On the bisimulation hierarchy of state-to-function transition systems

On the bisimulation hierarchy of state-to-function transition systems

Marino Miculan and Marco Peressotti

Dept. of Mathematics, Computer Science and Physics, University of Udine, Italy
marino.miculan@uniud.it   marco.peressotti@uniud.it

Abstract Weighted labelled transition systems (WLTSs) are an established (meta-)model aiming to provide general results and tools for a wide range of systems such as non-deterministic, stochastic, and probabilistic systems. In order to encompass processes combining several quantitative aspects, extensions of the WLTS framework have been further proposed, *state-to-function transition systems* (FuTSs) and *uniform labelled transition systems* (ULTraSs) being two prominent examples. In this paper we show that this hierarchy of meta-models collapses when studied under the lens of bisimulation-coherent encodings.

1 Introduction

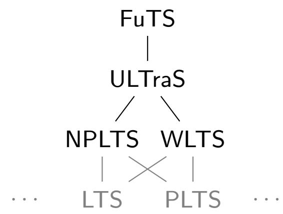
Weighted labelled transition systems (WLTSs) [10] is a meta-model for systems with quantitative aspects: transitions $P \xrightarrow{a,w} Q$ are labelled with *weights* w, taken from a given monoidal weight structure. Many computational aspects can be captured just by changing the underlying weight structure: weights can model probabilities, resource costs, stochastic rates, *etc.*; as such, WLTSs are a generalisation of labelled transition systems (LTSs), probabilistic systems (PLTSs) [6], stochastic systems [9], among others. Definitions and results developed in this setting instantiate to existing models, thus recovering known results and discovering new ones. In particular, the notion of *weighted bisimulation* [10] in WLTSs coincides with strong bisimulation for all the aforementioned models.

In the wake of these encouraging results, other meta-models have been proposed aiming to cover an even wider range of computational models and concepts. *Uniform labelled transition systems* (ULTraSs) [2] are systems whose transitions have the form $P \xrightarrow{a} \phi$, where ϕ is a *weight function* assigning weights to states; hence, ULTraSs can be seen both as a non-deterministic extension of WLTSs and as a generalisation of Segala's probabilistic systems [18] (NPLTSs). In [14, 15] a (coalgebraically derived) notion of bisimulation for ULTraSs is presented and shown to precisely capture bisimulations for weighted and Segala systems. *Function-to-state transition systems* (FuTSs) were introduced in [5] as a generalisation of the above, of IMC [8], and other models. Later, [11] defines a (coalgebraically derived) notion of bisimulation for FuTSs which instantiates to known bisimulations for the aforementioned models.

Given all these meta-models, it is natural to wonder about their *expressivity*. We should consider not only the class of systems these frameworks can represent, but also *whether* these representations are faithful with respect to the properties we are interested in. Intuitively, a meta-model M is *subsumed by* M' *according to a property* P if any system S which is an instance of M with the property P, is also an instance of M' preserving P.

In this paper, we aim to classify meta-models according their ability to correctly express *strong bisimulation*. Therefore, in our contest a meta-model M is *subsumed* by M' if any system S which is an instance of M, is also an instance of M' preserving strong bisimulation.

Previous work [2, 10, 11, 14, 15] have shown that, according to this order, each of the meta-models mentioned above subsumes the previous ones, thus forming the hierarchy shown aside. Still, an important question is open: is any of these meta-models *strictly more expressive* than others? In this work we address this question, proving that this is not the case: the black part of the hierarchy collapses!



In order to formally capture the notion of "expressiveness order between system classes with respect to (strong) bisimulation", we introduce the notion of *reduction* between classes of systems. Although the driving motivation is the study of the FuTS hierarchy under the lens of bisimulation, the notion of reduction is more general and, as defined in this work, can be used to study any class of state-based transition systems. In fact, all the constructions and results are developed abstracting from the "shape" of computation under scrutiny.

Synopsis Section 2 recalls an abstract and uniform account of transition systems on discrete state spaces, akin to [17]. Section 3 presents a general construction for extending equivalence relations over sets of states to sets of behaviours. Building on this relational extension, Section 4 provides a characterisation of (strong) bisimulations in a modular fashion. The notion of reduction is introduced in Section 5, along with general reductions. In Section 6 we provide a reduction from the category of FuTSs to the category of WLTSs together with intermediate reductions for special cases of FuTSs such as ULTraS, nested FuTSs, and combined FUTSs. Final remarks are in Section 7 and omitted proofs in Appendix A.

2 Discrete transition systems

For an alphabet A and set of states X, the function space X^A is understood as the set of all possible behaviours characterising deterministic input over A. In this context, a transition system exposing this computational behaviour is precisely described by a function $\alpha\colon X \to X^A$ mapping each state $x \in X$ to some element in X^A. For a function $f\colon X \to Y$ and $\phi \in X^A$, the assignment $\phi \mapsto f \circ \phi$ defines a function $(f)^A\colon X^A \to Y^A$ that extends the action of f from state spaces X and Y to behaviours defined over them in a coherent way. A function $f\colon X \to Y$ between the state spaces of systems (say, $\alpha\colon X \to X^A$ and

$\beta\colon Y \to Y^A$) preserves and reflect their structure whenever $f^A \circ \alpha = \beta \circ f$. Since they preserve and reflect the transition structure of systems, these functions are called *homomorphisms* (which are functional bisimulations, *cf.* [17, Thm. 2.5]).

All the structures and observations described in the above example stem from a single information: the "type" of the behaviour under scrutiny. This is well understood as an *endofunctor* over the category of state spaces [17] — in this context, the category of sets and functions.

Non-deterministic transitions are captured by the powerset endofunctor $\mathcal{P}$ mapping each set X its powerset $\mathcal{P}X$ and function f to its inverse image $\mathcal{P}f$ *i.e.* the function given by the assignment $Z \mapsto \{f(z) \mid z \in Z\}$. Since subsets are functions weighting elements over the monoid $\mathbb{B} = (\{\mathtt{tt}, \mathtt{ff}\}, \vee, \mathtt{ff})$, the above readily extends to quantitative aspects (such as probability distributions, stochastic rates, delays, etc.) by simply considering other a non-trivial abelian monoids[1] [10, 12, 15]. This yields the endofunctor $\mathcal{F}_M$ which assigns

– to each set X the set $\{\phi\colon X \to M \mid |\{x \mid \phi(x) \neq 0\}| \in \mathbb{N}\}$;
– to each function $f\colon X \to Y$ the map $(\mathcal{F}_M f)(\phi) = \lambda y \in Y. \sum_{x : f(x) = y} \phi(x)$. (summation is well defined because ϕ is finitely supported, by above).

