Liveness, Fairness and Impossible Futures

Rob van Glabbeek[1,2] and Marc Voorhoeve[3]

[1] National ICT Australia, Sydney
[2] School of Computer Science and Engineering, The University of New South Wales
[3] Dept. of Mathematics and Computer Science, Eindhoven University of Technology

Abstract. Impossible futures equivalence is the semantic equivalence on labelled transition systems that identifies systems iff they have the same "AGEF" properties: temporal logic properties saying that reaching a desired outcome is not doomed to fail. We show that this equivalence, with an added root condition, is the coarsest congruence containing weak bisimilarity with explicit divergence that respects deadlock/livelock traces (or fair testing, or any liveness property under a global fairness assumption) and assigns unique solutions to recursive equations.

1 Introduction

This paper deals with a class of system requirements, and related notions of process equivalence, that we introduce by the following tale.

Pete's mobile phone allows a number to be redialed as long as connection attempts are unsuccessful. The phone's manual charts this functionality (Fig. 1, left hand side; *ok* and *nok* are internal actions that cannot be observed or interacted with). After having lost several valuable business opportunities, Pete finds out that his redial module contains a bug. During the redial process, data can become corrupted so that all connection attempts fail from then on. Pete contacts his vendor for damages, who denies responsibility, since all code has been certified by a company named TEI (Testing Equivalences Inc). Upon contacting TEI, their spokesman says: "We indeed have discovered the feature you complain about. Our technical people have even charted the functionality implemented (Fig. 1, right hand side, dashed arc omitted). However you have nothing to complain about, because we have verified that the two systems are equivalent with respect to ready simulation. This is our finest equivalence, highly recommended by concurrency specialists [4]." Our hero is considering his next step.

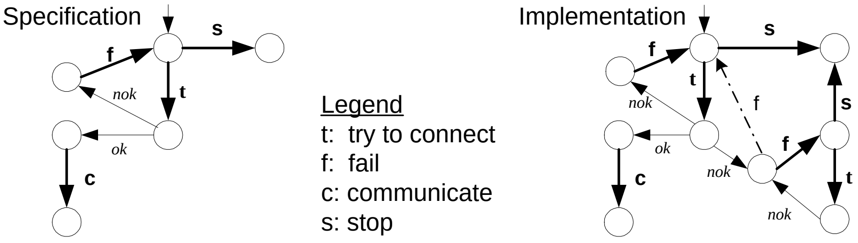

Fig. 1. Charts of Pete's mobile phone

The specification in his manual led Pete to believe that his phone satisfies the requirement: every redial attempt *may* succeed. Of course, the attempts may fail, but attempts that are *doomed* to fail are not acceptable. In CTL [7], when taking observable histories of states as atomic propositions, Pete's requirement may be formulated as, for all $k \in \mathbb{N}$, $\mathbf{AG}((tf)^k$ has occurred $\Rightarrow \mathbf{EF}$ $(tf)^k tc$ has occurred). We call such requirements *AGEF properties*. Pete's requirement is not preserved by testing equivalences such as ready simulation [4] or failures equivalence [6].

AGAF properties are (conditional) liveness requirements, stating that (depending on past activity) some condition will eventually hold. In Pete's case, such a requirement would be: if I keep hitting the redial button, I will eventually be connected. The implementation in Fig. 1 does not satisfy this AGAF property, but it is open to debate whether the specification satisfies it. In order to deduce that requirement, a *fairness* assumption is needed [9], e.g.: in a recurring state, a specific option cannot be avoided infinitely often. This assumption allows to distinguish the processes in Fig. 1, even without testing the AGEF property. We show that under a sufficiently strong fairness assumption any AGAF property can be reformulated as a conjunction of AGEF properties. Since the validity of AGEF properties does not depend on fairness, it appears preferable to directly verify the AGEF requirements rather than assume fairness and verify the AGAF requirement. *Fair testing equivalence* [5] preserves the subset of *testable* AGEF properties, including the reformulated AGAF properties. Absence of initial deadlock and livelock is an example of a testable AGEF property. However, many reasonable requirements such as Pete's not-doomed-to-fail requirement are in fact non-testable AGEF properties. If there is e.g. a possibility that the corrupted data in Pete's phone can become uncorrupted by redialling frantically, as indicated by the dashed arc in Fig. 1, the erroneous implementation is fair testing equivalent to the specification. However, Pete will still be far from satisfied.

In this paper we study the impossible futures (IF) equivalence of [13, 14] that preserves all AGEF properties. We prove that any process equivalence that is a congruence w.r.t. the operators of calculi like CCS and CSP either preserves all AGEF properties, all testable properties only, or no proper AGEF property at all. Moreover, any equivalence that preserves just the testable AGEF properties and is coarser than weak bisimilarity with explicit divergence ($\leftrightarrow^{\Delta}_{rw}$) does not respect the *recursive specification principle* (RSP), stating that solutions of guarded recursive equations are unique. This proof principle is of great importance in equational system verification [2]. So, among the semantic equivalences coarser than $\leftrightarrow^{\Delta}_{rw}$,[4] IF is the coarsest congruence allowing both the preservation of any chosen proper AGEF property and equational system verification.

2 Labelled Transition Systems

Let Σ^* denote the set of finite sequences over a given set Σ. Write ε for the empty sequence, $\sigma\rho$ for the concatenation of sequences σ and ρ, and a for the

[4] We were unable to extend our results to equivalences incomparable with both $\leftrightarrow^{\Delta}_{rw}$ and IF, preserving some AGEF property. However, no such equivalences are known.

sequence consisting of the single element $a \in \Sigma$. Write $\sigma \leq \rho$ if σ is a prefix of ρ, i.e. $\exists \nu \in \Sigma^* \cdot \sigma\nu = \rho$, and write $\sigma < \rho$ if $\sigma \leq \rho$ and $\sigma \neq \rho$.

We presuppose a countable action alphabet A, not containing the "silent" action τ and set $A_\tau = A \cup \{\tau\}$. We assume that our set A consists of complementary pairs; each $a \in A$ has a complement $\bar{a}$ such that $\bar{\bar{a}} = a$.

Definition 1. A *labelled transition system* (LTS) is a pair $(\mathbb{P}, \rightarrow)$, where $\mathbb{P}$ is a set (of *processes* or *states*) and $\rightarrow \subseteq \mathbb{P} \times A_\tau \times \mathbb{P}$ is a set of *transitions.*

Assuming a fixed transition system $(\mathbb{P}, \rightarrow)$, we write $p \xrightarrow{a} p'$ for $(p, a, p') \in \rightarrow$. $p \xrightarrow{a} p'$ means that process p can evolve into p', while performing the action a.

Definition 2. The ternary relation $\overset{-}{\Longrightarrow} \subseteq \mathbb{P} \times A^* \times \mathbb{P}$ is the least relation satisfying $p \overset{\varepsilon}{\Longrightarrow} p$, $\dfrac{p \xrightarrow{\tau} p'}{p \overset{\varepsilon}{\Longrightarrow} p'}$, $\dfrac{p \xrightarrow{a} p',\ a \neq \tau}{p \overset{a}{\Longrightarrow} p'}$, and $\dfrac{p \overset{\sigma}{\Longrightarrow} p' \overset{\rho}{\Longrightarrow} p''}{p \overset{\sigma\rho}{\Longrightarrow} p''}$.
We write $p \overset{\sigma}{\Longrightarrow}$ for $\exists p' \cdot p \overset{\sigma}{\Longrightarrow} p'$ and $p \overset{*}{\Longrightarrow} p'$ for $\exists \sigma \cdot p \overset{\sigma}{\Longrightarrow} p'$.
Let $\mathcal{T}(p) = \{\sigma \in A^* \mid p \overset{\sigma}{\Longrightarrow}\}$ be the set of *traces* of p,
and $\mathcal{A}(p) = \{a \in A \mid a$ or $\bar{a}$ occurs in a trace of $p\}$ the *alphabet* of p.

