

Characteristic Formulae for Fixed-Point Semantics: A General Framework*

Luca Aceto

School of Computer Science, Reykjavik University,
Menntavegur 1, Nauthólsvík, IS-101 Reykjavík, Iceland
luca@ru.is

Anna Ingólfssdóttir

School of Computer Science, Reykjavik University,
Menntavegur 1, Nauthólsvík, IS-101 Reykjavík, Iceland
annai@ru.is

Paul Blain Levy

University of Birmingham,
Birmingham B15 2TT, UK
pbl@cs.bham.ac.uk

Joshua Sack

School of Computer Science, Reykjavik University,
Menntavegur 1, Nauthólsvík, IS-101 Reykjavík, Iceland
joshua.sack@gmail.com

The literature on concurrency theory offers a wealth of examples of characteristic-formula constructions for various behavioural relations over finite labelled transition systems and Kripke structures that are defined in terms of fixed points of suitable functions. Such constructions and their proofs of correctness have been developed independently, but have a common underlying structure. This study provides a general view of characteristic formulae that are expressed in terms of logics with a facility for the recursive definition of formulae. It is shown how several examples of characteristic-formula constructions from the literature can be recovered as instances of the proposed general framework, and how the framework can be used to yield novel constructions. The paper also offers general results pertaining to the definition of co-characteristic formulae and of characteristic formulae expressed in terms of infinitary modal logics.

1 Introduction

Various types of automata are fundamental formalisms for the description of the behaviour of computing systems. For instance, a widely used model of computation is that of *labelled transition systems* (LTSs) [29]. LTSs underlie Plotkin's Structural Operational Semantics [47] and, following Milner's pioneering work on CCS [41], are by now the formalism of choice for describing the semantics of various process description languages.

Since automata like LTSs can be used for describing specifications of process behaviours as well as their implementations, an important ingredient in their theory is a notion of behavioural equivalence or preorder between (states of) LTSs. A behavioural equivalence describes formally when (states of) LTSs afford the same 'observations', in some appropriate technical sense. On the other hand, a behavioural preorder is a possible formal embodiment of the idea that (a state in) an LTS affords at least as many 'observations' as another one. Taking the classic point of view that an implementation correctly implements a specification when each of its observations is allowed by the specification, behavioural preorders may therefore be used to establish the correctness of implementations with respect to their specifications, and to support the stepwise refinement of specifications into implementations.

*The work of Luca Aceto, Anna Ingólfssdóttir and Joshua Sack has been partially supported by the projects 'New Developments in Operational Semantics' (nr. 080039021) and 'Processes and Modal Logics' of the Icelandic Research Fund. Joshua Sack has been further supported by a grant from Reykjavik University's Development Fund. Paul Blain Levy has been supported by the ESPRC Advanced Research Fellowship EP/E056091/1.

The lack of consensus on what constitutes an appropriate notion of observable behaviour for reactive systems has led to a large number of proposals for behavioural equivalences and preorders for concurrent processes. In his by now classic paper [22], van Glabbeek presented a taxonomy of extant behavioural preorders and equivalences for processes.

The approach to the specification and verification of reactive systems in which automata like LTSs are used to describe both implementations and specifications of reactive systems is often referred to as *implementation verification* or *equivalence checking*.

An alternative approach to the specification and verification of reactive systems is that of *model checking* [11, 12, 49]. In this approach, automata are still the formalism of choice for the description of the actual behaviour of a concurrent system. However, specifications of the expected behaviour of a system are now expressed using a suitable logic, for instance, a modal or temporal logic [19, 48]. Verifying whether a concurrent process conforms to its specification expressed as a formula in the logic amounts to checking whether the automaton describing the behaviour of the process is a model of the formula.

It is natural to wonder what the connection between these two approaches to the specification and verification of concurrent computation is. A classic, and most satisfying, result in the theory of concurrency is the characterization theorem of bisimulation equivalence [41, 45] in terms of Hennessy-Milner logic (HML) due to Hennessy and Milner [27]. This theorem states that two bisimilar processes satisfy the same formulae in Hennessy-Milner logic, and if the processes satisfy a technical finiteness condition, then they are also bisimilar when they satisfy the same formulae in the logic. This means that, for bisimilarity and HML, the process equivalence induced by the logic coincides with behavioural equivalence, and that, whenever two processes are *not* equivalent, we can always find a formula in HML that witnesses a reason why they are not. This distinguishing formula is useful for debugging purposes, and can be algorithmically constructed for finite processes—see, e.g., [30, 39].

The characterization theorem of Hennessy and Milner is, however, less useful if we are interested in using it directly to establish when two processes are behaviourally equivalent using model checking. Indeed, that theorem seems to indicate that to show that two processes are equivalent we need to check that they satisfy the same formulae expressible in the logic, and there are infinitely many such formulae, even modulo logical equivalence. Is it possible to find a *single* formula that characterizes the bisimulation equivalence class of a process p —in the sense that any process is bisimilar to p if, and only if, it affords that property? Such a formula, if it exists, is called a *characteristic formula*. When a characteristic formula for a process modulo a given notion of behavioural equivalence or preorder can be algorithmically constructed, implementation verification can be reduced to model checking, and we can translate automata to logic. (An investigation of the model checking problems that can be reduced to implementation verification may, for instance, be found in the paper [8].)

Characteristic formulae provide a very elegant connection between automata and logic, and between implementation verification and model checking. But, can they be constructed for natural, and suitably expressive, automata-based models and known logics of computation? To the best of our knowledge, this natural question was first addressed in the literature on concurrency theory in the paper [25]. In that study, Graf and Sifakis offered a translation from recursion-free terms of Milner’s CCS [41] into formulae of a modal language representing their equivalence class with respect to observational congruence.

Can one characterize the equivalence class of an arbitrary finite process—for instance one described in the regular fragment of CCS—up to bisimilarity using HML? The answer is negative because each formula in that logic can only describe a finite fragment of the initial behaviour of a process—see, for instance, [2] for a textbook presentation. However, as shown in, e.g., [28, 51], adding a facility for the recursive definition of formulae to (variants of) HML yields a logic that is powerful enough to

support the construction of characteristic formulae for various types of finite processes modulo notions of behavioural equivalence or preorder.

Following on the work presented in those original references, the literature on concurrency theory offers by now a wealth of examples of characteristic-formula constructions for various behavioural relations, over finite labelled transition systems, Kripke structures and timed automata, that are defined in terms of fixed points of suitable functions. (See, for instance, the references [3, 9, 13, 21, 33, 35, 44].) Such constructions and their proofs of correctness have been developed independently, but have a common underlying structure. It is therefore natural to ask oneself whether one can provide a general framework within which some of the aforementioned results can be recovered from general principles that isolate the common properties that lie at the heart of all the specific constructions presented in the literature. Not only do such general principles allow us to recover extant constructions in a principled fashion, but they may also yield novel characteristic-formula constructions ‘for free’.

In this study, we offer a general view of characteristic formulae that are expressed in terms of logics with a facility for the recursive definition of formulae. The proposed framework applies to behavioural relations that are defined as greatest or least fixed points of suitable monotone endofunctions over complete lattices. Examples of such relations are those belonging to the family of bisimulation- and simulation-based semantics, which can all be defined as greatest fixed points of monotone endofunctions over the complete lattice of binary relations over the set of states of a labelled transition system.