Probabilistic computations are a special case of the above where weight functions are distributions (*cf.* [10]) and are captured by the endofunctor $\mathcal{D}$ given on each set X as $\mathcal{D}X = \{\phi \in \mathcal{F}_{[0,\infty)}X \mid \sum \phi(x) = 1\}$ and on each function f as $\mathcal{F}_{[0,\infty)}f$.

From this perspective, $\mathcal{D}$ can be thought as a sort of "subtype" of $\mathcal{F}_{[0,\infty)}$. This situation is formalised by means of (component-wise) injective natural transformations (herein *injective transformations*). Composition and products of natural transformations are component-wise and the class of injective ones is closed under such operations. In general, for an injective transformation μ and a n endofunctor T, μ_T is again injective but $T\mu$ may not be so. The latter is injective given that T *preserves injective maps i.e.* Tf is injective whenever f is injective. All the examples listed in this paper meet this mild assumption.

Lemma 1. *Any composition and product of Id, $\mathcal{P}$, $\mathcal{F}_M$ preserve injections.*

Example 1. The endofunctor $\mathcal{P}\mathcal{F}_M$ models the alternation of non-deterministic steps with quantitative aspects captured by $(M, +, 0)$. There is an injective transformation $\eta\colon Id \to \mathcal{P}$ whose components are given by the mapping $x \mapsto \{x\}$ and hence, by composition, $\eta_{\mathcal{F}_M} : \mathcal{F}_M \to \mathcal{P}\mathcal{F}_M$ is an injective transformation. □

Definition 1. *For an endofunctor T over* Set*, a transition system of type T (T-*system*) is a pair (X, α) where X is the set of states (*carrier*) and $\alpha\colon X \to TX$ is the* transition map*. For (X, α) and (Y, β) T-systems, a T-*homomorphism *from the former to the latter is a function $f\colon X \to Y$ s.t. $Tf \circ \alpha = f \circ \beta$.*

Since system homomorphism composition is defined in terms of composition of the underlying functions on carriers it is immediate to check that the operation is associative and has identities. Therefore, any class of systems together with their homomorphisms defines a category.

[1] An abelian monoid is a set M equipped with an associative and commutative binary operation $+$ and a unit 0 for $+$; such structure is called trivial when M is a singleton.

We adopt the following notational conventions. A transition system (X, α) is referred by its transition map only; in this case its carrier is written $\mathrm{car}(\alpha)$. Homomorphisms are denoted by their underlying function. Categories of systems are written using sans serif font with $\mathsf{Sys}(T)$ being the category of all T-systems and T-homomorphisms and $\mathsf{C}|_T$ its subcategory of systems in the category C.

Example 2 (LTSs). For a set A of labels, labelled transition systems are $(\mathcal{P}-)^A$-systems, and image finite LTSs $(\mathcal{P}_f-)^A$-systems [17]. Hereafter let LTS denote the category of all image-finite labelled transition systems and let $\mathsf{LTS}(A) \triangleq \mathsf{Sys}((\mathcal{P}_f-)^A)$ be its subcategory of systems labelled over A. □

Example 3 (WLTSs). For a set of labels A and an abelian monoid M, weighted labelled transition systems are characterised by the endofunctor $(\mathcal{F}_M-)^A$ [10] and hence form the category $\mathsf{WLTS}(A, M) \triangleq \mathsf{Sys}((\mathcal{F}_M-)^A)$ *i.e.* the (A, M)-indexed component of WLTS, the category of all WLTSs. When the monoid $\mathbb{B}$ of boolean values under disjunction is considered, $\mathsf{WLTS}(A, \mathbb{B})$ is $\mathsf{LTS}(A)$. □

Example 4 (ULTraSs). We adopt the presentation of ULTraSs given in [14, 15]. For a set of labels A and an abelian monoid M, uniform labelled transition systems are characterised by the endofunctor $(\mathcal{P}\mathcal{F}_M-)^A$; image finite ULTraSs by $(\mathcal{P}_f\mathcal{F}_M-)^A$. We denote by ULTraS the category of all image-finite ULTraSs and by $\mathsf{ULTraS}(A, M)$ its subcategory of systems with labels in A and weights in M. WLTSs can be cast to ULTraSs by means of the injective transformation $(\eta_{\mathcal{F}_M})^A$ described in Example 1. These ULTraSs are called in [2] *functional.* □

Example 5 (FuTSs). FuTSs are T-systems for T generated by the grammar

$$T ::= (S-)^A \mid T \times (S-)^A \qquad S ::= \mathcal{F}_M \mid \mathcal{F}_M \circ S$$

where A and M range over (non-empty) sets of labels and (non-trivial) abelian monoids, respectively. Any such endofunctor is equivalently described by:

$$(\mathcal{F}_{\vec{M}} f)^{\vec{A}} \triangleq \textstyle\prod_{i=0}^n (\mathcal{F}_{\vec{M}_i} f)^{A_i} \qquad \text{and} \qquad (\mathcal{F}_{\vec{M}_i} f)^{A_i} \triangleq (\mathcal{F}_{M_{i,0}} \dots \mathcal{F}_{M_{i,m_i}} f)^{A_i}$$

for $\vec{A} = \langle A_0, \dots, A_n \rangle$ a sequence of non-empty sets, $\vec{M}_i = \langle M_{i,0}, \dots, M_{i,m_i} \rangle$ a sequence of non-trivial abelian monoids, and $\vec{M} = \langle \vec{M}_0, \dots, \vec{M}_n \rangle$ [12,15]. For any $\vec{A}$ and $\vec{M}$ as above define $\mathsf{FuTS}(\vec{A}, \vec{M})$ as $\mathsf{Sys}((\mathcal{F}_{\vec{M}}-)^{\vec{A}})$. Clearly, $\mathsf{FuTS}(\langle A \rangle, \langle M \rangle)$ and $\mathsf{FuTS}(\langle A \rangle, \langle B, M \rangle)$ coincide with $\mathsf{WLTS}(A, M)$ and $\mathsf{ULTraS}(A, M)$, respectively. Then, LTS, WLTS, and ULTraS are subcategories of FuTS, the category of all FuTSs. For $\vec{M} = \langle \langle M_{0,0}, \dots, M_{0,m_0} \rangle, \dots, \langle M_{n,0} \dots M_{n,m_n} \rangle \rangle$ as above, recall from [12] that a FuTS over $\vec{M}$ is called: *nested* if $n = 0$, *combined* if $m_i = 0$ for each $i \in \{0, \dots, n\}$, and *simple* if it is both combined and nested. □

3 Equivalence extensions

Several definitions of bisimulation found in literature use (more or less explicitly) some sort of extension of equivalence relations from state spaces to behaviours over these spaces. For instance, in [18] two probability distributions are considered equivalent with respect to an equivalence relation R on their domain if they assign the same probability to any equivalence class induced by R:

$$\phi \equiv_R \psi \overset{\triangle}{\iff} \forall C \in X/R \ \left(\sum_{x \in C} \phi(x) = \sum_{x \in C} \psi(x)\right).$$

This section defines equivalence extensions for arbitrary endofunctors (over Set) and studies how constructs such as composition or products reflect on these extensions, providing some degree of modularity.