In this paper we present particular processes that are instrumental in proving our results (cf. Figs. 2, 3). Therefore our results apply only to transition systems in which those processes exist. To ensure this, we assume that $(\mathbb{P}, \rightarrow)$

1. is closed under *action prefixing*, meaning that for any $p \in \mathbb{P}$ and $a \in A_\tau$ there is a process ap such that $ap \xrightarrow{c} q$ iff $a = c$ and $q = p$, and
2. is closed under *countable summation*, meaning that for every countable set of processes $P \subseteq \mathbb{P}$ there is a process $\sum P$ such that for all $a \in A$ we have $\sum P \xrightarrow{a} q$ iff there exists $p \in P$ such that $p \xrightarrow{a} q$.

Alternatively we may assume that $(\mathbb{P}, \rightarrow)$ contains finite-state processes only, and is closed under action prefixing and finite summation (cf. Remark 1 in Sect. 4). We also postulate the Fresh Atom Principle [11], allowing fresh actions in proofs.

When writing expressions with action prefixing and summation, we let 0 stand for $\sum \emptyset$ and $p + q$ for $\sum\{p, q\}$, and prefixing binds stronger than sum. We write a for $a0$ and, when $\sigma = a_1 \ldots a_n$, write σp for $a_1 \ldots a_n p$ and $\bar{\sigma}$ for $\bar{a}_1 \ldots \bar{a}_n$.

3 Coarsest Congruence Relations on Processes

Semantic equivalences on processes are used to assess whether an implementation has the same functionality as its specification, viz. Fig. 1. The equivalence of two processes should guarantee that if one has a certain desirable property, then so has the other. In the context of an LTS $(\mathbb{P}, \rightarrow)$, properties can be modelled as unary predicates $\varphi \subseteq \mathbb{P}$. A semantic equivalence relation $\sim \subseteq \mathbb{P} \times \mathbb{P}$ *respects* or *preserves* a property φ if $p \sim q \Rightarrow (\varphi(p) \Leftrightarrow \varphi(q))$. Thus, a semantic equivalence should respect all relevant properties of the systems on which it is applied. Naturally, what is relevant depends to a large extent on the intended application, and consequently many semantic equivalences have been proposed in the literature [10]. This paper focusses on system requirements that we call

AGEF and AGAF properties; they will be defined in Section 4.

A transition system $(\mathbb{P}, \rightarrow)$ is often equipped with process algebraic operators $f : \mathbb{P}^n \rightarrow \mathbb{P}$. Throughout this paper, we shall assume that $(\mathbb{P}, \rightarrow)$ is equipped with the *parallel composition* $(\_|\_)$ and *restriction* $\_\backslash H$ for $H \subseteq A$ of CCS [12].

Definition 3. The CCS *parallel composition operator* is a binary operator $(\_|\_)$ defined on $\mathbb{P}$ in such a way that, for all $p, q, r \in \mathbb{P}$ and for all $a \in A_\tau$, $p \,|\, q \xrightarrow{a} r$ iff

1. there exists $p' \in \mathbb{P}$ such that $p \xrightarrow{a} p'$ and $r = p' \,|\, q$; or
2. there exists $q' \in \mathbb{P}$ such that $q \xrightarrow{a} q'$ and $r = p \,|\, q'$; or
3. $a = \tau$ and $\exists p', q' \in \mathbb{P}$ and $b \in A$ such that $p \xrightarrow{b} p'$, $q \xrightarrow{\bar{b}} q'$ and $r = p' \,|\, q'$.

A *restriction operator* is a unary operator $\_\backslash H$ defined on $\mathbb{P}$ in such a way that, for all $p, q \in \mathbb{P}$ and for all $a \in A_\tau$, $p\backslash H \xrightarrow{a} q$ iff $a, \bar{a} \notin H$ and there exists $p' \in \mathbb{P}$ such that $p \xrightarrow{a} p'$ and $q = p'\backslash H$.

Component-based design often results in processes of the form $(p_0 \,|\, \cdots \,|\, p_n)\backslash H$ when formalised in CCS. If the component p_i is replaced by an equivalent component q_i, we want to be able to conclude that the resulting composition is equivalent to the original. Due to the state explosion phenomenon, it is often infeasible to check this explicitly. Therefore, a second requirement on semantic equivalence relations is that they are congruences for all relevant composition operators; this in order to allow compositional verification.

Definition 4. A semantic equivalence relation $\sim \subseteq \mathbb{P} \times \mathbb{P}$ is a *congruence* for an operator $f : \mathbb{P}^n \rightarrow \mathbb{P}$, or f is *compositional* for $\sim$, if $p_i \sim q_i$ for $i = 1, ..., n$ implies that $f(p_1, ..., p_n) \sim f(q_1, ..., q_n)$.

Often, one requires compositionality of all operators of CCS and CSP; the minimum requirement typically involves just the operators $|$ and $\backslash H$ of CCS, or alternatively the *parallel composition* and *concealment* operators of CSP.

Let $\sim, \approx \subseteq \mathbb{P} \times \mathbb{P}$ be equivalence relations. Then $\sim$ is called *finer* than $\approx$ and $\approx$ *coarser* than $\sim$ if $\sim \subseteq \approx$. (Note that we use these concepts in a non-strict sense.) As explained above, we seek semantic equivalences that (1) preserve important properties of the processes on which they will be applied, (2) are congruences for the operators that are used to compose processes, and (3) possibly satisfy some other requirements, such as RSP (see the introduction). When the requirements are completely clear and not subject to change, amongst multiple equivalences that meet all requirements, the coarsest of them, if it exists, constitutes the ultimate criterion for system verification, as it enables more implementations to be shown correct with respect to a given specification. The main goal of this paper is to characterise such coarsest equivalences.

4 AGEF and AGAF Properties

Definition 5. The set $\mathcal{I}(p)$ of *impossible futures* of a process p is the set of pairs $(\sigma, G) \in A^* \times \mathcal{P}(A^*)$ satisfying

$$\exists p' \cdot p \overset{\sigma}{\Longrightarrow} p' \wedge G \cap \mathcal{T}(p') = \emptyset.$$

Processes p, q are *IF-equivalent*, notation $p \sim_{\mathcal{I}} q$, iff $\mathcal{I}(p) = \mathcal{I}(q)$.

Note that $(tf, \{tc\}) \in \mathcal{I}(q) \setminus \mathcal{I}(p)$, where p, q are respectively the left- and right-hand processes of Fig. 1. (The transitions *ok* and *nok* are labelled τ.) So p and q are not IF-equivalent. The statement $(\sigma, G) \notin \mathcal{I}(p)$ expresses the property

$$\forall p' \cdot p \overset{\sigma}{\Longrightarrow} p' \Rightarrow \exists \rho \in G \cdot p' \overset{\rho}{\Longrightarrow}.$$

Pete's redialling requirement consists of the conjunction of these properties for $\sigma = (tf)^k$ and $G = \{tc\}$. We call them *AGEF properties.*

Definition 6. For $\sigma \in A^*$ and $G \subseteq A^*$, let AGEF(σ, G) be the property (subset) of processes with $p \in$ AGEF(σ, G) iff $\forall p' \cdot p \overset{\sigma}{\Longrightarrow} p' \Rightarrow \exists \rho \in G \cdot p' \overset{\rho}{\Longrightarrow}$.

Now a process p satisfies AGEF(σ, G) iff $(\sigma, G) \notin \mathcal{I}(p)$. Thus, an equivalence on processes respects all AGEF properties if and only if it is finer than $\sim_{\mathcal{I}}$. Note that AGEF$(\sigma, G \cup G') =$ AGEF(σ, G) if every $\rho \in G'$ has a prefix $\rho' \in G$. We therefore assume w.l.o.g. that the sets G have the *prefix property*: $\forall \rho, \nu \in G \cdot \rho \not< \nu$.

The name AGEF is derived from a way to express such properties in *Computation Tree Logic* (CTL) [7]. CTL is a formalism to specify temporal properties of systems that are modelled as states in *Kripke structures.* The latter are transition systems in which states rather than transitions are labelled. Amongst others, CTL features the formulas, interpreted on a state s,

$\mathbf{AF}\varphi$	meaning that every path from s eventually passes a state satisfying φ
$\mathbf{EF}\varphi$	meaning that some path from s eventually passes a state satisfying φ
$\mathbf{AG}\varphi$	meaning that on every path from s all states satisfy φ
$\mathbf{EG}\varphi$	meaning that on some path from s all states satisfy φ
ℓ	meaning that state s has label ℓ.