We show that if, in a suitable technical sense defined in Section 2.3, a collection of recursively defined logical formulae expresses the endofunction \mathcal{F} underlying the definition of a behavioural relation, then the greatest interpretation of that collection of formulae characterizes the behavioural relation that is the greatest fixed point of \mathcal{F} . (See Theorem 2.16.) Using this result, we are able to recover, as instances of the proposed general framework and essentially for free, several examples of characteristic-formula constructions from the literature. In particular, we focus on simulation [45], bisimulation [41, 45], ready simulation [6, 36], and prebisimulation semantics [1, 40]. In addition, we show that the framework can be used to yield novel, albeit at times somewhat expected, constructions. By way of example, we provide characteristic formulae for back and forth bisimilarity, back and forth bisimilarity with indistinguishable states [17], the boolean precongruences introduced in [38], conformance simulation semantics [20] and extended simulation semantics [55].

The techniques developed in Section 2.3 do not readily yield characteristic-formula constructions for the nested simulation semantics from [26] and for simulation equivalence (or mutual simulation). In Section 4 of the paper, however, we extend our approach, providing the theoretical tools needed to offer characteristic formulae for them.

All the behavioural relations mentioned above are obtained as (intersections of) greatest fixed points of suitable monotone endofunctions. In the main body of the paper, we also show how to define, in a principled fashion, *co-characteristic formulae* for such behavioural relations. (See Section 5.) The intuition behind the notion of co-characteristic formula is best understood when focusing on behavioural equivalences. In this setting, a co-characteristic formula for a process p expresses the property that any process that is *inequivalent* to p should satisfy. Least-fixed-point interpretations are the appropriate ones for the definition of co-characteristic formulae since, to show that two processes are *not* equivalent, we need to find some ‘finite observation’ that only one of them affords. As we show in Section 5, each of the results about characteristic formulae from Section 2.3 has a dual version that applies to co-characteristic formulae.

We trust that the general view of characteristic-formula constructions we provide in this article will offer a framework for the derivation of many more such results and for explaining the reasons underlying the success of extant constructions of this kind in the literature.

Further related work In this paper, we mostly focus on characteristic-formula constructions that are given in terms of logics with a facility for the recursive definition of formulae. The literature on concurrency theory and modal logics, however, also offers characteristic formulae for variations on bisimilarity that employ branching-time temporal logics or infinitary modal logics.

A classic, early result on characteristic formulae was obtained in the paper [9]. That paper shows that each finite Kripke structure can be characterized by a formula in Computation Tree Logic (CTL) [10] up to the natural variation on bisimilarity over Kripke structures. Another characteristic formula result is presented in that paper for an equivalence between states that takes ‘stuttering’ into account. (This equivalence is closely related to van Glabbeek’s and Weijland’s branching bisimilarity [23], for which logical characterizations have been offered by De Nicola and Vaandrager in the paper [16].) Browne, Clarke and Grumberg show that equivalence classes of states in a finite Kripke structure modulo stuttering equivalence are completely characterized by next-time-free CTL formulae. (The absence of the next-time operator is expected in light of the inability of stuttering equivalence to ‘count’ the number of steps in a stuttering sequence.) Kučera and Schnoebelen have presented a refinement of the above classic theorem by Browne, Clarke and Grumberg in the paper [32].

Characteristic formulae for bisimilarity, expressed in terms of infinitary modal logic, are given in [5, Theorem 3.2] over Kripke structures and in [4, Theorem 11.12] over non-wellfounded sets. Such characteristic formulae rely on the definition of the approximants of bisimilarity and on the non-trivial fact that the ‘branching degree’ of the process for which one is constructing the formula determines the approximant one needs to consider and the cardinality of the infinitary conjunctions and disjunctions in the formula. In the case of image-finite LTSs, this was shown by van Glabbeek, who proved in [24] that if a process p is image finite and p and q are related by the ω -approximant of bisimilarity, then p and q are indeed bisimilar. (This result is a sharpening of a classic theorem by Hennessy and Milner, who proved it under the assumption that both p and q are image finite.) The proof of Theorem 11.12 in [4] relies on an extension of the aforementioned result of van Glabbeek’s to infinite regular cardinals other than ω , which Barwise and Moss attribute to Baltag—see [4, Lemma 11.13]. In Section 6 of this study, we present a general account of the above-mentioned characteristic formula constructions in terms of infinitary modal logics, as well as of some novel ones.

The theoretical significance of characteristic formulae in infinitary modal logic is exemplified by the developments in [43], where Moss uses them to obtain new and elegant weak completeness and decidability proofs for standard systems of modal logic.

Last, but by no means least, we would be amiss not to mention the use of characteristic formulae in classic first-order logic. In that setting, they appear under the name of ‘Hintikka formulae’ and play a role in relating the notion of m -isomorphism with the equivalence over structures induced by formulae whose quantifier depth is at most m —see, e.g., [18, 54] for overviews.

Roadmap of the paper The paper is organized as follows. In Section 2 we describe the theoretical background, as well as the main technical results, the paper relies on. In particular, in that section we offer our main technical result (Theorem 2.16), which is at the heart of our general approach to the construction of characteristic formulae for behavioural semantics defined as greatest fixed points of monotone endofunctions. Section 3 is devoted to applications of our main theorem and to developments related to the above-mentioned notion of co-characteristic formula. As remarked earlier, the techniques developed in Section 2.3 do not readily yield characteristic-formula constructions for behavioural relations such as the nested simulation preorders and simulation equivalence. In Section 4, we provide the theoretical tools needed to offer characteristic formulae for those relations. In Section 6, we present a

general account of characteristic formula constructions in terms of infinitary modal logics. Finally, in Section 7 we give some concluding remarks and present directions for future research.

2 Some theoretical background

In this section we provide the theoretical background needed in the paper.

2.1 Posets, homomorphisms and fixed points

Definition 2.1

1. A partially ordered set, or poset, (A, \sqsubseteq_A) consists of a set A and a partial order \sqsubseteq_A over it. Often this poset is denoted simply by the set A and we write \sqsubseteq instead of \sqsubseteq_A if the meaning is clear from the context.
2. For posets A and B , a function $\phi : A \longrightarrow B$ is
 - monotone if it preserves the order over A , i. e. if $x \sqsubseteq_A y$ implies $\phi(x) \sqsubseteq_B \phi(y)$ for all $x, y \in A$,
 - an isomorphism if it is bijective and both ϕ and its inverse ϕ^{-1} are monotone.
3. If f is a monotone endofunction on a partially ordered set A , that is, a function from A to itself, then $x \in A$ is
 - a pre-fixed point of f when $f(x) \sqsubseteq x$,
 - a post-fixed point of f when $x \sqsubseteq f(x)$, and
 - a fixed point of f when $f(x) = x$, i.e. when x is both a pre-fixed point and a post-fixed point of f .

We write νf for the greatest post-fixed point and μf for the least pre-fixed point of f , if they exist.

Note that the greatest or least element of A satisfying any given property is unique if it exists.

The following result is well known.

Lemma 2.2 *Let f be a monotone endofunction on a poset A .*

1. *If νf exists, then it is the greatest fixed point of f .*
2. *If μf exists, then it is the least fixed point of f .*

Proof 1. Monotonicity of f gives $f(\nu f) \sqsubseteq f(f(\nu f))$, so $f(\nu f)$ is a post-fixed point of f , giving $f(\nu f) \sqsubseteq \nu f$. So νf is pre-fixed point of f , hence a fixed point. If $a \in A$ is a fixed point of f , it is a post-fixed point, so $a \sqsubseteq \nu f$.