Definition 2. *For an equivalence relation R on X its* T-extension *is the equivalence relation R^T on TX such that $\phi \; R^T \; \psi \overset{\triangle}{\iff} (T\kappa)(\phi) = (T\kappa)(\psi)$ where $\kappa\colon X \to X/R$ is the canonical projection to the quotient induced by R.*

As an example, let us consider the endofunctor $(-)^A$ describing deterministic inputs on A: the resulting extension for an equivalence relation R relates functions mapping the same inputs to states related by R. Formally:

$$\phi \; R^{(-)^A} \; \psi \iff \kappa \circ \phi = \kappa \circ \psi \iff \forall a \in A \ (\phi(a) \; R \; \psi(a)).$$

Extensions for $\mathcal{P}$ are precisely "subset closure" of relations (*cf.* [15]) and relate all and only those subsets for which the given relation is a correspondence. Formally:

$$\begin{aligned} Y \; R^{\mathcal{P}} \; Z &\iff \{\kappa(y) \mid y \in Y\} = \{\kappa(z) \mid z \in Z\} \\ &\iff (\forall y \in Y \exists z \in Z (y \; R \; z)) \wedge (\forall z \in Z \exists y \in Y (y \; R \; z)) \end{aligned}$$

Extension for $\mathcal{F}_M$ are generalise the subset closure to multisets and relate only weight functions assigning the same cumulative weight to each equivalence class induced by R: $\phi \; R^{\mathcal{F}_M} \; \psi \iff \forall C \in X/R \left(\sum_{x \in C} \phi(x) = \sum_{x \in C} \psi(x)\right)$ In particular, $R^{\mathcal{D}}$ is precisely Segala's equivalence $\equiv_R$ [18].

Consider extensions for the endofunctor $(\mathcal{P}-)^A$ describing LTSs:

$$\phi \; R^{(\mathcal{P}-)^A} \; \psi \iff \forall a \in A \begin{pmatrix} (\forall y \in \phi(a) \, \exists z \in \psi(a) \, (y \; R \; z)) \wedge \\ (\forall z \in \psi(a) \, \exists y \in \phi(a) \, (y \; R \; z)) \end{pmatrix}$$

Clearly, $R^{\mathcal{P}(-)^A}$ can be equivalently written as

$$\phi \; R^{\mathcal{P}(-)^A} \; \psi \iff \forall a \in A (\phi(a) \; R^{\mathcal{P}} \; \psi(a))$$

which suggests some degree of modularity in the definition of extensions to composite endofunctors. In general, this kind of reformulations is not possible since for arbitrary endofunctors T ans S, it holds only that $\phi \left(R^S\right)^T \psi \implies \phi \; R^{T \circ S} \; \psi$. The converse implication holds whenever T preserves injections.

Lemma 2. $\left(R^S\right)^T \subseteq R^{T \circ S}$ *and* $\left(R^S\right)^T \supseteq R^{T \circ S}$, *given T preserves injections.*

Endofunctors modelling inputs, such as $(-)^A$ and $(\mathcal{P}_f-)^A$, can be seen as products (in these cases as powers) of endofunctors indexed over the input space A. As suggested by the above examples, for product endofunctors it holds that:

$$\phi \; R^{(\prod T_i)} \; \psi \iff \forall i \in I (\pi_i(\phi) \; R^{T_i} \; \pi_i(\psi))$$

where $\pi_i \colon \prod T_i X \to T_i X$ is the projection on the i-th component of the product.

Lemma 3. *For $I \neq \emptyset$ and $\{T_i\}_{i \in I}$, $R^{\left(\prod_{i\in I} T_i\right)} \cong \prod_{i \in I} R^{T_i}$.*

FuTSs offer an instance of the above result: the endofunctor $(\mathcal{F}_{\vec{M}}-)^{\vec{A}}$ modelling FuTSs over $\vec{M} = \langle \vec{M}_0, \ldots, \vec{M}_n \rangle$ and $\vec{A} = \langle A_0; \ldots; A_n \rangle$ is a product indexed over $\{(i,a) \mid i \leq n \wedge a \in A_i\}$. Thus, the extension $R^{(\mathcal{F}_{\vec{M}}-)^{\vec{A}}}$ is described by:

$$\phi \; R^{(\mathcal{F}_{\vec{M}}-)^{\vec{A}}} \; \psi \iff \forall i \leq n \forall a \in A_i (\phi_i(a) \; R^{\mathcal{F}_{\vec{M}_i}} \; \psi_i(a)).$$

For an equivalence relation R define its restriction to X as the equivalence relation $R|_X \triangleq R \cap (X \times X)$. Both $(R|_X)^T$ and $R^T|_{TX}$ are equivalence relations over the set of T-behaviours for X and, in general, the former is finer than the latter, unless T preserves injections—in such case, the two coincide.

Lemma 4. *For R and equivalence relation on Y and $X \subseteq Y$, $(R|_X)^T \subseteq R^T|_{TX}$, and, provided T preserves injections, $(R|_X)^T \supseteq R^T|_{TX}$.*

Intuitively, this result allows us to encode multiple steps sharing the same computational aspects as single steps at the expense of bigger state spaces. In fact, it follows that $(R|_X)^{T^{n+1}} = R^T|_{T^n X}$, assuming T preserves injections.

Lemma 5. *Let $\mu \colon T \to S$ be an injective natural transformation. For R an equivalence relation on X, $\phi \; R^T \; \psi \iff \mu_X(\phi) \; R^S \; \mu_X(\psi)$.*

4 Bisimulations

In this section we give a general definition of bisimulation based on the notion of equivalence relation extension introduced above. This approach is somehow modular, as the definition reflects the structure of the endofunctors characterising systems under scrutiny. This allows to extend results developed in Section 3 to bisimulation and, in Section 5, to reductions.