Here φ is again a CTL formula. CTL formulas can also be combined with propositional connectives. In order to interpret CTL formulas on a process p in an LTS $\mathbb{L}$, we convert the part of $\mathbb{L}$ that is reachable from p into a Kripke structure by unwinding it into a tree, and labelling each state with the trace of the unique path leading to it. This leads to a Kripke structure $\mathbb{L}_p$ whose states are the finite paths π in $\mathbb{L}$ starting from p, labelled with the sequence of visible actions labelling π, and there is a transition $\pi \to \pi'$ iff the path π' can be obtained from π by adding one transition. We say that p satisfies a CTL formula φ iff the root of the tree-shaped Kripke structure $\mathbb{L}_p$ satisfies φ.[5] Now the property AGEF(σ, G) is expressed in CTL as $\mathbf{AG}(\sigma \Rightarrow \mathbf{EF} \bigvee_{\rho \in G} \sigma\rho)$.

We also consider *conditional liveness* requirements or *AGAF properties*, stating that *something good* will eventually happen when a specific past has been observed. The property AGAF(σ, G) with $G \neq \emptyset$ states that every run with visible content σ *will* be completed to a run with visible content $\sigma\rho$ for $\rho \in G$, *provided no visible action occurs that disables the potential of achieving G.* In contrast, the property AGEF(σ, G) says that any run with visible content σ *can* be completed to such a run. A liveness property of the form AGAF(ε, G) states that something good will happen unconditionally. Pete's liveness requirement "if

[5] Other translations from LTSs to Kripke structures have appeared in the literature [8], leading to different interpretations of CTL on LTSs.

I keep hitting the redial button, I will eventually be connected" is $\mathrm{AGAF}(\varepsilon, P)$ with $P = \{(tf)^k tc \mid k \in \mathbb{N}\}$.

We now formulate the fairness principle $\mathcal{F}$: during a system run, a specific set of states that remains reachable throughout cannot be avoided forever. This amounts to strong fairness [9] for finite-state processes. We say that a path satisfies $\mathcal{F}(\psi)$, with ψ a set of states, if it is not a infinite path with ψ reachable throughout and avoiding ψ forever. The requirement $\mathrm{AGAF}(\sigma, G)$ under the assumption $\mathcal{F}$ is written $\mathrm{AGAF}_{\mathcal{F}}(\sigma, G)$. The specification of Pete's phone satisfies $\mathrm{AGAF}_{\mathcal{F}}(\varepsilon, P)$ but not $\mathrm{AGAF}(\varepsilon, P)$. The implementation satisfies neither.

In order to conveniently express AGAF properties in temporal logic, we add a modality $\mathbf{A}^{\chi}\mathbf{F}$ to CTL. Here χ is a property on paths, and $\mathbf{A}^{\chi}\mathbf{F}\varphi$ holds in state s, if every (possibly infinite) path from s that is maximal (cannot be extended) amongst the paths satisfying χ, passes through a state satisfying φ. $\mathrm{AGAF}(\sigma, G)$ with $\sigma \in A^*$ and $\emptyset \neq G \subseteq A^*$ can be expressed as $\mathbf{AG}(\sigma \Rightarrow \mathbf{A}^{[G]}\mathbf{F}\psi)$, and $\mathrm{AGAF}_{\mathcal{F}}(\sigma, G)$ as $\mathbf{AG}(\sigma \Rightarrow \mathbf{A}^{\mathcal{F}(\psi) \wedge [G]}\mathbf{F}\psi)$, where $\psi = \bigvee_{\rho \in G} \sigma\rho$ and $[G]$ is the property of a path that all labels $\sigma\nu$ of its states satisfy $\exists \rho \in G \cdot \nu \leq \rho$.

Pete's liveness requirement cannot be expressed as a property of the form $\mathbf{AG}(\sigma \Rightarrow \mathbf{A}^{\mathcal{F}(\psi)}\mathbf{F}\psi)$ with $\psi = \bigvee_{\rho \in G} \sigma\rho$. When taking $\sigma = \varepsilon$ and $G = P$ this property says "eventual connection is guaranteed", which is easily refuted by hitting the *stop* button; taking $G = P \cup \{(tf)^k s \mid k \in \mathbb{N}\}$ yields a requirement that is satisfied by the buggy implementation.

We will show that any $\mathrm{AGAF}_{\mathcal{F}}$ property can be formulated as a conjunction of AGEF properties. We write $\mathrm{AGEF}\,C$ with $C \subseteq A^* \times \mathcal{P}(A^*)$ for the conjunction $\bigwedge_{(\sigma,G) \in C} \mathrm{AGEF}(\sigma, G)$, and similarly for $\mathrm{AGAF}\,C$ and $\mathrm{AGAF}_{\mathcal{F}}\,C$.

Let $\sigma \in A^*$ and $\emptyset \neq G \subseteq A^*$. Then $\uparrow(\sigma, G) := \{(\sigma\rho, \rho^{-1}G) \mid \rho \in {\downarrow}G\}$, where ${\downarrow}G := \{\nu \in A^* \mid \exists \rho \cdot \nu\rho \in G\} \setminus \{\rho\nu \mid \rho \in G\}$ (the set of proper prefixes of G) and $\sigma^{-1}G := \{\rho \mid \sigma\rho \in G\}$. For instance, $\uparrow(a, \{b, cd\}) = \{(a, \{b, cd\}), (ac, \{d\})\}$.

Lemma 1. *Let* $\sigma \in A^*$ *and* $\emptyset \neq G \subseteq A^*$. *Then*
$\mathrm{AGAF}(\sigma, G) \subseteq \mathrm{AGAF}_{\mathcal{F}}(\sigma, G) \subseteq \mathrm{AGEF}(\sigma, G)$,
$\mathrm{AGAF}(\sigma, G) = \mathrm{AGAF}{\uparrow}(\sigma, G)$ *and* $\mathrm{AGAF}_{\mathcal{F}}(\sigma, G) = \mathrm{AGAF}_{\mathcal{F}}{\uparrow}(\sigma, G)$.

Proof. The inclusions are trivial; something that will happen must surely be possible. The equalities state e.g. that $\mathrm{AGAF}(\sigma, G)$ implies $\mathrm{AGAF}(\sigma\rho, \rho^{-1}G)$ for any $\rho \in {\downarrow}G$: a promise remains valid as long as it hasn't been delivered. □

Theorem 1. *The property* $\mathrm{AGAF}_{\mathcal{F}}(\sigma, G)$ *is equal to* $\mathrm{AGEF}{\uparrow}(\sigma, G)$.

Proof. By Lemma 1, we find $\mathrm{AGAF}_{\mathcal{F}}(\sigma, G) = \mathrm{AGAF}_{\mathcal{F}}{\uparrow}(\sigma, G) \subseteq \mathrm{AGEF}{\uparrow}(\sigma, G)$. Let $\psi = \bigvee_{\rho \in G} \sigma\rho$. If $p \in \mathrm{AGEF}{\uparrow}(\sigma, G)$ then from every p' with $p \overset{\sigma\rho}{\Longrightarrow} p'$ and $\rho \in {\downarrow}G$, a ψ-state is reachable. Any run from p that starts with σ, avoids states labelled $\sigma\nu$ with $\nexists \rho \in G \cdot \nu \leq \rho$, and satisfies $\mathcal{F}(\psi)$, will eventually reach ψ. □

By Theorem 1, Pete's liveness requirement $\mathrm{AGAF}_{\mathcal{F}}(\varepsilon, P)$ (assuming fairness) is implied by $\mathrm{AGEF}{\uparrow}(\varepsilon, P) = \mathrm{AGEF}\{((tf)^n, P), ((tf)^n t, \{c\} \cup fP) \mid n \in \mathbb{N}\}$.