2. Dual argument. □

The theorem below, albeit simple, is the key to the general theory we present in this paper.

Theorem 2.3 *Let A and B be posets, f and g be monotone endofunctions on A and B respectively and $\phi : A \rightarrow B$ an isomorphism such that $\phi \circ f = g \circ \phi$, or equivalently such that the square*

$$\begin{array}{ccc} A & \xrightarrow{f} & A \\ \phi \downarrow & & \downarrow \phi \\ B & \xrightarrow{g} & B \end{array}$$

commutes. Then the following statements hold:

1. Let $x \in A$. Then x is a post-fixed point (resp. pre-fixed point, fixed point) of f iff $\phi(x)$ is a post-fixed point (resp. pre-fixed point, fixed point) of g .
2. νf exists iff νg exists and we then have $\phi(\nu f) = \nu g$.
3. μf exists iff μg exists and we then have $\phi(\mu f) = \mu g$.

Proof 1. This statement follows immediately from the assumptions.

2. By part (1), $\phi(\nu f)$ is a post-fixed point for g . Towards showing that it is the greatest post-fixed point, assume that y is a post-fixed point of g . Then $\phi^{-1}(y)$ is a post-fixed point of f by part (1), so $\phi^{-1}(y) \sqsubseteq \nu f$. Applying ϕ gives $y \sqsubseteq \phi(\nu f)$ as required.
3. Similar to the proof above. □

Definition 2.4 A poset A is a complete lattice when every $U \subseteq A$ has a least upper bound (lub), written $\bigsqcup U$.

Note that the lub of A is the greatest element of A and the lub of \emptyset is the least element of A . Furthermore, if each subset of a poset has a lub then each subset U also has a greatest lower bound, written $\bigsqcap U$, given by the lub of the set of all of its lower bounds.

It is well known that, for each set A , the collection $\mathcal{P}(A)$ of all subsets of A ordered by inclusion is a complete lattice.

The following theorem is due to Tarski [53].

Theorem 2.5 If A is a complete lattice and f is a monotone endofunction on A , then

- both νf and μf exist,
- $\nu f = \bigsqcup \{a \mid a \sqsubseteq f(a)\}$ and
- $\mu f = \bigsqcap \{a \mid f(a) \sqsubseteq a\}$.

Moreover, νf is the greatest post-fixed point of f and μf is its least pre-fixed point.

For two sets A and I , as usual, we let A^I denote the set of all functions from I to A . To reduce the need to name a function, in what follows, we may often denote a function by using the notation \mapsto ; for example, $i \mapsto (a, i)$ is the function that maps every element i of its domain into a pair, whose first coordinate is a and second is i .

Lemma 2.6 If A is a poset then A^I is a poset under the pointwise ordering given by $\sigma_1 \sqsubseteq \sigma_2$ iff $\sigma_1(i) \sqsubseteq_A \sigma_2(i)$ for all $i \in I$. In particular, A^I is a complete lattice if A is a complete lattice.

The following lemma on greatest fixed points will be applied in Sections 3.1.1 and 5.2 below.

Lemma 2.7 Let A be a set and let \mathcal{F} be a monotone endofunction on the complete lattice $\mathcal{P}(A \times A)$. Let $\tilde{\mathcal{F}} : S \mapsto (\mathcal{F}(S^{-1}))^{-1}$. Then $\tilde{\mathcal{F}}$ is monotone and we have

$$\begin{aligned} (\nu \mathcal{F})^{-1} &= \nu \tilde{\mathcal{F}} \\ (\mu \mathcal{F})^{-1} &= \mu \tilde{\mathcal{F}}. \end{aligned}$$

Proof $\tilde{\mathcal{F}}$ is monotone because it is a composition of monotone functions. The equations follow from Theorem 2.3(2)–(3), since the following square commutes.

$$\begin{array}{ccc} \mathcal{P}(\mathbf{P} \times \mathbf{P}) & \xrightarrow{\mathcal{F}} & \mathcal{P}(\mathbf{P} \times \mathbf{P}) \\ (-)^{-1} \downarrow & & \downarrow (-)^{-1} \\ \mathcal{P}(\mathbf{P} \times \mathbf{P}) & \xrightarrow{\tilde{\mathcal{F}}} & \mathcal{P}(\mathbf{P} \times \mathbf{P}) \end{array}$$

□

2.2 Hennessy-Milner logic with variables and declarations

We recall the standard Hennessy-Milner logic (HML) extended with variables (see, for instance, [34]). The logic depends on a finite set \mathbf{A} , whose elements will be viewed as actions, and an I -indexed set $X = \{X_i \mid i \in I\}$ of variables, where I is a finite set. We write $\mathcal{L}(I, \mathbf{A})$ for the set of formulae given by the following grammar

$$\begin{aligned}
 F & ::= X_i && (i \in I) \\
 & \mid \bigwedge_{k \in K} F_k && (K \text{ a finite set}) \\
 & \mid \bigvee_{k \in K} F_k && (K \text{ a finite set}) \\
 & \mid \langle a \rangle F && (a \in \mathbf{A}) \\
 & \mid [a] F && (a \in \mathbf{A}).
 \end{aligned}$$

If the meaning is clear from the context, we often omit I , \mathbf{A} or both. As usual, we write $\#$ for nullary conjunction, \wedge for binary conjunction, ff for nullary disjunction and \vee for binary disjunction.

The structures on which we interpret a logic $\mathcal{L}(I, \mathbf{A})$ are labelled transition systems.

Definition 2.8 An \mathbf{A} -labelled transition system (LTS) is a pair $P = (\mathbf{P}, \longrightarrow)$ consisting of

- a finite set \mathbf{P} , and
- a transition relation $\longrightarrow \subseteq \mathbf{P} \times \mathbf{A} \times \mathbf{P}$.

Typically, \mathbf{A} is clear from context, and we thus often refer to an \mathbf{A} -labelled transition system as simply a labelled transition system. As usual, we write $p \xrightarrow{a} p'$ for $(p, a, p') \in \longrightarrow$. We often think of \mathbf{P} as a set of processes, \mathbf{A} as a set of actions, and $p \xrightarrow{a} p'$ as a transition from process p to process p' via action a . We write $p \xrightarrow{a}$ if there exists a p' such that $p \xrightarrow{a} p'$, and we write $p \not\xrightarrow{a}$ if there is no such p' . Although we restrict the sets I , \mathbf{A} , and \mathbf{P} to be finite, all the results of Sections 2–5 extend easily to the case where infinite sets are used by allowing infinite conjunctions and disjunctions over sets of arbitrary cardinality, and by allowing the sets \mathbf{A} and I to be arbitrary as well. In Section 6 of the paper, we will explicitly consider the version of the logic $\mathcal{L}(\emptyset, \mathbf{A})$ with conjunctions and disjunctions over arbitrary cardinals and over a countable set \mathbf{A} .