Definition 3. *An equivalence relation R is a* strong T-bisimulation *(herein, bisimulation) for a T-system α iff $x \; R \; x' \implies \alpha(x) \; R^T \; \alpha(x')$. We denote by* $\mathrm{bis}(\alpha)$ *the set of all bisimulations for the system α.*

The notion of bisimulations as per Definition 3 coincides with Aczel-Mendler's notion of *precongruence* [1].

Definition 4. *An equivalence relation R on X is a (Aczel-Mendler) precongruence for $\alpha \colon X \to TX$ iff, for any two functions $f, f' \colon X \to Y$ such that $x \; R \; x' \implies f(x) = f'(x')$ it holds that $x \; R \; x' \implies (Tf \circ \alpha)(x) = (Tf' \circ \alpha)(x')$.*

Theorem 1. *For α a T-system, every strong T-bisimulation for α is an AM-precongruence and* vice versa.

Bisimulations for systems considered in this paper are known to be *kernel bisimulations* (*cf.* [10, 12, 15, 17]) *i.e.* kernels of functions carrying homomorphisms from systems under scrutiny [19]. These can be intuitively thought as defining *refinement systems* over the equivalence classes they induce.

Definition 5. *A relation R on X is a kernel bisimulation for $\alpha\colon X \to TX$ iff there is $\beta\colon Y \to TY$ and $f\colon \alpha \to \beta$ s.t. R is the kernel of the map underlying f.*

In general, Definition 3 is stricter than Definition 5 but the two coincide for endofunctors preserving (enough) injections—*e.g.* any example from this paper.

Corollary 1. *For $\alpha\colon X \to TX$, the following are true:*
- *A bisimulation for α is a kernel bisimulation for α.*
- *If T preserves injections, a kernel bisimulation for α is a bisimulation for α.*

From Corollary 1 and Lemma 1 it follows that Definition 3 captures strong bisimulation for LTSs [16], for WLTSs [10], for Segala systems [18], for ULTraSs [15], and for FuTSs [12], since these are all instances of kernel bisimulations.

Lemma 6. *For $T = \coprod_{i\in I} T_i$ and $\alpha \in \mathsf{Sys}(T)$, $\mathrm{bis}(\alpha) = \bigcap_{i\in I} \mathrm{bis}(\pi_i \circ \alpha)$.*

A special but well known instance of Lemma 6 is given by definitions of bisimulations found in the literature for LTSs, WLTSs and in general FuTSs. In fact, all these bisimulation contain a universal quantification over the set of labels. For instance, a R is a bisimulation for an LTS $\alpha\colon X \to (\mathcal{P}X)^A$ iff:

$$x \; R \; x' \implies \forall a \in A \begin{pmatrix} (\forall y \in \phi(a)\, \exists z \in \psi(a)\, (y \; R \; z)) \wedge \\ (\forall z \in \psi(a)\, \exists y \in \phi(a)\, (y \; R \; z)) \end{pmatrix}$$

that is, iff R is the intersection of an A-indexed family composed by a bisimulation for each transition system $\alpha_a\colon X \to \mathcal{P}X$ projection of α on $a \in A$.

Lemma 7. *For $n \in \mathbb{N}$ and $\alpha \in \mathsf{Sys}(T^{n+1})$, there is $\underline{\alpha} \in \mathsf{Sys}(T)$ such that:*
- *$R \in \mathrm{bis}(\alpha) \implies \exists R' \in \mathrm{bis}(\underline{\alpha})(R = R'|_{\mathrm{car}(\alpha)})$,*
- *$R \in \mathrm{bis}(\underline{\alpha}) \implies R|_{\mathrm{car}(\alpha)} \in \mathrm{bis}(\alpha)$.*

Proof. Let $\underline{X}$ be $\coprod_{i=0}^{n} T^i X$ and $\underline{\alpha}\colon \underline{X} \to T(\underline{X})$ be $[T\iota_n\alpha_0, T\iota_0\alpha_1 \dots, T\iota_{n-2}\alpha_{n-1}]$ where $\iota_i\colon T^iX \to \underline{X}$ is the i-th coproduct injection, $\alpha_0\colon X \to T(T^nX)$ is α, and $\alpha_{i+1}\colon T^{i+1}X \to T(T^iX)$ is given by the identity for $T^{i+1}X$. If $R \in \mathrm{bis}(\overline{\alpha})$, then:

$$\begin{aligned} x \; R|_X \; x' &\implies x \; R \; x' \overset{(i)}{\implies} \underline{\alpha}(x) \; R^T \; \underline{\alpha}(x') \overset{(ii)}{\iff} \alpha(x) \; R^T|_{T^{n+1}X} \; \alpha(x') \\ &\overset{(iii)}{\iff} \alpha(x) \; (R|_X)^{T^{n+1}} \; \alpha(x') \end{aligned}$$

where (i) follows by $R \in \mathrm{bis}(\underline{\alpha})$; (ii) follows by noting that $\underline{\alpha}$ acts as α on X and hence both $\underline{\alpha}(x) = \alpha(x)$ and $\underline{\alpha}(y) = \alpha(y)$ are elements of $T^{n+1}X$; (iii) follows by inductively applying Lemmas 2 and 4. Therefore, $R|_X \in \mathrm{bis}(\alpha)$.

Assume $R \in \mathrm{bis}(\alpha)$ and define $\underline{R} = \coprod_{i=0}^{n} R^{T^i}$. By construction of $\underline{R}$, $\underline{x} \; \underline{R} \; \underline{x'}$ implies that $\underline{x}, \underline{x'} \in T^iX$ for some $i \in \{0, \dots, n\}$ meaning that the proof can be carried out by cases on each R^{T^i} composing $\underline{R}$. Assume $\underline{x}, \underline{x'} \in T^0X = X$, then:

$$\begin{aligned} \underline{x} \; \underline{R} \; \underline{x'} &\iff \underline{x} \; R \; \underline{x'} \overset{(i)}{\implies} \alpha(\underline{x}) \; R^{T^{n+1}} \; \alpha(\underline{x'}) \overset{(ii)}{\iff} \underline{\alpha}(\underline{x}) \; (R^{T^n})^T \; \underline{\alpha}(\underline{x'}) \\ &\iff \underline{\alpha}(\underline{x}) \; \underline{R}^T \; \underline{\alpha}(\underline{x'}) \end{aligned}$$

where (i) and (ii) follow by $R \in \mathrm{bis}(\alpha)$ and Lemma 2, respectively. Assume $\underline{x}, \underline{x'} \in T^{i+1}X$, we have that:

$$\begin{aligned} \underline{x}\ \underline{R}\ \underline{x'} &\iff \underline{x}\ R^{T^{i+1}}\ \underline{x'} \overset{(i)}{\iff} \underline{x}\ (R^{T^i})^T\ \underline{x'} \overset{(ii)}{\iff} \underline{\alpha}(\underline{x})\ (R^{T^i})^T\ \underline{\alpha}(\underline{x'}) \\ &\iff \underline{\alpha}(\underline{x})\ \underline{R}^T\ \underline{\alpha}(\underline{x'}) \end{aligned}$$

where (i) and (ii) follow by Lemma 2 and by definition of $\underline{\alpha}$ on $T^{i+1}X$. Therefore, $\underline{R} \in \mathrm{bis}(\overline{\alpha})$ and clearly $\underline{R}|_X = R$. □

