ be the preorder defined by: $p \sqsubseteq^d_{FDI} q$ iff the right-hand side of Theorem 7 holds.
"$\Leftarrow$": It suffices to establish that $\sqsubseteq^d_{FDI}$ is a precongruence for $\mathcal{O}$ that respects all conditional liveness properties.

To show that $\sqsubseteq^d_{FDI}$ respects conditional liveness properties, let $C, G \subseteq A^*$, $p \sqsubseteq^d_{FDI} q$, and suppose $p \models \mathit{liveness}_C(G)$. I need to show that $q \models \mathit{liveness}_C(G)$. So suppose $\sigma \in CT(q)$ and $\rho \in C$ for some prefix $\rho \leq \sigma$. Then one out of three possibilities must apply: either $\sigma \in \mathit{diverg}(g) \subseteq \mathit{diverg}(p)$ or $\sigma \in \mathit{inf}(q) \subseteq \mathit{inf}_d(q) \subseteq \mathit{inf}_d(p)$ or $\langle \sigma, A\rangle \in \mathit{failures}(q) \subseteq \mathit{failures}_d(q) \subseteq \mathit{failures}_d(p)$. In the first and last case, one has $\sigma \in CT(p)$. Since $p \models \mathit{liveness}_C(G)$, there must be a $\xi \leq \sigma$ with $\xi \in G$, which had to be shown. In the second case either $\sigma \in \mathit{inf}(p) \subseteq CT(p)$, in which case the argument proceeds as above, or $\exists \nu \in \mathit{diverg}(p) \subseteq CT(p)$ with $\rho \leq \nu < \sigma$. In the latter case, there must be a $\xi \leq \nu$ with $\xi \in G$, and as $\nu < \sigma$ it follows that $q \models \mathit{liveness}_C(G)$.

That $\sqsubseteq^d_{FDI}$ is a precongruence for $\|_S$, τ_I and λ^m_s follows from the following observations:

$$
\begin{aligned}
\mathit{diverg}(p\|_S q) &= \{\sigma \mid \exists \langle \nu, X\rangle \in \mathit{failures}_d(p), \xi \in \mathit{diverg}(q).\ \sigma \in \nu\|_S \xi\} \cup \\
&\quad \{\sigma \mid \exists \nu \in \mathit{diverg}(p), \langle \xi, X\rangle \in \mathit{failures}_d(q).\ \sigma \in \nu\|_S \xi\} \\
\mathit{inf}_d(p\|_S q) &= \{\sigma \mid \exists \nu \in \mathit{inf}_d(p), \xi \in \mathit{inf}_d(q).\ \sigma \in \nu\|_S \xi\} \cup \\
&\quad \{\sigma \mid \exists \langle \nu, X\rangle \in \mathit{failures}_d(p), \xi \in \mathit{inf}_d(q).\ \sigma \in \nu\|_S \xi\} \cup \\
&\quad \{\sigma \mid \exists \nu \in \mathit{inf}_d(p), \langle \xi, X\rangle \in \mathit{failures}_d(q).\ \sigma \in \nu\|_S \xi\} \cup \\
&\quad \{\sigma \in A^\infty \mid \forall \rho < \sigma \exists \nu \in \mathit{diverg}(p\|_S q).\ \rho \leq \nu < \sigma\} \\
\mathit{failures}_d(p\|_S q) &= \{\langle \sigma, X \cup Y\rangle \mid \exists \langle \nu, X\rangle \in \mathit{failures}_d(p), \langle \xi, Y\rangle \in \mathit{failures}_d(q). \\
&\qquad\qquad X \setminus S = Y \setminus S \wedge \sigma \in \nu\|_S \xi\} \\
&\quad \cup \{\langle \sigma, X\rangle \mid \sigma \in \mathit{diverg}(p\|_S q) \wedge X \subseteq A\}. \\
\mathit{diverg}(\tau_I(p)) &= \{\tau_I(\sigma) \mid \tau_I(\sigma) \in A^* \wedge \sigma \in \mathit{inf}_d(p) \cup \mathit{diverg}(p)\} \\
\mathit{inf}_d(\tau_I(p)) &= \{\tau_I(\sigma) \mid \tau_I(\sigma) \in A^\infty \wedge \sigma \in \mathit{inf}_d(p)\} \\
&\quad \cup \{\sigma \in A^\infty \mid \forall \rho < \sigma \exists \nu \in \mathit{diverg}(\tau_I(p)).\ \rho \leq \nu < \sigma\} \\
\mathit{failures}_d(\tau_I(p)) &= \{\langle \tau_I(\sigma), X\rangle \mid \langle \sigma, X \cup I\rangle \in \mathit{failures}_d(p)\} \\
&\quad \cup \{\langle \sigma, X\rangle \mid \sigma \in \mathit{diverg}(\tau_I(p)) \wedge X \subseteq A\} \\
\mathit{diverg}(\lambda^m_s(p)) &= \{\lambda^m_s(\sigma) \mid \sigma \in \mathit{diverg}(p)\} \\
\mathit{inf}_d(\lambda^m_s(p)) &= \{\lambda^m_s(\sigma) \mid \sigma \in \mathit{inf}_d(p)\} \\
&\quad \cup \{\sigma \in A^\infty \mid \forall \rho < \sigma \exists \nu \in \mathit{diverg}(\lambda^m_s(p)).\ \rho \leq \nu < \sigma\} \\
\mathit{failures}_d(\lambda^m_s(p)) &= \{\langle \lambda^m_s(\sigma), X\rangle \mid \langle \sigma, \lambda^{-m}_s(X)\rangle \in \mathit{failures}_d(p)\} \\
&\quad \cup \{\langle \sigma, X\rangle \mid \sigma \in \mathit{diverg}(\lambda^m_s(p)) \wedge X \subseteq A\}.
\end{aligned}
$$