The property $\mathrm{AGAF}(\sigma, G)$ can be expressed in CTL as $\mathrm{AGEF}{\uparrow}(\sigma, G) \wedge \mathbf{AG}(\sigma \Rightarrow \mathbf{AF} \bigvee_{\rho \notin {\downarrow}G} \sigma\rho)$. We therefore do not need the modality $\mathbf{A}^{\chi}\mathbf{F}$ for stating AGAF properties with or without $\mathcal{F}$.

In [5] it is defined when a process p *should pass* a test. A *test* is given by a test process t, whose alphabet may contain an extra action $\checkmark$ that cannot occur in the alphabet of the process p. The test consists of running p and t in parallel using the CSP parallel composition operator $\|_A$ that forces all actions of p and t to synchronise, except for the action $\checkmark$. The occurrence of $\checkmark$ denotes a successful outcome of the test. We give an alternative formulation using the CCS operators: The process p *should pass* the test, notation p **shd** t, if $(p\,|\,t)\backslash(A \setminus \{\checkmark\})$ satisfies $\text{AGEF}(\varepsilon, \{\checkmark\})$. Processes p, q are *fair testing equivalent* [5], notation $p =_{\mathbf{shd}} q$, iff p **shd** $t \Leftrightarrow q$ **shd** t for all tests t.

Definition 7. A property φ on transition systems is *(should-) testable* if there exists a test t such that for all processes p one has p **shd** t iff p satisfies φ.

A property is called *trivial* if it either always holds or always fails. As p **shd** $\checkmark$ for any p and p **shd** 0 for no p, all trivial properties are testable. The trivial AGEF properties are $\text{AGEF}(\sigma, G)$ with $\varepsilon \in G$, and $\text{AGEF}(\varepsilon, \emptyset)$.

Proposition 1. *A nontrivial AGEF property AGEF(σ, G) is testable iff for each sequence $b\rho$ in G also its prefix b is in G.*

Proof. "If": We assume $\varepsilon \notin G$ and G has the prefix property $(\forall \rho, \nu \in G \cdot \rho \not< \nu)$. So G consists of singleton traces only. Let $\sigma = a_1 \ldots a_m$. We define the processes T_i $(0 \leq i \leq m)$ by $T_i = \bar{a}_{i+1} T_{i+1} + \checkmark$ for $0 \leq i < m$ and $T_m = \sum\{\bar{b}\checkmark \mid b \in G\}$. Fig. 2 displays the processes T_i $(0 \leq i \leq m)$ for finite $G = \{b_1, \ldots, b_n\}$.

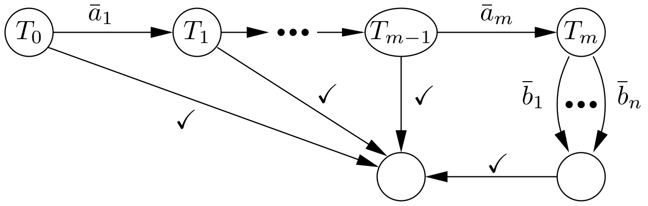

Fig. 2. The test process T_0

Now p satisfies $\text{AGEF}(\sigma, G)$ iff p **shd** T_0. Namely, if $(\sigma, G) \in \mathcal{I}(p)$, i.e. p does not satisfy $\text{AGEF}(\sigma, G)$, then there exists p' with $p \overset{\sigma}{\Longrightarrow} p'$ such that $\forall \rho \in G \cdot p' \overset{\rho}{\not\Longrightarrow}$, so $p\,|\,T_0 \overset{\varepsilon}{\Longrightarrow} p'\,|\,T_m \overset{\checkmark}{\not\Longrightarrow}$. Conversely, if $(\sigma, G) \notin \mathcal{I}(p)$, then for any (strict or not) prefix ν of σ and any p' with $p \overset{\nu}{\Longrightarrow} p'$ the $\checkmark$ can be done.

"Only if": Suppose G contains a sequence $b\rho$ but not b. Set $p := \sigma(b\rho + b)$ and $q := \sigma b\rho + \sigma b$ if $\sigma \neq \varepsilon$ and $p := \tau(b\rho + b)$ and $q := \tau b\rho + \tau b$ otherwise. Then p, q are fair testing equivalent [5], whereas $(\sigma, G) \in \mathcal{I}(q) \setminus \mathcal{I}(p)$. So $AGEF(\sigma, G)$ is not testable. □

Theorem 2. *All properties of the form* $\text{AGEF}{\uparrow}(\sigma, G)$ *are testable.*

Proof. Use the same test as above, but with T_m replaced by the deterministic process T_G with $\mathcal{T}(T_G) = \{\rho \in A^* \mid \exists \nu \cdot \rho\nu \in G\} \cup \{\rho\checkmark \mid \rho \in G\}$. □

Remark 1. When working in the context of a finite-state LTS, we only consider AGEF and AGAF properties with finite sets G. This way the test processes used above will be finite. That the correspondence between AGEF properties and $\sim_{\mathcal{I}}$ is unaffected by this change follows by

Lemma 2. *If $(\sigma, G) \in \mathcal{I}(q) \setminus \mathcal{I}(q)$ and q is a finite-state process, then there is a finite G' with $(\sigma, G') \in \mathcal{I}(q) \setminus \mathcal{I}(q)$.*

Proof. The set $R = \{r \mid q \stackrel{\sigma}{\Longrightarrow} r\}$ is finite and for each $r \in R$ we can choose a $\rho_r \in G$ such that $r \stackrel{\rho_r}{\Longrightarrow}$. Hence, $(\sigma, \{\rho_r \mid r \in R\}) \in \mathcal{I}(p) \setminus \mathcal{I}(q)$. □

A *safety property* says that *something bad will not happen.* Formalising "bad" as a predicate $B \subseteq A^*$ on the visible content of system runs, a safety property has the form $B \cap \mathcal{T}(x) = \emptyset$, and can be written as $\bigwedge_{\sigma \in B}$AGEF$(\sigma, \emptyset)$. Considering that the class of testable properties is closed under conjunction (for p **shd** $\tau t + \tau t'$ iff p **shd** t and p **shd** t'), Prop. 1 implies that safety properties are testable. A property AGEF(σ, G) or AGEF$\uparrow(\sigma, G)$ is called *proper* if it is neither trivial, nor a safety property, i.e. if $\varepsilon \notin G \neq \emptyset$.

5 Four Levels of Respect for AGEF Properties

In this section we show that only four types of congruences exist: those that respect all AGEF properties, those that respect all testable AGEF properties but no others, those that respect all safety properties but no other non-trivial AGEF properties, and those that do not respect a single non-trivial AGEF property. Examples in each of the four classes are weak bisimilarity [12], fair testing equivalence [5], trace equivalence—defined as $p =_{\mathcal{T}} q$ iff $\mathcal{T}(q) = \mathcal{T}(q)$—and failures equivalence [6] (where the absence of traces occurring past a divergence is not recorded), respectively. In this section "congruence" means congruence for the CCS parallel composition and restriction operators; we could also have used the CSP parallel composition and concealment operators. The results in this section are not needed further on, although we will reuse the proof of Lemma 3.

We say that a congruence $\sim$ is *non-IF* if there exist processes p, q with $p \sim q$ such that $\mathcal{I}(p) \neq \mathcal{I}(q)$. For a non-IF congruence there exists an AGEF property that it does not preserve; we shall now prove that in fact it does not preserve any non-testable AGEF property.

Lemma 3. *If $\sim$ is a non-IF congruence, then for all $c \in A$ there exist processes p_c, q_c with $\mathcal{A}(p_c) = \mathcal{A}(q_c) = \{c\}$, such that $p_c \sim q_c$ and $(\varepsilon, \{cc\}) \in \mathcal{I}(p_c) \setminus \mathcal{I}(q_c)$.*

Proof. The congruence $\sim$ is non-IF, so there exist processes p, q and σ, G such that $p \sim q$ and $(\sigma, G) \in \mathcal{I}(p) \setminus \mathcal{I}(q)$. Note that $\varepsilon \notin G$: if $\varepsilon \in G$ then $(\sigma, G) \notin \mathcal{I}(p)$.