Lemma 7 and its proof provide us with an encoding from systems whose steps are composed by multiple substeps to systems of substeps while preserving and reflecting their semantics in term of bisimulations. The trade-off of the encoding is a bigger statespace due to the explicit account of intermediate steps.

Lemma 8. *For $\mu\colon T \to S$ injective and $\alpha \in \mathsf{Sys}(T)$, $\mathrm{bis}(\alpha) = \mathrm{bis}(\mu_{\mathrm{car}(\alpha)} \circ \alpha)$.*

By applying the Lemma 8 to Example 1 we conclude that that bisimulations for ULTraSs coincide with bisimulations for WLTSs when these are seen as functional ULTraS as shown in [14, 15].

5 Reductions

In this section we formalize the intuition that a behaviour "shape" is (at least) as expressive as another whenever systems and homomorphisms of the latter can be "encoded" as systems and homomorphisms of the former, provided that their semantically relevant structures are preserved and reflected.

Definition 6. *For systems α and β, a* (system) reduction *$\sigma\colon \alpha \to \beta$ is given by a function $\sigma^c\colon \mathrm{car}(\alpha) \to \mathrm{car}(\beta)$ and a correspondence $\sigma^b \subseteq \mathrm{bis}(\alpha) \times \mathrm{bis}(\beta)$ s.t. σ^c carries a relation homomorphism for any pair of bisimulations in σ^b,* i.e.*:*

$$R\ \sigma^b\ R' \implies (x\ R\ x' \iff \sigma^c(x)\ R'\ \sigma^c(x')). \tag{1}$$

A system reduction $\sigma\colon \alpha \to \beta$ is called full *if $\sigma^c\colon \mathrm{car}(\alpha) \to \mathrm{car}(\beta)$ is surjective.*

For $\sigma\colon \alpha \to \beta$ a reduction, σ^c is always injective: the identity relation is always a bisimulation and hence condition (1) forces all x,x' such that $\sigma^c(x) = \sigma^c(x')$ to be equal in the beginning. Therefore the correspondence σ^b is always left-unique hence a surjection from $\mathrm{bis}(\beta)$ to $\mathrm{bis}(\alpha)$. This is indeed stronger than requiring preservation of bisimilarity since it entails that any bisimulation for α can be recovered by restricting some bisimulation for β to the image of $\mathrm{car}(\alpha)$ in $\mathrm{car}(\beta)$ through the map σ^c. Fullness implies σ^c and σ^b are isomorphism.

Remark 1. Condition (1) can be relaxed in two ways:
(a) $R\ \sigma^b\ R' \implies (x\ R\ x' \implies \sigma^c(x)\ R'\ \sigma^c(x'))$,
(b) $R\ \sigma^b\ R' \implies (x\ R\ x' \impliedby \sigma^c(x)\ R'\ \sigma^c(x'))$.

The condition (a) requires every bisimulation for α to be contained in some bisimulation for β whereas (b) requires every bisimulation for α to contain some bisimulation for β. Hence the two can be thought as *completeness* and *soundness* conditions for the reduction σ, respectively. □

System reductions can be extended to whole categories of systems provided they respect the structure of homomorphisms. Formally:

Definition 7. *For* C *and* D *categories of system, a* reduction σ *from* C *to* D*, written* $\sigma\colon \mathsf{C} \to \mathsf{D}$*, is a mapping that*

1. *assigns to any transition system* α *in* C *a system* $\sigma(\alpha)$ *in* D *and a system reduction* $\sigma_\alpha\colon \alpha \to \sigma(\alpha)$*;*
2. *assigns to any* $f\colon \alpha \to \beta$ *in* C *an homomorphism* $\sigma(f)\colon \sigma(\alpha) \to \sigma(\beta)$ *s.t.: (a)* $\sigma^c_\beta \circ f = \sigma(f) \circ \sigma^c_\alpha$*; (b)* $\sigma(id_\alpha) = id_{\sigma(\alpha)}$*; (c)* $\sigma(g \circ f) = \sigma(g) \circ \sigma(f)$*.*

A reduction $\sigma\colon \mathsf{C} \to \mathsf{D}$ *is called* full *if, and only if, every system reduction* σ_α *is full. A category* C *is said to* reduce *(resp.* fully *reduce) to* D*, if there is a reduction (resp. full reduction) from the* C *to* D*.*

Reductions can be easily composed at the level of their defining assignments. In particular, for reductions $\sigma\colon \mathsf{C} \to \mathsf{D}$ and $\tau\colon \mathsf{D} \to \mathsf{E}$, their composite reduction $\tau \circ \sigma\colon \mathsf{C} \to \mathsf{E}$ is a mapping that assigns to each system α the system $(\tau \circ \sigma)(\alpha)$ and the reduction given by $(\tau \circ \sigma)^c_\alpha \triangleq \tau^c_{\sigma(\alpha)} \circ \sigma^c_\alpha$ and $(\tau \circ \sigma)^b_\alpha \triangleq \tau^b_{\sigma(\alpha)} \circ \sigma^b_\alpha$; and to each $f\colon \alpha \to \alpha'$ the homorphism $(\tau \circ \sigma)(f)$. Reduction composition is associative and admits identities which are given on every C as the identity assignments for systems and homomorphisms. Any reduction restricts to a reduction from a subcategory of its domain and extends to a reduction to a super-category of its codomain. Moreover, fullness is preserved by the above operations.