"$\Rightarrow$": Let $\sqsubseteq$ be any precongruence for $\mathcal{O}$ that respects conditional liveness properties, and suppose $p \sqsubseteq q$. I have to establish that $p \sqsubseteq^d_{FDI} q$. W.l.o.g. I may assume that neither p nor q has any trace containing the actions c or g. The argument for this is as in the proof of Theorem 7.

Suppose $\mathit{diverg}(p) \not\supseteq \mathit{diverg}(q)$; say $\sigma \in \mathit{diverg}(q) \setminus \mathit{diverg}(p)$. Let r be a deterministic process such that $CT(r) = \{\sigma cg\}$. Then each complete trace of $p\|^{c,g} r$ that contains c also contains g. Here I write $\|^{c,g}$ for $\|_{A\setminus\{c,g\}}$, the interleaving operator that synchronises on all visible actions except c and g. As $\sqsubseteq$ is a precongruence, $p \sqsubseteq q$ implies $p\|^{c,g} r \sqsubseteq q\|^{c,g} r$, and since $\sqsubseteq$ respects the canonical conditional liveness property, I obtain that each complete

trace of $q\|^{c,g}r$ that contains c must also contain g. However, as $\sigma \in \mathit{diverg}(q)$, $\sigma c \in \mathit{diverg}(q\|^{c,g}r) \subseteq CT(q\|^{c,g}r)$, although σc does not contain g.

Suppose $\mathit{inf}_d(p) \not\supseteq \mathit{inf}_d(q)$; say $\sigma \in \mathit{inf}_d(q) \setminus \mathit{inf}_d(p)$. So $\sigma \notin \mathit{inf}(p)$ and there is a $\rho < \sigma$ such that $\rho \leq \rho\nu < \sigma$ for no sequence $\rho\nu \in \mathit{diverg}(p)$. Let r be a deterministic process such that $CT(r) = \{\rho c \nu g \mid \rho\nu < \sigma\} \cup \{\sigma\}$. Then each complete trace of $p\|^g r$ that contains c, must also contain g. As $\sqsubseteq$ is a precongruence, $p \sqsubseteq q$ implies $p\|^{c,g}r \sqsubseteq q\|^{g,c}r$, and since $\sqsubseteq$ respects the canonical conditional liveness property, I obtain that each complete trace of $q\|^{c,g}r$ that contains c must also contain g. However, either $\sigma \in \mathit{inf}(q)$ or $\rho\nu \in \mathit{diverg}(q)$ for some $\rho \leq \rho\nu < \sigma$. In each case $q\|^{c,g}r$ has a complete trace that contains c but not g.