Let $\sigma = a_1 \ldots a_m$ and define $H := \mathcal{A}(p) \cup \mathcal{A}(q)$. We first establish the lemma for all actions $c \notin H$, and then consider the case $c \in H$. Let U_i $(0 \leq i \leq m)$, V and W be defined by $U_i = \tau V + \bar{a}_{i+1} U_{i+1}$ for $0 \leq i < m$, $U_m = \tau V + \tau c W$, $V = c(\tau c + \tau)$ and $W = \sum_{\rho \in G} \bar{\rho}(\tau c + \tau)$ (Fig. 3 displays the processes U_i

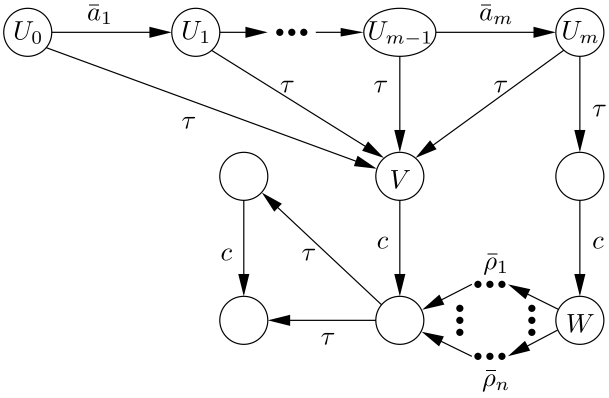

Fig. 3. The process U_0

$(0 \leq i \leq m)$, V and W for the case that $G = \{\rho_1, \ldots, \rho_n\}$). As we assume our alphabet A, and hence $G \subseteq A^*$, to be countable, the sum W is countable too. If q is a finite-state process, by Lemma 2 we may even assume it to be finite. Let $p_c = (U_0 \,|\, p)\backslash H$ and $q_c = (U_0 \,|\, q)\backslash H$. By construction, $\mathcal{A}(p_c) = \mathcal{A}(q_c) = \{c\}$. Since $p \sim q$ and $\sim$ is a congruence for $|$ and $\backslash H$, we have $(U_0 \,|\, p)\backslash H \sim (U_0 \,|\, q)\backslash H$. There exists a p' with $p \overset{\sigma}{\Longrightarrow} p'$ and $p' \overset{\rho}{\not\Longrightarrow}$ for all $\rho \in G$. Therefore, we have $(U_0 | p)\backslash H \overset{\varepsilon}{\Longrightarrow} (cW | p')\backslash H$ and $(cW | p')\backslash H \overset{cc}{\not\Longrightarrow}$. Hence, $(\varepsilon, \{cc\}) \in \mathcal{I}((U_0 \,|\, p)\backslash H)$. However, as $\forall q' \cdot (q \overset{\sigma}{\Longrightarrow} q') \Rightarrow (\exists \rho \in G \cdot q' \overset{\rho}{\Longrightarrow})$, we have $(\varepsilon, \{cc\}) \notin \mathcal{I}((U_0 \,|\, q)\backslash H)$. This proves our lemma for all actions $c \notin H$.

To obtain the required results for $c \in H$, first choose $d \notin H$ (appealing to the Fresh Atom Principle [11] if $A \setminus H$ is empty). By the above, there exist p_d and q_d with $\mathcal{A}(p_d) = \mathcal{A}(q_d) = \{d\}$ such that $p_d \sim q_d$ and $(\varepsilon, \{dd\}) \in \mathcal{I}(p_d) \setminus \mathcal{I}(q_d)$. Now, since $c \in A \setminus \{d\}$, the required p_c and q_c are obtained by running the same arguments again, taking $p := p_d,\ q := q_d,\ \sigma := \varepsilon,\ G := \{dd\}$ and $H := \{d\}$. □

From this lemma we deduce that no non-testable AGEF property is preserved by a non-IF congruence.

Theorem 3. *Let $\sim$ be a non-IF congruence. Then for any non-testable property* AGEF(σ, G) *there are processes p, q such that $p \sim q$ and $(\sigma, G) \in \mathcal{I}(p) \setminus \mathcal{I}(q)$.*

Proof. Let (σ, G) be a non-testable AGEF property. Pick $b\rho \in G$ such that $\varepsilon, b \notin G$. Let c be an action that does not occur in $\sigma b \rho$. By Lemma 3 there are processes p and q with $\mathcal{A}(p) = \mathcal{A}(q) = \{c\}$, $p \sim q$ and $(\varepsilon, \{cc\}) \in \mathcal{I}(p) \setminus \mathcal{I}(q)$. As $\sim$ is a congruence for $|$ and $\backslash H$, we have $(p \,|\, \sigma b \bar{c} \bar{c} \rho)\backslash\{c\} \sim (q \,|\, \sigma b \bar{c} \bar{c} \rho)\backslash\{c\}$.

Whenever $(q \,|\, \sigma b \bar{c} \bar{c} \rho)\backslash\{c\} \overset{\sigma}{\Longrightarrow} r$, the process r must be of the form $(q' \,|\, b \bar{c} \bar{c} \rho)\backslash\{c\}$ with $q \overset{\varepsilon}{\Longrightarrow} q'$. Since $(\varepsilon, \{cc\}) \notin \mathcal{I}(q)$, we have $q' \overset{cc}{\Longrightarrow}$ and hence $(q' \,|\, b \bar{c} \bar{c} \rho)\backslash\{c\} \overset{b\rho}{\Longrightarrow}$. Thus $(\sigma, G) \notin \mathcal{I}((q \,|\, \sigma b \bar{c} \bar{c} \rho)\backslash\{c\})$.

Since $(\varepsilon, \{cc\}) \in \mathcal{I}(p)$, we have $p \overset{\varepsilon}{\Longrightarrow} p'$ for a process p' with $p' \overset{cc}{\not\Longrightarrow}$. Hence $(p \,|\, \sigma b \bar{c} \bar{c} \rho)\backslash\{c\} \overset{\sigma}{\Longrightarrow} (p' \,|\, b \bar{c} \bar{c} \rho)\backslash\{c\}$, and $(p' \,|\, b \bar{c} \bar{c} \rho)\backslash\{c\} \overset{\nu}{\Longrightarrow}$ only if $\nu = \varepsilon$ or $\nu = b$. It follows that $(\sigma, G) \in \mathcal{I}((p \,|\, \sigma b \bar{c} \bar{c} \rho)\backslash\{c\})$. □

Next, we prove that a congruence $\sim$ either preserves all testable properties (AGEF or otherwise) or does not preserve any proper AGEF property, nor any proper liveness property under the global fairness assumption $\mathcal{F}$.

Lemma 4. *If $\sim$ is a congruence that does not respect all testable properties, then for all $c \in A$ there exist processes p_c, q_c with $\mathcal{A}(p_c) = \mathcal{A}(q_c) = \{c\}$, such that $p_c \sim q_c$ and $(\varepsilon, \{c\}) \in \mathcal{I}(p_c) \setminus \mathcal{I}(q_c)$.*

Proof. Let φ be a testable property that is not preserved by $\sim$. As φ is testable, there is a test process t such that $\forall p \in \mathbb{P}$ one has p **shd** t iff p satisfies φ. As φ is not preserved by $\sim$, there are processes p, q such that $p \sim q$ and $\varphi(q)$ but not $\varphi(p)$. Hence q **shd** t but p $\mathbf{s\not{h}d}$ t. Let $p_\checkmark = (p \,|\, t)\backslash(A \backslash \{\checkmark\})$ and $q_\checkmark = (q \,|\, t)\backslash(A \backslash \{\checkmark\})$. Then q satisfies AGEF$(\varepsilon, \{\checkmark\})$ but p does not, so $(\varepsilon, \{\checkmark\}) \in \mathcal{I}(p_\checkmark) \setminus \mathcal{I}(q_\checkmark)$. $\mathcal{A}(p_\checkmark) = \mathcal{A}(q_\checkmark) = \{\checkmark\}$. As $\sim$ is a congruence for $|$ and $\backslash H$, we have $p_\checkmark \sim q_\checkmark$. The result for actions $c \neq \checkmark$ is obtained in a manner similar to the one used in the proof of Lemma 3. □

Theorem 4. *Let $\sim$ be a congruence that does not respect all testable properties. Then for any proper* AGEF *property* AGEF(σ, G) *there are processes p and q such that $p \sim q$ and $p \notin$* AGEF$(\sigma, G) \supseteq$ AGEF$\uparrow(\sigma, G) \ni q$.