For products, reductions can be given component-wise by suitable families of reductions that are "well-behaved" on homomorphisms. Formally:

Definition 8. *A family of reductions* $\{\sigma_i\colon \mathsf{C}_i \to \mathsf{D}_i\}_{i\in I}$ *is called* coherent *iff the following conditions hold for any* $i, j \in I$*:*

1. *if a function* f *extends to* $f_i \in \mathsf{C}_i$ *then there is* $f_j \in \mathsf{C}_j$ *s.t.* f *extends to* f_j*;*
2. $\sigma_i(f_i)$ *and* $\sigma_j(f_j)$ *share their underlying function whenever* f_i *and* f_j *do.*

Theorem 2. *A coherent family of (full) reductions* $\{\sigma_i\colon \mathsf{Sys}(T_i) \to \mathsf{Sys}(S_i)\}_{i\in I}$ *defines a (full) reduction* $\sigma\colon \mathsf{Sys}(\prod_{i\in I} T_i) \to \mathsf{Sys}(\prod_{i\in I} S_i)$*.*

Proof. Assume $\{\sigma_i\}_{i\in I}$ as above. For $\alpha \in \mathsf{Sys}(\prod T_i)$ let $\alpha_i = \pi_i \circ \alpha$ and define

$$\sigma(\alpha) \triangleq \langle \dots, \sigma_i(\alpha_i), \dots \rangle \qquad \sigma^c_\alpha \triangleq \sigma^c_{i,\alpha_i} \qquad \sigma^b_\alpha \triangleq \textstyle\bigcap_{i\in I} \sigma^b_{i,\alpha_i}$$

The assignment extends to all systems in $\mathsf{Sys}(\prod_{i\in I} T_i)$ and is well-defined by coherency and Lemma 6 since $R\,\sigma^b_\alpha\, R' \iff \forall i \in I (R\, \sigma^b_{i,\alpha_i}\, R')$ and for all $i \in I$, $\sigma^c_\alpha = \sigma^c_{i,\alpha_i}$. For any $i \in I$, $f\colon \alpha \to \beta$ defines an homomorphism $f_i\colon \alpha_i \to \beta_i$ in $\mathsf{Sys}(T_i)$ sharing its underlying function. Define $\sigma(f)$ as the homomorphism arising from the function underlying $\sigma_i(f_i)$. By coherency, the mapping is well-defined and satisfies all the necessary conditions since all σ_i are reductions. □

Correspondences for bisimulations presented in Lemmas 7 and 8 extend to reductions: injective transformations define full reductions and homogeneous systems reduce to systems for the base endofunctor, as formalised below.

Theorem 3. *For $\mu\colon T \to S$ an injective transformation, there is a full reduction $\hat{\mu}\colon \mathsf{Sys}(T) \to \mathsf{Sys}(S)$ given, on each α and each $f\colon \alpha \to \beta$ as $\hat{\mu}(\alpha) \triangleq \mu_{\mathrm{car}(\alpha)} \circ \alpha$, $\hat{\mu}^c_\alpha \triangleq id_{\mathrm{car}(\alpha)}$, $\hat{\mu}^b_\alpha \triangleq id_{\mathrm{bis}(\alpha)}$, and $\hat{\mu}(f) \triangleq f$.*

This theorem allows us to formalise the hierarchy shown in Section 1. For instance, the transformation described in Example 1 defines a full reduction from WLTSs to ULTraSs. Probabilistic systems are covered by the transformation induced by the inclusion $\mathcal{D}X \subseteq \mathcal{F}_{[0,\infty)}X$ whereas the remaining cases are trivial.

Theorem 4. *If T preserves injections then $\mathsf{Sys}(T^{n+1})$ reduces to $\mathsf{Sys}(T)$.*

Proof. Recall from Lemma 7 the construction of $\underline{\alpha}\colon \underline{X} \to T\underline{X}$ for any $\alpha\colon X \to T^{n+1}X$ and let $\iota_0\colon X \to \underline{X}$ denote the obvious injection. Define $\sigma\colon \mathsf{Sys}(T^{n+1}) \to \mathsf{Sys}(T)$ as the reduction given on each transition system α in $\mathsf{Sys}(T^{n+1})$ as

$$\sigma(\alpha) \triangleq \underline{\alpha} \quad \sigma^c_\alpha \triangleq \iota_0 \quad \sigma^b_\alpha \triangleq \{(R, \underline{R}) \mid R = \underline{R}|_X,\ R \in \mathrm{bis}(\alpha),\ \underline{R} \in \mathrm{bis}(\underline{\alpha})\}$$

and on each homomorphism $f\colon \alpha \to \beta$ in $\mathsf{Sys}(T^{n+1})$ as $\sigma(f) \triangleq \coprod_{i=0}^{n} T^i f$. By Lemma 7, σ^b_α is a correspondence and by construction σ respects homomorphism composition and identities. Thus, σ is a reduction from $\mathsf{Sys}(T^{n+1})$ to $\mathsf{Sys}(T)$. □

6 Application: reducing FuTSs to WLTSs

In this section we apply the theory presented in the previous sections to prove that (categories of) FuTSs reduce to (categories of) simple FuTSs, *i.e.* WLTSs. The reduction is given in stages reflecting the endofunctors structure.

Definition 9. *A monoid sequence $\vec{M}$ is called* homogeneous *if its elements are the same. FuTSs on $\vec{M}$ are called* homogeneous *if $\vec{M}$ is homogeneous.*

Lemma 9. *The FuTSs category fully reduces to that of homogeneous FuTSs.*

Proof. For a sequence of monoids $\vec{M} = \langle M_0, \dots, M_n\rangle$ let N denote the product monoid $\prod_{i=0}^{n} M_i$. Let 0_j denote the unit of M_j. For each $i \in \{0, \dots, n\}$, the assignment $x \mapsto \langle 0_0, \dots, 0_{i-1}, x, 0_{i+1}, \dots 0_n\rangle$ extends to an injective monoid homomorphism $m_i\colon M_i \to N$. The assignment $\phi \mapsto m_i \circ \phi$ defines an injective natural transformation $\mathcal{F}_{M_i} \to \mathcal{F}_N$ which extends to an injective transformation $\mathcal{F}_{\vec{M}} \to \mathcal{F}_{\vec{N}}$. We conclude by Theorems 2 and 3. □