Suppose $\mathit{failures}_d(p) \not\supseteq \mathit{failures}_d(q)$; say $\langle \sigma, X\rangle \in \mathit{failures}_d(q) \setminus \mathit{failures}_d(p)$. So $\langle \sigma, X\rangle \notin \mathit{failures}(p)$ and $\sigma \notin \mathit{diverg}(p)$. Let r be a deterministic process with $CT(r) = \{\sigma c a \mid a \in X\}$, let C be the set of sequences containing c, and consider the conditional liveness property given by C and $G := \{\sigma c a \mid a \in X\}$. Then $p\|^c r \models \mathit{liveness}_C(G)$. As $\sqsubseteq$ is a precongruence, $p \sqsubseteq q$ implies $p\|^c r \sqsubseteq q\|^c r$, and since $\sqsubseteq$ respects conditional liveness properties, also $q\|^g r \models \mathit{liveness}_C(G)$. However, either $\langle \sigma, X\rangle \in \mathit{failures}(q)$ or $\sigma \in \mathit{diverg}(q)$. So $\sigma c \in CT(q\|^g r)$, contradicting that $q\|^c r \models \mathit{liveness}_C(G)$. □

In [14], Bill Roscoe has shown that $\sqsubseteq^d_{FDI}$ is a precongruence for all operators of CSP; he also developed a new fixed point theory that shows that it is a congruence for recursion as well.

6 Linear Time Properties

Safety, liveness, and conditional liveness properties, as studied in the previous sections, are special cases of *linear time properties*. A linear time property can be thought of as any requirement on the observable content of the runs of a process. The property is satisfied by a process when the observable content of all its maximal runs satisfy this requirement. Hence a linear time property can be formalised by the set of sequences over A^ω that, when performed in a maximal run of a process, meet the requirement.

Definition 12 A *linear time property* of processes in an LTS is given by a set $P \subseteq A^\omega$. A process p *satisfies* this property, notation $p \models P$, when $CT(p) \subseteq P$.

A safety property is a special kind of linear time property, namely $\mathit{safety}(B) = \{\sigma \in A^\omega \mid \neg\exists \rho \in B.\ \rho \leq \sigma\}$. Likewise, $\mathit{liveness}(G) = \{\sigma \in A^\omega \mid \exists \rho \in G.\ \rho \leq \sigma\}$, and $\mathit{liveness}_C(G) = \{\sigma \in A^\omega \mid (\exists \rho \in C.\ \rho \leq \sigma) \Rightarrow (\exists \nu \in G.\ \nu \leq \sigma)\}$.

In [9] and most subsequent work, liveness properties are formalised in a different way than in this paper. For the canonical liveness property it is fundamentally impossible to ever tell that it is not going to be satisfied when one has only observed a finite prefix of a maximal run of a process. For if "something good" is promised to happen, it is always possible to assume it will be further in the future. In [9], this is taken to be the defining characteristic of liveness properties, and a property P is called a liveness property iff $\forall \rho \in A^*.\exists \sigma \in P.\ \rho \leq \sigma$.

The property *liveness*(G) with $G = \{a\}$ for instance says that the first visible action of a process should be an a. It is a liveness property in my sense, since the first action being an a can be thought of as a good thing that happened eventually; here the requirement that it has to happen as first action could be part of one's concept of *good*. However, it is not a liveness property as formalised in [9] and subsequent work, since the occurrence of a $b \neq a$ as first action proves that the property will never be satisfied.

The property that from some point onwards all visible actions a process performs should be g's, is an example of a liveness property in the sense of [9] that is not a liveness property in my sense. Namely, at no point can one ever tell that something good has happened.

A well know theorem [9] says that any linear time property P can be written as the conjunction $\mathit{safety}(B) \cap P_{\mathit{liveness}}$ of a safety property and a liveness property in the sense of [9]. Namely,

$$B := \{\rho \in A^* \mid \neg\exists \sigma \in P.\ \rho \leq \sigma\} \quad \text{and} \quad P_{\mathit{liveness}} := P \cup (A^\omega \setminus \mathit{safety}(B)).$$

Such a theorem does not hold for my liveness properties.