Fixing \mathbf{A} , a formula is interpreted over an \mathbf{A} -labelled transition system $(\mathbf{P}, \longrightarrow)$ as the set of elements from \mathbf{P} that satisfy the formula, where satisfaction is generally determined by either a relation or a function. We will define satisfaction both ways, as the relation involves a notation that may be more readable when using longer formulae, and the function involves a notation that may clarify certain relationships among the semantics of formulae and other concepts. As a formula typically contains variables, it has to be interpreted with respect to a variable *interpretation* $\sigma \in \mathcal{P}(\mathbf{P})^I$ (possibly with a subscript) that associates to each $i \in I$ the set of processes in \mathbf{P} that are assumed to satisfy the variable X_i . We first interpret a formula $F \in \mathcal{L}(I)$ by means of a satisfaction relation \models from $\mathcal{P}(\mathbf{P})^I \times \mathbf{P}$ to $\mathcal{L}(I)$. This relation tells us when a process p satisfies the formula F under the interpretation σ and is defined by structural recursion on F as follows.

$$\begin{aligned}
 \sigma, p \models X_i & \Leftrightarrow p \in \sigma(i), \\
 \sigma, p \models \bigwedge_{k \in K} F_k & \Leftrightarrow \sigma, p \models F_k \text{ for all } k \in K, \\
 \sigma, p \models \bigvee_{k \in K} F_k & \Leftrightarrow \sigma, p \models F_k \text{ for some } k \in K, \\
 \sigma, p \models \langle a \rangle F & \Leftrightarrow \text{there is some } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F, \\
 \sigma, p \models [a] F & \Leftrightarrow \text{for all } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F.
 \end{aligned}$$

Now, for each $F \in \mathcal{L}(I)$, we define $\llbracket F \rrbracket : \mathcal{P}(\mathbf{P})^I \rightarrow \mathcal{P}(\mathbf{P})$ by

$$\llbracket F \rrbracket \sigma = \{p \in \mathbf{P} \mid \sigma, p \models F\}.$$

In particular $\llbracket X_i \rrbracket \sigma = \sigma(i)$, $\llbracket ff \rrbracket \sigma = \emptyset$ and $\llbracket \# \rrbracket \sigma = \mathbf{P}$ for all $\sigma \in \mathcal{P}(\mathbf{P})^I$. Moreover, $\llbracket \bigwedge_{k \in K} F_k \rrbracket \sigma = \bigcap_{k \in K} (\llbracket F_k \rrbracket \sigma)$ and $\llbracket \bigvee_{k \in K} F_k \rrbracket \sigma = \bigcup_{k \in K} (\llbracket F_k \rrbracket \sigma)$.

Lemma 2.9 1. For any $F \in \mathcal{L}(I)$, the function $\llbracket F \rrbracket : \mathcal{P}(\mathbf{P})^I \rightarrow \mathcal{P}(\mathbf{P})$ is monotone.

2. $\mathcal{P}(\mathbf{P})^I$ is a complete lattice.

Proof The first claim can be proved easily by structural induction on $F \in \mathcal{L}(I)$. The second one follows from Lemma 2.6 and the fact that $\mathcal{P}(\mathbf{P})$ is a complete lattice under inclusion. \square

As mentioned above, the formulae of $\mathcal{L}(I)$ (as well as most variations of this language that we will consider) include variables indexed by I and can therefore only be interpreted with respect to a given variable interpretation. Unlike in the classical modal μ -calculus [31], where variables are typically bound by fixed point operators in the language that will serve to determine the appropriate variable interpretation, we instead adopt the approach followed in the so-called equational μ -calculus and involve systems of equations that help induce a unique variable interpretation. Following Larsen [34], we define systems of equations implicitly using declarations.

Definition 2.10 • Given index sets I, J and given language $\mathcal{L}(J)$, we call a function $D : I \rightarrow \mathcal{L}(J)$ (or equivalently $D \in \mathcal{L}(J)^I$) an I -indexed declaration for a language $\mathcal{L}(J)$.

- We call a function $E : I \rightarrow \mathcal{L}(I)$ (or equivalently $E \in \mathcal{L}(I)^I$) an I -indexed endodeclaration or an endodeclaration for $\mathcal{L}(I)$.

Semantically a declaration $D : I \rightarrow \mathcal{L}(J)$ induces a function $\llbracket D \rrbracket : \mathcal{P}(\mathbf{P})^J \rightarrow \mathcal{P}(\mathbf{P})^I$ as follows.

Definition 2.11 If $D : I \rightarrow \mathcal{L}(J)$ is a declaration then we define $\llbracket D \rrbracket : \mathcal{P}(\mathbf{P})^J \rightarrow \mathcal{P}(\mathbf{P})^I$ by

$$\forall i \in I : (\llbracket D \rrbracket \sigma)(i) = \llbracket D(i) \rrbracket \sigma.$$

If E is an endodeclaration, then $\llbracket E \rrbracket$ is an endofunction on the complete lattice $\mathcal{P}(\mathbf{P})^I$. This is significant, since by the next lemma (Lemma 2.12), we will see that $\llbracket E \rrbracket$ is monotone, and hence has a greatest and a least fixed point which we refer to as the *greatest and least interpretation of E* respectively. This will be the method for inducing a unique variable interpretation from a system of equations that we use throughout the paper.

Lemma 2.12 If E is an endodeclaration then $\llbracket E \rrbracket$ is a monotone endofunction on $\mathcal{P}(\mathbf{P})^I$ and therefore $\nu \llbracket E \rrbracket$ and $\mu \llbracket E \rrbracket$ exist.

Proof Assume that $\sigma_1, \sigma_2 \in \mathcal{P}(\mathbf{P})^I$ where $\sigma_1 \subseteq \sigma_2$. We have to prove that

$$\llbracket E \rrbracket \sigma_1 \subseteq \llbracket E \rrbracket \sigma_2,$$

or equivalently that

$$(\llbracket E \rrbracket \sigma_1)(i) \subseteq (\llbracket E \rrbracket \sigma_2)(i) \text{ for all } i \in I.$$

As, by definition, $(\llbracket E \rrbracket \sigma_j)(i) = \llbracket E(i) \rrbracket \sigma_j$ for $j = 1, 2$ the claim follows from Lemma 2.9(1). \square

We recall that the greatest and the least fixed points of the endofunction $\llbracket E \rrbracket$ induced by an endodeclaration E are elements of $\mathcal{P}(\mathbf{P})^I$, i. e. variable interpretations for the logic.

2.3 Characteristic Endodeclarations

The aim of this section is to investigate how we can characterize all processes $p \in \mathbf{P}$ up to a binary relation S over processes (such as an equivalence or a preorder) using a declaration. To achieve this aim, we take $I = \mathbf{P}$ in the definitions in the previous section and consider the logic $\mathcal{L}(\mathbf{P})$. We have seen that each endodeclaration $E \in \mathcal{L}(\mathbf{P})^{\mathbf{P}}$ induces an endofunction $\llbracket E \rrbracket$ on the complete lattice of variable interpretations $\mathcal{P}(\mathbf{P})^{\mathbf{P}}$. Our goal is to find an appropriate endodeclaration, such that the greatest fixed point of its induced endofunction characterizes every process in \mathbf{P} .

Definition 2.13 An endodeclaration E for the logic $\mathcal{L}(\mathbf{P})$ characterizes $S \subseteq \mathbf{P} \times \mathbf{P}$ iff for each $p, q \in \mathbf{P}$,

$$(p, q) \in S \text{ iff } q \in (\nu \llbracket E \rrbracket)(p).$$

In many of our examples, we seek declarations that characterize a relation, such as similarity or bisimilarity, which is of the form $\nu \mathcal{F}$, where \mathcal{F} is a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$. In what follows, we will describe how we can devise a characterizing declaration for a relation that is obtained as a fixed point of a monotone endofunction, which can be expressed in the logic. In Definition 2.15 to follow, we use the notation introduced in Definition 2.14 below.