Proof. Let (σ, G) be given, satisfying $\varepsilon \notin G \neq \emptyset$, and take $\rho \in G$. Let c be an action that does not occur in the sequence $\sigma\rho$. By Lemma 4 there are processes p and q with $\mathcal{A}(p) = \mathcal{A}(q) = \{c\}$, $p \sim q$ and $(\varepsilon, \{c\}) \in \mathcal{I}(p) \setminus \mathcal{I}(q)$. Since $\sim$ is a congruence for $|$ and $\backslash H$, we have $(p \,|\, \sigma\bar{c}\rho)\backslash\{c\} \sim (q \,|\, \sigma\bar{c}\rho)\backslash\{c\}$. As in the proof of Theorem 3, we find that $(\sigma, G) \in \mathcal{I}((p \,|\, \sigma\bar{c}\rho)\backslash\{c\}) \setminus \mathcal{I}((q \,|\, \sigma\bar{c}\rho)\backslash\{c\})$ and moreover $p \notin$ AGEF$(\sigma, G) \supseteq$ AGEF$\uparrow(\sigma, G) \ni q$. □

Finally we show that a congruence that fails to respect a safety property does not respect any nontrivial AGEF property.

Lemma 5. *If $\sim$ is a congruence that does not respect all safety properties, then for all $c \in A$ there exist processes p_c, q_c with $\mathcal{A}(p_c) = \mathcal{A}(q_c) = \{c\}$, such that $p_c \sim q_c$ and $c \in \mathcal{T}(p_c) \setminus \mathcal{T}(q_c)$.*

Proof. The congruence $\sim$ violates a safety property $\bigwedge_{\sigma \in G}$AGEF$(\sigma, \emptyset)$, so there must be an $\sigma \in G$ and processes p and q with $p \sim q$ and $(\sigma, \emptyset) \in \mathcal{I}(p) \setminus \mathcal{I}(q)$, i.e. $\sigma \in \mathcal{T}(p) \setminus \mathcal{T}(q)$. Note that $\sigma \neq \varepsilon$: if $\sigma = \varepsilon$ then $(\sigma, \emptyset) \in \mathcal{I}(q)$. By placing p and q in a context $(\_ \,|\, \bar{\sigma}c)\backslash H$ with $c \notin H := \mathcal{A}(p) \cup \mathcal{A}(q)$ we obtain processes p_c, q_c as required. The result for $c \in H$ is obtained just as in the proof of Lemma 3. □

Theorem 5. *Let $\sim$ be a congruence that does not respect all safety properties. Then for any nontrivial property* AGEF(σ, G) *there are processes p and q with $p \sim q$ and $(\sigma, G) \in \mathcal{I}(p) \setminus \mathcal{I}(q)$, i.e. $q \in$* AGEF(σ, G) *and $p \notin$* AGEF(σ, G).

Proof. The case $\sigma = \varepsilon$ follows from Theorem 4, as safety properties are testable and nontrivial properties AGEF(ε, G) are proper. So assume $\sigma \neq \varepsilon$. Let c be an action that does not occur in σ. By Lemma 5 there are processes p, q with $p \sim q$, $\mathcal{A}(p) = \mathcal{A}(q) = \{c\}$ and $c \in \mathcal{T}(p) \setminus \mathcal{T}(q)$. Since $\sim$ is a congruence for $|$ and $\backslash H$, we have $(p \,|\, \bar{c}\sigma)\backslash\{c\} \sim (q \,|\, \bar{c}\sigma)\backslash\{c\}$. Now $(\sigma, G) \in \mathcal{I}((p \,|\, \bar{c}\sigma)\backslash\{c\}) \setminus \mathcal{I}((q \,|\, \bar{c}\sigma)\backslash\{c\})$.□

6 The Recursive Specification Principle

The *Recursive Specification Principle* (RSP) [3] says that systems of guarded recursive equations have unique solutions. Our aim is to characterise $\sim_{\mathcal{I}}$ as the coarsest congruence that respects a proper AGEF property and satisfies RSP. To this end, we only need a simplification of RSP, called RSP*, saying that equations of the form $X = \sigma X + p$ with $\sigma \in (A_\tau)^* \setminus \{\tau\}^*$ have unique solutions. We denote the unique solution of such an equation as $\sigma^* p$. Alternatively, we could introduce $\sigma^*\_$ as an operator on processes, with $\sigma^* p \xrightarrow{a} p'$ iff $p \xrightarrow{a} p'$ or $(\sigma = a\rho$ and $p' = \rho(\sigma^* p))$.

Definition 8. An equivalence relation $\sim$ satisfies RSP* if for all $\sigma \in (A_\tau)^* \backslash \{\tau\}^*$ and processes p, q

$$p \sim \sigma p + q \quad \Rightarrow \quad p \sim \sigma^* q \ .$$

Fair testing congruence [5] does not satisfy RSP*, since $t = a^*0 + a^*ab$ is fair testing congruent to $at + ab$ but not to a^*ab.

7 Impossible Futures Congruence

Just like weak bisimulation equivalence [12] and most other weak equivalences, impossible futures equivalence $\sim_{\mathcal{I}}$ (defined in Sect. 4) fails to be a congruence for the CCS choice operator $+$. We apply the usual fix to this problem: the addition of a root condition. Just like for failures or fair testing equivalence, a one-bit root condition is sufficient: we merely need to distinguish processes that can do an initial τ-step from those that can not.

Definition 9. Processes p, q are *IF-congruent* (notation $p =_{\mathcal{I}} q$) iff $p \sim_{\mathcal{I}} q \wedge (p \xrightarrow{\tau}) \Leftrightarrow (q \xrightarrow{\tau})$.

Theorem 12 in [5] (Theorem 4.8 in the journal preprint) shows that $=_{\mathcal{I}}$ is finer than fair testing equivalence, which is itself finer than trace equivalence. In [14], it is shown that $=_{\mathcal{I}}$ is coarser than weak bisimulation congruence. Since $a + \tau b$ and $\tau(a{+}b){+}\tau b$ are IF congruent but not weakly bisimilar, $=_{\mathcal{I}}$ is strictly coarser. We now show that $=_{\mathcal{I}}$ is a congruence and satisfies RSP*.

Proposition 2. *$=_{\mathcal{I}}$ is a congruence satisfying* RSP (*and hence* RSP*).

Proof. The argument in [14] can be adapted and extended to yield the required result. For example, RSP follows from a standard deductive argument [3] based on the auxiliary finite projection operator and the induction principle AIP. □

Below, we show that $=_{\mathcal{I}}$ is the coarsest of all equivalences that (1) respect a proper AGEF property, (2) are congruences for the CCS parallel composition, restriction, choice and prefixing, (3) satisfy RSP*, and (4) are coarser than an equivalence $\leftrightarroweq_{\max}$. We show that in (1) it does not matter which AGEF property we take, so it could for instance be the property AGEF(ε, A), saying that the system can always do a first visible action, i.e. has no initial livelock or deadlock.

Instead of (1) we may also require that the equivalence respects a proper property $\mathrm{AGAF}_{\mathcal{F}}(\sigma, G) = \mathrm{AGEF}{\uparrow}(\sigma, G)$, i.e. a conditional liveness property assuming global fairness. We do not know whether our result is valid without (4), but we use it in our proofs. Below, we establish our result with *rooted weak bisimilarity* in the rôle of $\leftrightarroweq_{\max}$. In CCS, weak bisimilarity is the equivalence of choice for e.g. comparing specifications and implementations. This equivalence is not a congruence for all CCS operators, but it becomes so after extending it with a root condition, yielding *rooted weak bisimilarity* (*observational congruence*) [12].