My characterisation of $\sqsubseteq_{\mathit{liveness}}$ would still be valid if I would have taken as class of liveness properties the intersection of mine and the ones from [9]. This follows immediately from Theorem 5, as the canonical liveness property is in this intersection. So the extra generality in my definition is harmless. However, the extra restriction makes a difference, as the canonical conditional liveness property, for instance, is a liveness property in the sense of [9].

Liveness properties in the sense of [9] are studied because proving them requires a different tool set than proving safety properties. However, as far as practical applications are concerned, one is mostly interested in conjunctions of safety and liveness properties, i.e. general linear time properties. I will therefore not try to characterise coarsest congruences that respect just the liveness properties in the sense of [9].

The coarsest congruence respecting all linear time properties has been characterised as *NDFD-equivalence* by Roope Kaivola and Antti Valmari in [8]; this results extends to preorders in a straightforward way. The NDFD preorder can be defined just like $\sqsubseteq^d_{FDI}$, except that $\inf(\_)$ is used instead of $\mathit{inf}_\perp(\_)$. In fact, this result can also be obtained as corollary of what we have seen so far.

Theorem 10 $p \sqsubseteq_{\mathit{lt\text{-}properties}} q \quad \Leftrightarrow \quad \begin{array}{rcl} \mathit{diverg}(p) & \supseteq & \mathit{diverg}(q) \wedge \\ \mathit{inf}(p) & \supseteq & \mathit{inf}(q) \wedge \\ \mathit{failures}_d(p) & \supseteq & \mathit{failures}_d(q). \end{array}$

Proof: Let $\sqsubseteq_{NDFD}$ be the preorder defined by: $p \sqsubseteq_{NDFD} q$ iff the right-hand side of Theorem 10 holds.
"$\Leftarrow$": It suffices to establish that $\sqsubseteq_{NDFD}$ is a precongruence for $\mathcal{O}$ that respects all linear time properties.

To show that $\sqsubseteq_{NDFD}$ respects linear time properties, let $P \subseteq A^\omega$, $p \sqsubseteq_{NDFD} q$, and suppose $p \models P$. I need to show that $q \models P$. So suppose $\sigma \in CT(q)$. Then either $\sigma \in \mathit{diverg}(g) \subseteq \mathit{diverg}(p)$ or $\sigma \in \mathit{inf}(q) \subseteq \mathit{inf}(q)$ or $\langle \sigma, A\rangle \in \mathit{failures}(q) \subseteq$

$\mathit{failures}_d(q) \subseteq \mathit{failures}_d(p)$. In the last case, one has either $\langle \sigma, A\rangle \in \mathit{failures}(p)$ or $\sigma \in \mathit{diverg}(p)$. So in all cases $\sigma \in CT(p)$. Since $p \models P$, it must be that $\sigma \in P$. It follows that $CT(q) \subseteq P$, i.e. $q \models P$.

That $\sqsubseteq^d_{FDI}$ is a precongruence for $\|_S$, τ_I and λ^m_s follows from similar, but simpler, observations as in the proof of Theorem 9.

"$\Rightarrow$": Let $\sqsubseteq$ be any precongruence for $\mathcal{O}$ that respects linear time properties, and suppose $p \sqsubseteq q$. I have to establish that $p \sqsubseteq_{NDFD} q$. That $\mathit{diverg}(p) \supseteq \mathit{diverg}(q)$ and $\mathit{failures}_d(p) \supseteq \mathit{failures}_d(q)$ follows immediately from Theorem 9, using that conditional liveness properties are linear time properties. That $\mathit{inf}(p) \supseteq \mathit{inf}(q)$ follows immediately by considering the linear time property $CT(p)$. □

To obtain this result it suffices to define $\sqsubseteq_{\mathit{lt\text{-}properties}}$ as the coarsest precongruence w.r.t. $\|_S$ and injective renaming that respects all linear time properties. However, it happens to also be a precongruence for all operators of CSP.