Definition 2.14 If $S \subseteq \mathbf{P} \times \mathbf{P}$ we define the variable interpretation $\sigma_S \in \mathcal{P}(\mathbf{P})^{\mathbf{P}}$ associated to S by

$$\sigma_S(p) = \{q \in \mathbf{P} \mid (p, q) \in S\}, \text{ for each } p \in \mathbf{P}.$$

Thus σ_S assigns to p all those processes q that are related to it via S .

Definition 2.15 We say that an endodeclaration E for $\mathcal{L}(\mathbf{P})$ expresses a monotone endofunction \mathcal{F} on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$ when

$$(p, q) \in \mathcal{F}(S) \text{ iff } \sigma_S, q \models E(p),$$

for every relation $S \subseteq \mathbf{P} \times \mathbf{P}$ and every $p, q \in \mathbf{P}$.

Now we are ready to state the main theorem of this paper.

Theorem 2.16 Let \mathcal{F} be a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$ and E an endodeclaration for $\mathcal{L}(\mathbf{P})$ that expresses \mathcal{F} . Then E characterizes $\nu \mathcal{F}$.

We use the following result to prove the theorem. We begin with a definition.

Definition 2.17 Let $\Phi : \mathcal{P}(\mathbf{P} \times \mathbf{P}) \rightarrow \mathcal{P}(\mathbf{P})^{\mathbf{P}}$ be defined by $\Phi(S) = \sigma_S$.

Lemma 2.18 $\Phi : \mathcal{P}(\mathbf{P} \times \mathbf{P}) \rightarrow \mathcal{P}(\mathbf{P})^{\mathbf{P}}$ is an isomorphism.

Proof To show that Φ is monotone, assume that $S_1, S_2 \in \mathcal{P}(\mathbf{P} \times \mathbf{P})$ and $S_1 \subseteq S_2$. Then, for any $p \in \mathbf{P}$, if $q \in \Phi(S_1)(p)$ then $(p, q) \in S_1$, so $(p, q) \in S_2$, so $q \in \Phi(S_2)(p)$, and thus $\Phi(S_1)(p) \subseteq \Phi(S_2)(p)$. We conclude $\Phi(S_1) \subseteq \Phi(S_2)$.

Next, define $\Phi^{-1} : \mathcal{P}(\mathbf{P})^{\mathbf{P}} \rightarrow \mathcal{P}(\mathbf{P} \times \mathbf{P})$ to map σ to $\{(p, q) \in \mathbf{P} \times \mathbf{P} \mid q \in \sigma(p)\}$. To show that Φ^{-1} is monotone, suppose that $\sigma_1, \sigma_2 \in \mathcal{P}(\mathbf{P})^{\mathbf{P}}$ and $\sigma_1 \subseteq \sigma_2$. If $(p, q) \in \Phi^{-1}(\sigma_1)$ then $q \in \sigma_1(p)$ so $q \in \sigma_2(p)$ so $(p, q) \in \Phi^{-1}(\sigma_2)$. We conclude $\Phi^{-1}(\sigma_1) \subseteq \Phi^{-1}(\sigma_2)$.

Finally we need to show that Φ and Φ^{-1} are inverse.

- Let $S \in \mathcal{P}(\mathbf{P} \times \mathbf{P})$. For any $p, q \in \mathbf{P}$ we have

$$\begin{aligned} (p, q) \in \Phi^{-1}(\Phi(S)) &\Leftrightarrow q \in \Phi(S)(p) \\ &\Leftrightarrow q \in \sigma_S(p) \\ &\Leftrightarrow (p, q) \in S \end{aligned}$$

We conclude $\Phi^{-1}(\Phi(S)) = S$.

- Let $\sigma \in \mathcal{P}(\mathbf{P})^{\mathbf{P}}$ and $p \in \mathbf{P}$. For any $q \in \mathbf{P}$ we have

$$\begin{aligned} q \in \Phi(\Phi^{-1}(\sigma))(p) &\Leftrightarrow (p, q) \in \Phi^{-1}(\sigma) \\ &\Leftrightarrow q \in \sigma(p) \end{aligned}$$

so $\Phi(\Phi^{-1}(\sigma))(p) = \sigma(p)$. We conclude $\Phi(\Phi^{-1}(\sigma)) = \sigma$. □

Now we are ready to prove Theorem 2.16.

Proof (of Theorem 2.16). To prove the theorem, we first prove that $\llbracket E \rrbracket \circ \Phi = \Phi \circ \mathcal{F}$ or equivalently that the following diagram commutes.

$$\begin{array}{ccc} \mathcal{P}(\mathbf{P} \times \mathbf{P}) & \xrightarrow{\Phi} & \mathcal{P}(\mathbf{P})^{\mathbf{P}} \\ \mathcal{F} \downarrow & & \downarrow \llbracket E \rrbracket \\ \mathcal{P}(\mathbf{P} \times \mathbf{P}) & \xrightarrow{\Phi} & \mathcal{P}(\mathbf{P})^{\mathbf{P}} \end{array} \quad (1)$$

To prove (1), we proceed as follows. Let $S \subseteq \mathbf{P} \times \mathbf{P}$. Then we have

$$(\llbracket E \rrbracket \circ \Phi)(S) = \llbracket E \rrbracket(\Phi(S)) = \llbracket E \rrbracket \sigma_S,$$

and

$$(\Phi \circ \mathcal{F})(S) = \Phi(\mathcal{F}(S)) = \sigma_{\mathcal{F}(S)}.$$

To prove (1), it is therefore sufficient to prove that $\llbracket E \rrbracket \sigma_S = \sigma_{\mathcal{F}(S)}$. Towards proving this, assume that $p \in \mathbf{P}$. Since E expresses \mathcal{F} (in the sense of Definition 2.15),

$$(\llbracket E \rrbracket \sigma_S)(p) = \llbracket E(p) \rrbracket \sigma_S = \{q \mid \sigma_S, q \models E(p)\} = \{q \mid (p, q) \in \mathcal{F}(S)\} = \sigma_{\mathcal{F}(S)}(p).$$

By Lemma 2.18, Φ is an isomorphism. Hence, Theorem 2.3 yields that $\Phi(v\mathcal{F}) = v\llbracket E \rrbracket$. As $(p, q) \in v\mathcal{F}$ iff $q \in \sigma_{v\mathcal{F}}(p)$ and $\sigma_{v\mathcal{F}}(p) = \Phi(v\mathcal{F})(p)$, this implies that, for any $p, q \in \mathbf{P}$,

$$(p, q) \in v\mathcal{F} \Leftrightarrow q \in (v\llbracket E \rrbracket)(p).$$

Therefore E characterizes $v\mathcal{F}$ (in the sense of Definition 2.13). □

3 Applications

Our order of business in this section will be to apply Theorem 2.16 to obtain characteristic formulae for various behavioral relations, which are defined as greatest fixed points of monotone endofunctions, over variations on labelled transition systems. Most of the results we present in this section are known from the literature on concurrency theory. The characteristic formula constructions presented in Sections 3.1.3, 3.1.4, 3.2.3, 3.2.4, and 3.4 are, to the best of our knowledge, new.

3.1 Types of similarity

In this section, we give examples of characteristic formulae for various types of similarity.

3.1.1 Simulation [45]

Given an \mathbf{A} -labelled transition system $P = (\mathbf{P}, \longrightarrow)$, and a relation $S \subseteq \mathbf{P} \times \mathbf{P}$, let $\mathcal{F}_{sim}(S)$ be defined by

$$(p, q) \in \mathcal{F}_{sim}(S) \text{ iff for every } a \in \mathbf{A} \text{ and } p' \in \mathbf{P}, \\ \text{if } p \xrightarrow{a} p' \text{ then there exists a } q' \in \mathbf{P} \text{ such that } q \xrightarrow{a} q' \text{ and } (p', q') \in S.$$

\mathcal{F}_{sim} is a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$ and its greatest fixed point, the *simulation preorder*, is denoted by \sqsubseteq_{sim} .