Lemma 10. *The nested FuTSs category reduces to that of simple FuTSs.*

Proof. By Lemma 9 and Theorems 2 and 4. □

Lemma 11. *The FuTSs category reduces to that of combined FuTSs.*

Proof. By Theorem 2 and Lemma 10. □

Lemma 12. *The combined FuTSs category fully reduces to that of simple FuTSs.*

Proof. For a sequences $\vec{A} = \langle A_0, \dots, A_n \rangle$ and $\vec{M} = \langle M_0, \dots, M_n \rangle$ let B and N denote the cartesian product $\prod_{i=0}^{n} A_i$ and the product monoid $\prod_{i=0}^{n} M_i$, respectively. The mapping $\langle \phi_0, \dots, \phi_n \rangle \mapsto \lambda \langle a_0, \dots, a_n \rangle . \lambda x . \langle \phi_0(a_0)(x), \dots, \phi_n(a_n)(x) \rangle$ extends to an injective natural transformation from $(\mathcal{F}_{\vec{M}} -)^{\vec{A}} = \prod_{i=0}^{n} (\mathcal{F}_{M_i} -)^{A_i}$ to $(\mathcal{F}_N -)^B$. We conclude by Theorem 3. □

Theorem 5. *The FuTSs category reduces to that of simple FuTSs* i.e. *WLTSs.*

Proof. By Lemmas 11 and 12. □

For instance, consider an ULTraS $\alpha \colon X \to (\mathcal{P}_f \mathcal{F}_M X)^A$. By Lemma 9 it fully reduces to a homogeneous nested FuTS (X, α') for the sequences of labels and monoids $\langle A \rangle$ and $\langle \langle \mathbb{B} \times M, \mathbb{B} \times M \rangle \rangle$, respectively and such that:

$$\alpha'(x)(a)(\phi) \triangleq \begin{cases} \langle \mathtt{tt}, 0 \rangle & \text{given } \psi \in \alpha(x)(a) \text{ s.t. } \phi(y) = \langle \psi(y), 0 \rangle \text{ for all } y \in X \\ \langle \mathtt{ff}, 0 \rangle & \text{otherwise} \end{cases}$$

By Lemma 10, α' reduces to the WLTS $(X + \mathcal{F}_{\mathbb{B} \times M} X, \underline{\alpha'})$ with labels from A, weights from $\mathbb{B} \times M$, and such that:

$$\underline{\alpha'}(y)(a)(y') \triangleq \begin{cases} y(y') & \text{if } y \in \mathcal{F}_{\mathbb{B} \times M} X \text{ and } y' \in X \\ \alpha'(y)(a)(y') & \text{if } y \in X \text{ and } y' \in \mathcal{F}_{\mathbb{B} \times M} X \\ \langle \mathtt{ff}, 0 \rangle & \text{otherwise} \end{cases}$$

As exemplified by the above reduction for ULTraSs, FuTSs can be reduced to WLTSs by extending the original state space with weight functions and splitting steps accordingly. From this perspective, weight functions are *hidden states* in the original systems which the proposed reduction renders explicit. This observation highlights a trade-off between state and behaviour complexity of these semantically equivalent meta-models.

7 Conclusions

In this paper we have introduced a notion of *reduction* for categories of discrete state transition systems, and some general results for deriving reductions from the shape of computational aspects. As an application of this theory we have shown that FuTSs reduce to WLTSs, thus collapsing the upper part of the hierarchy in Section 1. Besides the classification interest, this result offers a solid bridge for porting existing and new results from WLTSs to FuTSs. For instance, SOS specifications formats presented in [10, 15] can cope now with FuTSs, and any abstract GSOS for these systems admits a specification in the format presented in [15]. Likewise, developing an HML style logic for bisimulation on WLTSs would readily yield a logic capturing bisimulation on FuTSs.

It remains an open question whether the hierarchy can be further collapsed, especially when other notion of reduction are considered. In fact, requiring a correspondence between bisimulations for the original and reduced systems may be too restrictive in some applications like bisimilarity-based verification techniques. This suggests to investigate laxer notions of reductions, such as those indicated in Remark 1. Another direction is to consider different behavioural equivalences, like trace equivalence or weak bisimulation. We remark that, as shown in [3, 4, 7], in order to deal with these and similar equivalences, endofunctors need to be endowed with a monad (sub)structure; although WLTSs are covered in [3, 13], an analogous account of FuTSs is still an open problem.

References

1. P. Aczel and N. Mendler. A final coalgebra theorem. In *Proc. CTCS*, volume 389 of *LNCS*, pages 357–365. Springer, 1989.
2. M. Bernardo, R. De Nicola, and M. Loreti. A uniform framework for modeling nondeterministic, probabilistic, stochastic, or mixed processes and their behavioral equivalences. *Inf. Comput.*, 225:29–82, 2013.
3. T. Brengos, M. Miculan, and M. Peressotti. Behavioural equivalences for coalgebras with unobservable moves. *JLAMP*, 84(6):826–852, 2015.
4. T. Brengos and M. Peressotti. A Uniform Framework for Timed Automata. In *Proc. CONCUR*, volume 59 of *LIPIcs*, pages 26:1–26:15, 2016.
5. R. De Nicola, D. Latella, M. Loreti, and M. Massink. A uniform definition of stochastic process calculi. *ACM Computing Surveys*, 46(1):5, 2013.
6. R. van Glabbeek, S. A. Smolka, and B. Steffen. Reactive, generative and stratified models of probabilistic processes. *Inf. Comput.*, 121:130–141, 1990.
7. I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace semantics via coinduction. *LMCS*, 3(4), 2007.
8. H. Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*, volume 2428 of *LNCS*. Springer, 2002.
9. J. Hillston. *A compositional approach to performance modelling*. Cambridge, 1996.
10. B. Klin and V. Sassone. Structural operational semantics for stochastic and weighted transition systems. *Inf. Comput.*, 227:58–83, 2013.
11. D. Latella, M. Massink, and E. de Vink. Bisimulation of labelled state-to-function transition systems coalgebraically. *LMCS*, 11(4), 2015.
12. D. Latella, M. Massink, and E. de Vink. A definition scheme for quantitative bisimulation. In *QAPL*, volume 194 of *EPTCS*, pages 63–78, 2015.
13. M. Miculan and M. Peressotti. Weak bisimulations for labelled transition systems weighted over semirings. *CoRR*, abs/1310.4106, 2013.
14. M. Miculan and M. Peressotti. GSOS for non-deterministic processes with quantitative aspects. In *Proc. QAPL*, volume 154 of *EPTCS*, pages 17–33, 2014.
15. M. Miculan and M. Peressotti. Structural operational semantics for non-deterministic processes with quantitative aspects. To appear in *TCS*, 2016.
16. R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
17. J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *TCS*, 249(1):3–80, 2000.
18. R. Segala and N. A. Lynch. Probabilistic simulations for probabilistic processes. *Nord. J. Comput.*, 2(2):250–273, 1995.
19. S. Staton. Relating coalgebraic notions of bisimulation. *LMCS*, 7(1), 2011.