Linear time properties do not capture the entire observable behaviour or processes in the neutral environment. Orthogonal to them are *possibility properties*, such as: a process *may* do an action g. As argued by Leslie Lamport, "verifying possibility properties tells you nothing interesting about a system" [10]. Nevertheless, it is not hard to characterise the coarsest precongruence for $\mathcal{O}$ that respects linear time properties as well as all possibility properties, and thereby arguably the entire observable behaviour of a processes in a neutral environment. It is $\equiv_{NDFD}$, the symmetric closure of $\sqsubseteq_{NDFD}$.

7 Concluding remark

The methodology of the paper is close in spirit to the work on testing equivalences by Rocco De Nicola and Matthew Hennessy [5], and the results in Sections 3 and 4 are comparable as well. The notion of *must testing* of [5] could be reinterpreted as a way to test liveness properties, and hence, unsurprisingly, my preorder $\sqsubseteq_{\mathit{liveness}}$ is exactly the must-testing preorder of [5]. However, my safety preorder is exactly the *inverse* of the *may testing* preorder of [5]. This can be explained by thinking, in the context of may testing, of the "success"-action ω as marking a state of *failure*, rather than one of *success*. Now the property of whether a process may reach ω is exactly the negation of whether it will always avoid ω. This turns may-testing around, from testing certain possibility properties, to testing safety properties. It remains to elaborate a theory of testing that captures the concept of conditional liveness.

References

1. M. Alexander & W. Gardner, editors (2008): *Process Algebra for Parallel and Distributed Processing*. Chapman & Hall.
2. J.C.M. Baeten & J.A. Bergstra (1988): *Global renaming operators in concrete process algebra*. *Information and Computation* 78(3), pp. 205–245.
3. J.A. Bergstra & J.W. Klop (1985): *Algebra of communicating processes with abstraction*. *Theoretical Computer Science* 37(1), pp. 77–121.

4. 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.
5. R. De Nicola & M. Hennessy (1984): *Testing equivalences for processes*. *Theoretical Computer Science* 34, pp. 83–133.
6. R.J. van Glabbeek & M. Voorhoeve (2006): *Liveness, Fairness and Impossible Futures*. In: Christel Baier & Holger Hermanns, editors: *CONCUR*, LNCS 4137, Springer, pp. 126–141. Available at `http://dx.doi.org/10.1007/11817949_9`.
7. 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. Available at `http://Boole.stanford.edu/pub/spectrum1.ps.gz`.
8. R. Kaivola & A. Valmari (1992): *The Weakest Compositional Semantic Equivalence Preserving Nexttime-less Linear Temporal Logic*. In: Rance Cleaveland, editor: *CONCUR'92*, LNCS 630, Springer, pp. 207–221. Available at `http://dx.doi.org/10.1007/BFb0084793`.
9. L. Lamport (1977): *Proving the correctness of multiprocess programs*. *IEEE Transactions on Software Engineering* 3(2), pp. 125–143.
10. Leslie Lamport (1998): *Proving Possibility Properties*. *Theoretical Computer Science* 206(1-2), pp. 341–352. See especially `http://research.microsoft.com/en-us/um/people/lamport/pubs/pubs.html#lamport-possibility`.
11. P.B. Levy (2008): *Infinite trace equivalence*. *Annals of Pure and Applied Logic* 151(2-3), pp. 170–198. Available at `http://dx.doi.org/10.1016/j.apal.2007.10.007`.
12. 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-Verlag, 1980.
13. A. Puhakka (2001): *Weakest Congruence Results Concerning "Any-Lock"*. In: N. Kobayashi & B.C. Pierce, editors: *Proceedings TACS'01*, LNCS 2215, Springer, pp. 400–419.
14. A. W. Roscoe (2004): *Seeing Beyond Divergence*. In: Ali E. Abdallah, Cliff B. Jones & Jeff W. Sanders, editors: *25 Years Communicating Sequential Processes*, *Lecture Notes in Computer Science* 3525, Springer, pp. 15–35. Available at `http://dx.doi.org/10.1007/11423348_2`.
15. A.W. Roscoe (1997): *The Theory and Practice of Concurrency*. Prentice-Hall. Available at `http://www.comlab.ox.ac.uk/bill.roscoe/publications/68b.pdf`.