Following the recipe of Section 2.3, we seek an endodeclaration E_{sim} over $\mathcal{L}(\mathbf{P})$ expressing \mathcal{F}_{sim} in the sense of Definition 2.15. To this end, assume that $S \subseteq \mathbf{P} \times \mathbf{P}$ and observe that:

$$(p, q) \in \mathcal{F}_{sim}(S) \Leftrightarrow \forall a \in \mathbf{A}. \forall p' \in \mathbf{P}. (p \xrightarrow{a} p' \Rightarrow \exists q' \in \mathbf{P}. q \xrightarrow{a} q' \& (p', q') \in S) \\ \Leftrightarrow \sigma_S, q \models \bigwedge_{a \in \mathbf{A}} \bigwedge_{p' \in \mathbf{P}. p \xrightarrow{a} p'} \langle a \rangle X_{p'}.$$

This shows that the endodeclaration $E_{sim} : p \mapsto \bigwedge_{a \in \mathbf{A}} \bigwedge_{p' \in \mathbf{P}. p \xrightarrow{a} p'} \langle a \rangle X_{p'}$ expresses \mathcal{F}_{sim} . Therefore the following proposition follows from Theorem 2.16.

Proposition 3.1 The endodeclaration E_{sim} characterizes the preorder \sqsubseteq_{sim} .

Next we seek characteristic formulae for $\sqsubseteq_{opsim} = (\sqsubseteq_{sim})^{-1}$. Lemma 2.7 implies that $\sqsubseteq_{opsim} = \nu \mathcal{F}_{opsim}$ where $\mathcal{F}_{opsim} : S \mapsto (\mathcal{F}_{sim}(S^{-1}))^{-1}$. Now we proceed as follows: For all $p, q \in \mathbf{P}$,

$$(p, q) \in \mathcal{F}_{opsim}(S) \text{ iff } (q, p) \in \mathcal{F}_{sim}(S^{-1}) \text{ iff for every } a \in \mathbf{A} \text{ and } q' \in \mathbf{P}, \\ \text{if } q \xrightarrow{a} q' \text{ then there exists some } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } (q', p') \in S^{-1}$$

or equivalently,

$$(p, q) \in \mathcal{F}_{opsim}(S) \text{ iff for every } a \in \mathbf{A} \text{ and } q' \in \mathbf{P}, \\ \text{if } q \xrightarrow{a} q' \text{ then there exists some } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } (p', q') \in S.$$

This can be expressed in the logic $\mathcal{L}(\mathbf{P})$ in the following way:

$$(p, q) \in \mathcal{F}_{opsim}(S) \Leftrightarrow \forall a \in \mathbf{A}. \forall q' \in \mathbf{P}. (q \xrightarrow{a} q' \Rightarrow \exists p' \in \mathbf{P}. p \xrightarrow{a} p' \& (p', q') \in S) \\ \Leftrightarrow \sigma_S, q \models \bigwedge_{a \in \mathbf{A}} [a] \bigvee_{p' \in \mathbf{P}. p \xrightarrow{a} p'} X_{p'}.$$

This shows that the endodeclaration $E_{opsim} : p \mapsto \bigwedge_{a \in \mathbf{A}} [a] \bigvee_{p' \in \mathbf{P}. p \xrightarrow{a} p'} X_{p'}$ expresses \mathcal{F}_{opsim} . Therefore Theorem 2.16 gives us the following:

Proposition 3.2 The endodeclaration E_{opsim} characterizes the preorder \sqsubseteq_{opsim} .

3.1.2 Ready simulation [6, 36]

Given an \mathbf{A} -labelled transition system $P = (\mathbf{P}, \longrightarrow)$ and a relation $S \subseteq \mathbf{P} \times \mathbf{P}$, let $\mathcal{F}_{RS}(S)$ be defined such that

$$(p, q) \in \mathcal{F}_{RS}(S) \text{ iff for every } a \in \mathbf{A}, \\ 1. \text{ if } p \xrightarrow{a} p' \text{ then there exists some } q' \in \mathbf{P}, \text{ such that } q \xrightarrow{a} q' \text{ and } (p', q') \in S, \text{ and} \\ 2. \text{ if } q \xrightarrow{a} \text{ then } p \xrightarrow{a}.$$

Note that \mathcal{F}_{RS} is monotone; its greatest fixed point is the *ready simulation preorder*, which is denoted by \sqsubseteq_{RS} .

Seeking a declaration for $\mathcal{L}(\mathbf{P})$ that expresses \mathcal{F}_{RS} , we reason as follows:

$$\begin{aligned} (p, q) \in \mathcal{F}_{RS}(S) &\Leftrightarrow \forall a \in \mathbf{A}. \forall p' \in \mathbf{P}. (p \xrightarrow{a} p' \Rightarrow \exists q' \in \mathbf{P}. (q \xrightarrow{a} q' \& (p', q') \in S)) \& \\ &\quad \forall a \in \mathbf{A}. (p \not\xrightarrow{a} \Rightarrow q \not\xrightarrow{a}) \\ &\Leftrightarrow \sigma_S, q \models \bigwedge_{a \in \mathbf{A}} \bigwedge_{p' \in \mathbf{P}. p \xrightarrow{a} p'} \langle a \rangle X_{p'} \wedge \bigwedge_{a \in \mathbf{A}. p \not\xrightarrow{a}} [a] ff. \end{aligned}$$

Thus the endodeclaration E_{RS} for $\mathcal{L}(\mathbf{P})$ given by

$$E_{RS} : p \mapsto \bigwedge_{a \in \mathbf{A}} \bigwedge_{p' \in \mathbf{P}. p \xrightarrow{a} p'} \langle a \rangle X_{p'} \wedge \bigwedge_{a \in \mathbf{A}. p \not\xrightarrow{a}} [a] ff.$$

expresses \mathcal{F}_{RS} . Therefore Theorem 2.16 gives us the following proposition.

Proposition 3.3 *The endodeclaration E_{RS} characterizes the preorder \sqsubseteq_{RS} .*

3.1.3 Conformance simulation [20]

Given an \mathbf{A} -labelled transition system $P = (\mathbf{P}, \longrightarrow)$ and $S \subseteq \mathbf{P} \times \mathbf{P}$, let $\mathcal{F}_{CS}(S)$ be defined such that

$$\begin{aligned} (p, q) \in \mathcal{F}_{CS}(S) &\text{ iff for every } a \in \mathbf{A} \text{ and } q' \in \mathbf{P}, \\ &1. \text{ if } q \xrightarrow{a} q' \text{ and } p \xrightarrow{a}, \text{ then there exists } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } (p', q') \in S, \text{ and} \\ &2. \text{ if } p \xrightarrow{a}, \text{ then } q \xrightarrow{a}. \end{aligned}$$

Note that \mathcal{F}_{CS} is monotone. Its greatest fixed point is called *conformance simulation preorder* and is denoted by \sqsubseteq_{CS} .