A Omitted proofs

Proof of Lemma 1 For f injective, the assignments

$$\psi \mapsto f \circ \psi \qquad Z \mapsto \{f(z) \mid z \in Z\} \qquad \phi \mapsto \lambda y. \textstyle\sum_{x: f(x)=y} \phi(x)$$

describe injective functions. □

Proof of Lemma 2 Let $\kappa^S \colon SX \to SX/R^S$ be the canonical projection to the quotient induced by the equivalence relation R^S. Since, by definition, $\kappa^S(\rho) = \kappa^S(\theta)$ implies $(S\kappa)(\rho) = (S\kappa)(\theta)$ there is a (unique) function $q^S \colon SX/R^S \to S(X/R)$ such that $S\kappa = q^S \circ \kappa^S$. From $TS\kappa = Tq^S \circ T\kappa^S$ and the definition of R^{TS} and $\left(R^S\right)^T$, it follows that: $\phi \left(R^S\right)^T \psi \implies (TS\kappa)(\phi) = (TS\kappa)(\psi) \implies \phi\ R^{TS}\ \psi$ proving first part of the thesis. Since $\rho\ R^S\ \theta \iff \kappa^S(\rho) = \kappa^S(\theta)$ we conclude that q^S is an injection and, by hypothesis, Tq^S is an injection too. Therefore:

$$\begin{aligned} \phi\ R^{TS}\ \psi &\implies (TS\kappa)(\phi) = (TS\kappa)(\psi) \implies (T\kappa^S)(\phi) = (T\kappa^S)(\psi) \\ &\implies \phi \left(R^S\right)^T \psi \end{aligned}$$

completing the proof. □

Proof of Lemma 3 Write T for $\prod_{i\in I} T_i$ and recall that $\left(\prod_{i\in I} T_i\right) X$ is $\prod_{i\in I} T_i X$. Then:

$$\begin{aligned} \phi\ R^T\ \psi &\iff (\textstyle\prod T_i\kappa)(\phi) = (\prod T_i\kappa)(\psi) \iff \prod (T_i\kappa)(\phi_i) = \prod (T_i\kappa)(\psi_i) \\ &\iff \phi \textstyle\prod R^{T_i}\ \psi \end{aligned}$$

where $\kappa \colon X \to X/R$ is the canonical projection to the quotient induced by R and $\pi_i \colon \prod_{i\in I} T_i X \to T_i X$ is the i-th projection. □

Proof of Lemma 4 Let $\kappa \colon Y \to Y/R$ and $\kappa' \colon X \to X/R|_X$ be the canonical projections induced by R and $R|_X$, respectively. Since the latter is given by restriction of the former to $X \subseteq Y$, there is a unique and injective map $q \colon X/R|_X \to Y/R$ such that $\kappa = q \circ \kappa$. The first part of the thesis follows by:

$$\begin{aligned} \phi\ (R|_X)^T\ \psi &\implies (T\kappa')(\phi) = (T\kappa')(\psi) \implies (T\kappa)(\phi) = (T\kappa)(\psi) \\ &\implies \phi\ R^T\ \psi \quad \text{since } T\kappa = Tq \circ T\kappa'. \end{aligned}$$

On the other hand, by hypothesis on T, Tq is injective and hence

$$\begin{aligned} \phi\ R^T|_{TX}\ \psi &\implies (T\kappa)(\phi) = (T\kappa)(\psi) \implies (T\kappa')(\phi) = (T\kappa')(\psi) \\ &\implies \phi\ (R|_X)^T\ \psi \end{aligned}$$ □

Proof of Lemma 5 It holds that

$$\begin{aligned}\phi\ R^T\ \psi &\iff T\kappa(\phi) = T\kappa(\psi) \stackrel{(i)}{\iff} (\mu_X \circ T\kappa)(\phi) = (\mu_X \circ T\kappa)(\psi)\\ &\stackrel{(ii)}{\iff} (S\kappa \circ \mu_X)(\phi) = (S\kappa \circ \mu_X)(\psi) \iff \mu_X(\phi)\ R^S\ \mu_X(\psi)\end{aligned}$$

where (i) and (ii) follow by μ_X being injective and by μ being a natural transformation, respectively. □

Proof of Theorem 1 Assume R is a bisimulation for $\alpha\colon X \to TX$. For $f, f'\colon X \to Y$ s.t. $x\ R\ x' \implies f(x) = f'(x')$ we have that:

$$\begin{aligned}x\ R\ x &\implies \alpha(x)\ R^T\ \alpha(x') \iff (T\kappa \circ \alpha)(x) = (T\kappa \circ \alpha)(x')\\ &\stackrel{(i)}{\implies} (Tf \circ \alpha)(x) = (Tf' \circ \alpha)(x')\end{aligned}$$

where (i) follows by noting that, since $\kappa\colon X \to X/R$ is a canonical projection and $x\ R\ x' \implies f(x) = f'(x')$, there is (a unique) $q\colon X/R \to Y$ s.t. $f = q \circ \kappa = f'$.

Assume R is a precongruence for α, we have that:

$$\begin{aligned}x\ R\ x &\stackrel{(i)}{\iff} \kappa(x) = \kappa(x') \stackrel{(ii)}{\implies} (T\kappa \circ \alpha)(x) = (T\kappa \circ \alpha)(x')\\ &\stackrel{(iii)}{\iff} \alpha(x)\ R^T\ \alpha(x')\end{aligned}$$

where (i) follows by definition of $\kappa\colon X \to X/R$, (ii) by R being a precongruence, and (iii) by definition of R^T. □

Proof of Corollary 1 By Theorem 1 and [19, Thm. 4.1]. □

Proof of Lemma 6 By Theorem 2, $\alpha(x)\ R^T\ \alpha(x') \iff \alpha(x)\ \left(\prod_{i \in I} R^{T_i}\right)\ \alpha(x')$ and hence $\alpha(x)\ R^T\ \alpha(x') \iff \forall i \in I\ (\pi_i\alpha)(x)\ R_i^T\ (\pi_i\alpha)(x')$. □

Proof of Lemma 8 For a relation R, $R \in \mathrm{bis}(\alpha)$ iff $x\ R\ y \implies \alpha(x)\ R^T\ \alpha(y)$ and $R \in \mathrm{bis}(\mu_X \circ \alpha)$ iff $x\ R\ y \implies (\mu_X \circ \alpha)(x)\ R^S\ (\mu_X \circ \alpha)(y)$. We conclude by Lemma 5. □

Proof of Theorem 3 By Lemma 8. □