Definition 10. A relation $R \subseteq \mathbb{P} \times \mathbb{P}$ is called a *weak bisimulation* if it satisfies for all processes p, q with $p \mathrel{R} q$ and for all $\sigma \in A^*$:
– for all p' with $p \overset{\sigma}{\Longrightarrow} p'$ there exists q' such that $q \overset{\sigma}{\Longrightarrow} q'$ and $p' \mathrel{R} q'$,
– for all q' with $q \overset{\sigma}{\Longrightarrow} q'$ there exists p' such that $p \overset{\sigma}{\Longrightarrow} p'$ and $p' \mathrel{R} q'$.
Processes p, q are called *weakly bisimilar* (notation $p \leftrightarroweq_w q$) if there exists a weak bisimulation R such that $p \mathrel{R} q$. They are called *rooted weakly bisimilar* (notation $p \leftrightarroweq_{rw} q$) if they satisfy the additional *root conditions*:
– for all p' with $p \xrightarrow{\tau} p'$ there exists q' such that $q \overset{\varepsilon}{\Longrightarrow}\xrightarrow{\tau} q'$ and $p' \mathrel{R} q'$,
– for all q' with $q \xrightarrow{\tau} q'$ there exists p' such that $p \overset{\varepsilon}{\Longrightarrow}\xrightarrow{\tau} p'$ and $p' \mathrel{R} q'$.

Write $\tau_a(p)$ for $(p \,|\, \bar{a}^*0)\backslash\{a\}$. The operator τ_a renames action a into τ (and disables $\bar{a}$). By construction of rooted weak bisimulations, one can deduce the following equivalences for $a \in A_\tau$, $\sigma \in \{a, \tau\}^* \setminus \{\tau\}^*$ and processes p:

$$\mathrm{T1}: \quad a\tau p \leftrightarroweq_{rw} ap \qquad \mathrm{KFAR}: \quad \tau_a(\sigma^* p) \leftrightarroweq_{rw} \tau\tau_a(p).$$

KFAR [3] identifies divergent processes (capable of an infinite sequence of τ steps) and non-divergent ones. We prove a lemma for later use.

Lemma 6. Let p, q, r be processes with $p \overset{\varepsilon}{\Longrightarrow}\xrightarrow{\tau} r \leftrightarroweq_w q$. Then $p \leftrightarroweq_{rw} p + \tau q$.

Proof. Let R be $\leftrightarroweq_w$ augmented with the pair $(p, p + \tau q)$. This is a weak bisimulation satisfying the root conditions. □

A process equivalence $\sim$ is called a *w-congruence* iff it is a congruence w.r.t. the CCS parallel composition, restriction, choice and prefixing, and is coarser than $\leftrightarroweq_{rw}$. We proceed to characterise IF congruence as the coarsest w-congruence that satisfies RSP* and preserves some arbitrary proper AGEF property. Recall that a congruence $\sim$ is called *non-IF* iff there exist processes p, q with $p \sim q$ such that $\mathcal{I}(p) \neq \mathcal{I}(q)$. We start with some lemmas.

Lemma 7. *If $\sim$ is a non-IF w-congruence, then for any $c \in A$*

$$\tau c(\tau c + \tau) \sim \tau c(\tau c + \tau) + \tau c$$

Proof. Start with the proof of Lemma 3, up to $(U_0 \,|\, p)\backslash H \sim (U_0 \,|\, q)\backslash H$. Next we define the relation R in Table 1. This relation is a weak bisimulation, which can be verified by checking all steps. The crucial argument is that for each q' satisfying $q \overset{\sigma}{\Longrightarrow} q'$ there exists a $\rho \in G$ such that $q' \overset{\rho}{\Longrightarrow}$. The relation satisfies the root conditions, so

$$\tau c(\tau c + \tau) \leftrightarroweq_{rw} (U_0 \,|\, q)\backslash H.$$

simple term	merge term	condition
$\tau c(\tau c + \tau)$	$(U_0 \mid q)\backslash H$	*true*
$c(\tau c + \tau)$	$(U_k \mid q')\backslash H$	$0 \leq k \leq m \wedge q \overset{a_1 \ldots a_k}{\Longrightarrow} q'$
$c(\tau c + \tau)$	$(V \mid q')\backslash H$	$q \overset{*}{\Longrightarrow} q'$
$\tau c + \tau$	$(\tau c + \tau \mid q')\backslash H$	$q \overset{*}{\Longrightarrow} q'$
c	$(c \mid q')\backslash H$	$q \overset{*}{\Longrightarrow} q'$
0	$(0 \mid q')\backslash H$	$q \overset{*}{\Longrightarrow} q'$
$c(\tau c + \tau)$	$(cW \mid q')\backslash H$	$q \overset{\sigma}{\Longrightarrow} q'$
$\tau c + \tau$	$(W \mid q')\backslash H$	$q \overset{\sigma}{\Longrightarrow} q'$
$\tau c + \tau$	$(\bar{\vartheta}(\tau c + \tau) \mid q')\backslash H$	$\exists \nu \neq \varepsilon \cdot \nu\vartheta \in G \wedge q \overset{\sigma}{\Longrightarrow}\overset{\nu}{\Longrightarrow} q' \overset{\vartheta}{\Longrightarrow}$
0	$(\bar{\vartheta}(\tau c + \tau) \mid q')\backslash H$	$\exists \nu \neq \varepsilon \cdot \nu\vartheta \in G \wedge q \overset{\sigma}{\Longrightarrow}\overset{\nu}{\Longrightarrow} q' \overset{\vartheta}{\not\Longrightarrow}$

Table 1. The relation R

As $(\sigma, G) \in \mathcal{I}(p)$, there must be a process p' such that $p \overset{\sigma}{\Longrightarrow} p'$ and $\forall \rho \in G \cdot p' \overset{\rho}{\not\Longrightarrow}$. It is trivial to construct a weak bisimulation showing that $(cW \mid p')\backslash H \mathrel{\underline{\leftrightarrow}}_w c$ and hence $(U_0 \mid p)\backslash H \overset{\varepsilon}{\Longrightarrow}\overset{\tau}{\longrightarrow} (cW \mid p')\backslash H \mathrel{\underline{\leftrightarrow}}_w c$. Using Lemma 6, this implies that $(U_0 \mid p)\backslash H \mathrel{\underline{\leftrightarrow}}_{rw} (U_0 \mid p)\backslash H + \tau c$. Therefore, as $\sim$ is a congruence, we have $\tau c(\tau c + \tau) \mathrel{\underline{\leftrightarrow}}_{rw} (U_0 \mid q)\backslash H \sim (U_0 \mid p)\backslash H \mathrel{\underline{\leftrightarrow}}_{rw} (U_0 \mid p)\backslash H + \tau c \sim (U_0 \mid q)\backslash H + \tau c \mathrel{\underline{\leftrightarrow}}_{rw} \tau c(\tau c + \tau) + \tau c$. Since $\sim$ is transitive and coarser than $\mathrel{\underline{\leftrightarrow}}_{rw}$ we obtain the desired result for $c \in A\backslash H$. For $c \in H$, we proceed as in the proof of Lemma 3. □

Lemma 8. *If $\sim$ is a non-IF w-congruence, then for all processes P, Q and $a \in A$,*

$$\tau a(\tau P + Q) \sim \tau a(\tau P + Q) + \tau a P. \tag{1}$$

Proof. Pick $c \notin \mathcal{A}(P) \cup \mathcal{A}(Q) \cup \{a\}$ and place the processes equated by Lemma 7 in the context $(\_ \mid \tau a \bar{c}(\tau P + \bar{c}(\tau P + Q))\backslash\{c\})$. □

We now use RSP* to show equivalence of processes with and without deadlock.