Seeking a \mathbf{P} -indexed declaration that expresses \mathcal{F}_{CS} , we reason as follows:

$$\begin{aligned} (p, q) \in \mathcal{F}_{CS}(S) &\Leftrightarrow \forall a \in \mathbf{A}. \forall q' \in \mathbf{P}. (q \xrightarrow{a} q' \& p \xrightarrow{a} \Rightarrow \exists p' \in \mathbf{P}. (p \xrightarrow{a} p' \& (p', q') \in S)) \\ &\quad \& \quad \forall a \in \mathbf{A}. (p \xrightarrow{a} \Rightarrow q \xrightarrow{a}) \\ &\Leftrightarrow \sigma_S, q \models \bigwedge_{a \in \mathbf{A}. p \xrightarrow{a}} [a] \bigvee_{p' \in \mathbf{P}. p \xrightarrow{a} p'} X_{p'} \wedge \bigwedge_{a \in \mathbf{A}. p \not\xrightarrow{a}} \langle a \rangle tt \\ &\Leftrightarrow \sigma_S, q \models \bigwedge_{a \in \mathbf{A}. p \xrightarrow{a}} (\langle a \rangle tt \wedge [a] \bigvee_{p' \in \mathbf{P}. p \xrightarrow{a} p'} X_{p'}) \end{aligned}$$

Thus the endodeclaration $E_{CS} : p \mapsto \bigwedge_{a \in \mathbf{A}. p \xrightarrow{a}} (\langle a \rangle tt \wedge [a] \bigvee_{p' \in \mathbf{P}. p \xrightarrow{a} p'} X_{p'})$ expresses \mathcal{F}_{CS} . Therefore Theorem 2.16 gives us the following proposition.

Proposition 3.4 *The endodeclaration E_{CS} characterizes the preorder \sqsubseteq_{CS} .*

3.1.4 Extended simulation [55]

We now consider \mathbf{A} -labelled transition systems extended with a preorder relation $\sqsubseteq_{\mathbf{A}}$ over the set \mathbf{A} of labels. Given an extended \mathbf{A} -labelled transition system $(\mathbf{P}, \longrightarrow, \sqsubseteq_{\mathbf{A}})$ and $S \subseteq \mathbf{P} \times \mathbf{P}$, we define \mathcal{F}_{ext} such that

$$\begin{aligned} (p, q) \in \mathcal{F}_{ext}(S) &\text{ iff for every } a \in \mathbf{A}, \\ &\text{if } p \xrightarrow{a} p' \text{ then there exist } q' \in \mathbf{P} \text{ and } b \in \mathbf{A} \text{ such that } a \sqsubseteq_{\mathbf{A}} b, q \xrightarrow{b} q', \text{ and } (p', q') \in S. \end{aligned}$$

We denote the greatest fixed point of \mathcal{F}_{ext} by \sqsubseteq_{ext} .

Seeking a \mathbf{P} -indexed endodeclaration that expresses \mathcal{F}_{ext} , we reason as follows: For each $S \subseteq \mathbf{P} \times \mathbf{P}$ and $p, q \in \mathbf{P}$,

$$\begin{aligned} (p, q) \in \mathcal{F}_{ext}(S) &\Leftrightarrow \forall a \in \mathbf{A}. \forall p' \in \mathbf{P}. (p \xrightarrow{a} p' \Rightarrow \exists b \in \mathbf{A}. \exists q' \in \mathbf{P}. (a \sqsubseteq_{\mathbf{A}} b \ \& \ q \xrightarrow{b} q' \ \& \ (p', q') \in S)) \\ &\Leftrightarrow \sigma_s, q \models \bigwedge_{a \in \mathbf{A}} \bigwedge_{p' \in \mathbf{P}. p \xrightarrow{a} p'} \bigvee_{b \in \mathbf{A}. a \sqsubseteq_{\mathbf{A}} b} \langle b \rangle X_{p'}. \end{aligned}$$

Thus the endodeclaration $E_{ext} : p \mapsto \bigwedge_{a \in \mathbf{A}} \bigwedge_{p' \in \mathbf{P}. p \xrightarrow{a} p'} \bigvee_{b \in \mathbf{A}. a \sqsubseteq_{\mathbf{A}} b} \langle b \rangle X_{p'}$ expresses \mathcal{F}_{ext} , and by Theorem 2.16, we have the following proposition.

Proposition 3.5 *The endodeclaration E_{ext} characterizes the preorder \sqsubseteq_{ext} .*

3.2 Types of bisimilarity

In this section, we provide examples of characteristic formulae for various types of bisimilarity.

3.2.1 Strong bisimulation [45, 41]

Given an \mathbf{A} -labelled transition system $P = (\mathbf{P}, \longrightarrow)$ and a relation $S \subseteq \mathbf{P} \times \mathbf{P}$, let $\mathcal{F}_{bisim}(S)$ be defined by

$$\begin{aligned} (p, q) \in \mathcal{F}_{bisim}(S) &\text{ iff for every } a \in \mathbf{A}, \\ &1. \text{ if } p \xrightarrow{a} p', \text{ then there exists some } q' \in \mathbf{P} \text{ such that } q \xrightarrow{a} q' \text{ and } (p', q') \in S, \text{ and} \\ &2. \text{ if } q \xrightarrow{a} q', \text{ then there exists some } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } (p', q') \in S. \end{aligned}$$

\mathcal{F}_{bisim} is a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$. Its greatest fixed point, known as *bisimulation equivalence*, is denoted by \sim_{bisim} .

Toward finding a declaration that expresses \mathcal{F}_{bisim} , we observe in the following lemma that if we have declarations expressing the monotone endofunctions \mathcal{F}_1 and \mathcal{F}_2 , we can obtain one expressing $S \mapsto \mathcal{F}_1(S) \cap \mathcal{F}_2(S)$. This generalizes to arbitrary intersections, as follows.

Lemma 3.6 *Let $\{\mathcal{F}_j\}_{j \in J}$ be a family of monotone endofunctions on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$. For each $j \in J$, let E_j be a \mathbf{P} -indexed endodeclaration expressing \mathcal{F}_j . Then the \mathbf{P} -indexed declaration $p \mapsto \bigwedge_{j \in J} E_j$ expresses $S \mapsto \bigcap_{j \in J} \mathcal{F}_j(S)$.*

We note that

$$\mathcal{F}_{bisim} : S \mapsto \mathcal{F}_{sim}(S) \cap \mathcal{F}_{opsim}(S)$$

and observe that since E_{sim} and E_{opsim} express \mathcal{F}_{sim} and \mathcal{F}_{opsim} respectively, then $p \mapsto E_{sim}(p) \wedge E_{opsim}(p)$ expresses $S \mapsto \mathcal{F}_{sim}(S) \cap \mathcal{F}_{opsim}(S)$. Thus the endodeclaration for $\mathcal{L}(\mathbf{P})$ given by

$$E_{bisim} : p \mapsto \left(\bigwedge_{a \in \mathbf{A}} \bigwedge_{p' \in \mathbf{P}. p \xrightarrow{a} p'} \langle a \rangle X_{p'} \right) \wedge \left(\bigwedge_{a \in \mathbf{A}} [a] \bigvee_{p' \in \mathbf{P}. p \xrightarrow{a} p'} X_{p'} \right)$$

expresses \mathcal{F}_{bisim} . The following proposition now follows from Theorem 2.16.

Proposition 3.7 *The endodeclaration E_{bisim} characterizes the equivalence \sim_{bisim} .*

The endodeclaration E_{bisim} is exactly the one proposed in [28].