Lemma 9. *Let $\sim$ be a non-IF w-congruence satisfying RSP*. Then for any process Q we have $\tau(\tau Q + \tau) \sim \tau Q$.*

Proof. Choose Q and $a \notin \mathcal{A}(Q)$. Set $P = a^*0$ and $R = (\tau a)^*(\tau Q + \tau.P)$, so

$$(2)\ P \sim aP \qquad\qquad (3)\ R \sim \tau P + \tau Q + \tau a R.$$

Since $\tau a R = \tau a(\tau P + \tau Q + \tau a R)$, (1) yields $\tau a R \sim \tau a R + \tau a P$. Hence,

$$\tau a R + \tau Q \sim \tau a R + \tau a P + \tau Q \overset{2}{\sim} \tau a R + \tau P + \tau Q \overset{3}{\sim} R.$$

Since $R \sim \tau a R + \tau Q$, RSP* yields $R \sim (\tau a)^* \tau Q$. So, since $\sim$ is a congruence,

$$\tau.(\tau Q + \tau\tau 0) \mathrel{\underline{\leftrightarrow}}_{rw} \tau_a((\tau a)^*(\tau Q + \tau(a^*0))) \sim \tau_a((\tau a)^* \tau Q) \mathrel{\underline{\leftrightarrow}}_{rw} \tau\tau Q$$

applying KFAR and $\tau_a(Q) = Q$. Using T1, we obtain $\tau(\tau Q + \tau) \sim \tau Q$. □

Fair testing [5] is a non-IF w-congruence that preserves all testable AGEF properties. However, non-IF w-congruences satisfying RSP do not preserve *any* proper AGEF property, nor any nontrivial property $\text{AGAF}_{\mathcal{F}}(\sigma, G) = \text{AGEF}{\uparrow}(\sigma, G)$.

Theorem 6. *Let $\sim$ be a non-IF w-congruence satisfying RSP*. Then for any proper* AGEF *property* AGEF(σ, G) *there are processes p and q such that $p \sim q$ and $p \notin$* AGEF$(\sigma, G) \supseteq$ AGEF$\uparrow(\sigma, G) \ni q$.

Proof. Let (σ, G) be proper, i.e. $\varepsilon \notin G \neq \emptyset$, and pick $\rho \in G$. Set $Q = \rho$. By Lemma 9, the fact that $\sim$ is a congruence for the prefix operator, and by identity T1, we have $p = \sigma(\tau Q + \tau) \sim \sigma Q = q$. Clearly, $(\sigma, G) \in \mathcal{I}(p) \setminus \mathcal{I}(q)$ and $p \notin$ AGEF$(\sigma, G) \supseteq$ AGEF$\uparrow(\sigma, G) \ni q$. □

This theorem also gives a partial answer to van Glabbeek's first problem in [1]: what is the coarsest congruence $\sim$ satisfying RSP and respecting deadlock/livelock traces?

Definition 11. A sequence $\sigma \in A^*$ is a *deadlock/livelock trace* of a process p if $\exists p' \cdot p \overset{\sigma}{\Longrightarrow} p' \wedge \mathcal{T}(p') = \{\varepsilon\}$. A process equivalence $\sim$ *respects deadlock/livelock traces* iff $p \sim q$ implies that p and q have the same deadlock/livelock traces.

Note that σ is a deadlock/livelock trace of p iff $(\sigma, A) \in \mathcal{I}(p)$. This is the negation of a proper AGEF property, so from Theorem 6 we deduce that $=_{\mathcal{I}}$ is the coarsest w-congruence satisfying RSP and respecting deadlock/livelock traces. The answer is partial, since there exist congruences that respect deadlock/livelock traces but are not coarser than $\leftrightarroweq_{rw}$, such as branching bisimilarity [10]. So the existence of (non-w) congruences respecting deadlock/livelock traces and satisfying RSP that are incomparable with $=_{\mathcal{I}}$ is conceivable.

The results in this section can be generalised to a divergence sensitive setting. Divergence of a process is the possibility to perform an infinite sequence of τ steps. By only allowing to relate divergent processes to divergent processes, one defines *weak bisimulation congruence with explicit divergence* (notation $\leftrightarroweq^{\Delta}_{rw}$) [10]. We now relax the requirement in Theorem 6 that $\sim$ must be a *w*-congruence. Instead of requiring $\sim$ to be coarser than $\leftrightarroweq_{rw}$ we merely require it to be coarser than $\leftrightarroweq^{\Delta}_{rw}$, i.e. we use $\leftrightarroweq^{\Delta}_{rw}$ for $\leftrightarroweq_{\max}$. The proof, which is omitted due to lack of space, requires a slight extension of RSP*, still implied by RSP.

Theorem 7. $=_{\mathcal{I}}$ *is the coarsest congruence coarser than $\leftrightarroweq^{\Delta}_{rw}$, satisfying* RSP *and respecting deadlock/livelock traces.*

This is a useful addition, because many equivalences in the linear time – branching time spectrum of [10], such as the CSP failures equivalence [6], fail to be coarser than $\leftrightarroweq_{rw}$, although virtually all are coarser than $\leftrightarroweq^{\Delta}_{rw}$. Moreover, the few equivalences from [10] that are not coarser than $\leftrightarroweq^{\Delta}_{rw}$ are certainly finer than $\sim_{\mathcal{I}}$ or do not respect deadlock/livelock traces; thus no known equivalence is ruled out by the restriction "coarser than $\leftrightarroweq^{\Delta}_{rw}$".

8 Conclusion

We have discussed the connection between AGAF properties, expressing (conditional) lifeness requirements, AGEF properties, expressing not-doomed-to-fail requirements, and impossible futures congruence, which we have characterised, under a mild side-condition, as the coarsest congruence that allows verification of testable AGEF properties—or AGAF properties under a global fairness assumption—and assigns unique solutions to guarded recursive equations. Thus, where such properties are deemed important, equational system verification [2] requires a semantic equivalence at least as fine as impossible futures congruence.

The fact that we have used the operators of CCS bears no relevance. We could have used any process calculus that allows action prefix, choice, communication merge, restriction and abstraction, such as CSP, ACP, LOTOS and many others.

Acknowledgement
We are grateful for the support and advice of our colleague Bas Luttik.

References

1. L. Aceto (moderator) (2003): *Some open problems in Process Algebra.* `http://www.cs.auc.dk/~luca/BICI/open-problems.html`
2. J.C.M. Baeten, editor (1990): *Applications of Process Algebra.* Cambridge Tracts in Theoretical Computer Science 17. Cambridge University Press.
3. J.C.M. Baeten, J.A. Bergstra & J.W. Klop (1987): *On the consistency of Koomen's fair abstraction rule.* *Theoretical Computer Science* 51(1/2), pp. 129–176.
4. B. Bloom, S. Istrael & A. Meyer (1995): *Bisimulation Can't Be Traced.* *Journal of the ACM* 42(1), pp. 232–268.
5. E. Brinksma, A. Rensink & W. Vogler (1995): *Fair Testing.* In *Proceedings CONCUR '95* (I. Lee & S.A. Smolka, eds.), LNCS 962, Springer, pp. 311–327. Journal preprint: `http://eprints.eemcs.utwente.nl/1623/01/submitted.pdf`
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.
7. E.M. Clarke & E.A. Emerson (1981): *Design and synthesis of synchronization skeletons using branching-time temporal logic.* In Proceedings workshop on *Logic of Programs* (D. Kozen, ed.), LNCS 131, Springer, pp. 52–71.
8. R. De Nicola & F.W. Vaandrager (1995): *Three logics for branching bisimulation.* *Journal of the ACM* 42(2), pp. 458–487.
9. N. Francez (1986): *Fairness.* Springer.
10. R.J. van Glabbeek (1993): *The Linear Time – Branching Time Spectrum II: The semantics of sequential systems with silent moves (extended abstract).* In *Proceedings CONCUR '93* (E. Best, ed.), LNCS 715, Springer, pp. 66–81.
11. R.J. van Glabbeek (2005): *A Characterisation of Weak Bisimulation Congruence.* In *Processes, Terms and Cycles: Steps on the Road to Infinity:* Essays Dedicated to J.W. Klop on the Occasion of His 60th Birthday (A. Middeldorp, V. van Oostrom, F. van Raamsdonk & R. de Vrijer, eds.), LNCS 3838, Springer, pp. 26-39.
12. R. Milner (1990): *Communication and Concurrency*, Prentice-Hall International, Englewood Cliffs, 1989. An earlier version appeared as *A Calculus of Communicating Systems*, LNCS 92, Springer-Verlag, 1980.
13. W. Vogler (1992): *Modular Construction and Partial Order Semantics of Petri nets.* LNCS 625, Springer.
14. M. Voorhoeve & S. Mauw (2001): *Impossible Futures and Determinism.* *Information Processing Letters* 80(1), Elsevier, pp. 51–58.