3.2.2 Weak bisimulation [41]

Let $P = (\mathbf{P}, \longrightarrow)$ be an \mathbf{A} -labelled transition system with one label $\tau \in \mathbf{A}$ to be viewed as a *silent step*. We derive a family $\{\overset{a}{\Rightarrow}\}_{a \in \mathbf{A}}$ of transition relations over \mathbf{P} as follows:

- $\overset{\tau}{\Rightarrow}$ is the reflexive transitive closure $\overset{\tau}{\longrightarrow}^*$ of $\overset{\tau}{\longrightarrow}$, and
- $\overset{a}{\Rightarrow}$ is the composition $\overset{\tau}{\Rightarrow} \circ \overset{a}{\longrightarrow} \circ \overset{\tau}{\Rightarrow}$, for $a \neq \tau$.¹

We consider a variation of the language of HML, where the family of modal operators $\langle a \rangle$ and $[a]$ are replaced with the family of modal operators $\langle\langle a \rangle\rangle$ and $[[a]]$ for the derived relations $\overset{a}{\Rightarrow}$, with $a \in \mathbf{A}$. Thus the semantics is:

$$\begin{aligned} \sigma, p \models \langle\langle a \rangle\rangle F_1 & \text{ iff } \sigma, p' \models F_1 \text{ for some } p' \text{ for which } p \overset{a}{\Rightarrow} p' \\ \sigma, p \models [[a]] F_1 & \text{ iff } \sigma, p' \models F_1 \text{ for all } p' \text{ for which } p \overset{a}{\Rightarrow} p'. \end{aligned}$$

Let $\mathcal{F}_{wbsm}(S)$ be defined such that

- $(p, q) \in \mathcal{F}_{wbsm}(S)$ iff for every $a \in \mathbf{A}$,
1. if $p \overset{a}{\Rightarrow} p'$, then there exists $q' \in \mathbf{P}$ such that $q \overset{a}{\Rightarrow} q'$ and $(p', q') \in S$, and
 2. if $q \overset{a}{\Rightarrow} q'$, then there exists $p' \in \mathbf{P}$ such that $p \overset{a}{\Rightarrow} p'$ and $(p', q') \in S$.

As \mathcal{F}_{wbsm} is monotone, it has a greatest (post-)fixed point, which is the seminal notion of weak bisimulation equivalence that we denote by \sim_{wbsm} .

Note that \mathcal{F}_{wbsm} is defined exactly as \mathcal{F}_{bism} , but with relations $\overset{a}{\longrightarrow}$ replaced by derived relations $\overset{a}{\Rightarrow}$. As $[[a]]$ and $\langle\langle a \rangle\rangle$ are the corresponding modalities for the derived symbols, we replace every instance of $[a]$ and $\langle a \rangle$ in E_{bism} by $[[a]]$ and $\langle\langle a \rangle\rangle$. Thus the endodeclaration for $\mathcal{L}(\mathbf{P})$ given by

$$E_{wbsm} : p \mapsto \left(\bigwedge_{a \in \mathbf{A}} \bigwedge_{p' \in \mathbf{P}. p \overset{a}{\Rightarrow} p'} \langle\langle a \rangle\rangle X_{p'} \right) \wedge \left(\bigwedge_{a \in \mathbf{A}} \bigvee_{p' \in \mathbf{P}. p \overset{a}{\Rightarrow} p'} X_{p'} \right)$$

expresses \mathcal{F}_{wbsm} , and hence by Theorem 2.16, we have the following proposition.

Proposition 3.8 *The endodeclaration E_{wbsm} characterizes the equivalence \sim_{wbsm} .*

Note that since $q \overset{\tau}{\Rightarrow} q$ holds for each q , unlike in the case for E_{bism} , the formula for E_{wbsm} can never have conjuncts of the form $[[\tau]]ff$.

3.2.3 Back and forth bisimulation

In this section we will introduce a new semantic equivalence. This is a variant of the back and forth bisimulation equivalence introduced in [15] that allows for several possible past states. The semantics introduced in [15] assumes that the past is unique and consequently the derived equivalence coincides with the standard strong bisimulation equivalence. This is not the case for the multiple possible past semantics considered here. The introduction of this behavioural equivalence serves as a stepping stone towards the one introduced in the subsequent section.

Given $P = (\mathbf{P}, \longrightarrow)$ and $S \subseteq \mathbf{P} \times \mathbf{P}$, let $\mathcal{F}_{bfb}(S)$ be defined such that

- $(p, q) \in \mathcal{F}_{bfb}(S)$ iff $(p, q) \in \mathcal{F}_{bism}(S)$ and for every $a \in \mathbf{A}$
1. $\forall p' \in \mathbf{P}. p' \overset{a}{\longrightarrow} p \Rightarrow \exists q' \in \mathbf{P}. q' \overset{a}{\longrightarrow} q$ and $(p', q') \in S$ and

¹Composition \circ between relations S and R is defined as $R \circ S = \{(x, z) \mid \text{there is } y \text{ such that } (x, y) \in R \ \& \ (y, z) \in S\}$.

$$2. \forall q' \in \mathbf{P}. q' \xrightarrow{a} q \Rightarrow \exists p' \in \mathbf{P}. p' \xrightarrow{a} p \text{ and } (p', q') \in S.$$

\mathcal{F}_{bfb} is a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$, and its greatest fixed point, called *back and forth bisimulation equivalence*, is denoted by \sim_{bfb} .

To express such behaviour in the logical language considered so far, we add two operators $\langle \bar{a} \rangle$ and $[\bar{a}]$ to it for every $a \in \mathbf{A}$. The semantics for these is given by

$$\begin{aligned} \sigma, p \models \langle \bar{a} \rangle F_1 & \text{ iff } \sigma, p' \models F_1 \text{ for some } p' \text{ for which } p' \xrightarrow{a} p \text{ and} \\ \sigma, p \models [\bar{a}] F_1 & \text{ iff } \sigma, p' \models F_1 \text{ for all } p' \text{ for which } p' \xrightarrow{a} p. \end{aligned}$$

Clearly $\llbracket F \rrbracket$ is monotone for each F in the extended language.

Toward finding an endodeclaration that expresses \sim_{bfb} , we reason similarly to the way we did for finding an endodeclaration that express \sim_{bisim} , and observe the following equivalence:

$$(p, q) \in \mathcal{F}_{bfb}(S) \Leftrightarrow \sigma_{S, q} \models E_{bisim}(p) \wedge \left(\bigwedge_{a \in \mathbf{A}} \bigwedge_{p' \in \mathbf{P}. p' \xrightarrow{a} p} \langle \bar{a} \rangle X_{p'} \right) \wedge \left(\bigwedge_{a \in \mathbf{A}} [\bar{a}] \bigvee_{p' \in \mathbf{P}. p' \xrightarrow{a} p} X_{p'} \right).$$

Then the endodeclaration given by

$$E_{bfb} : p \mapsto E_{bisim}(p) \wedge \left(\bigwedge_{a \in \mathbf{A}} \bigwedge_{p' \in \mathbf{P}. p' \xrightarrow{a} p} \langle \bar{a} \rangle X_{p'} \right) \wedge \left(\bigwedge_{a \in \mathbf{A}} [\bar{a}] \bigvee_{p' \in \mathbf{P}. p' \xrightarrow{a} p} X_{p'} \right)$$

expresses \mathcal{F}_{bfb} , and hence, by Theorem 2.16, we have the following proposition.

Proposition 3.9 *The endodeclaration E_{bfb} characterizes the equivalence \sim_{bfb} .*

3.2.4 Back and forth bisimulation with indistinguishable states [17]

In this section we consider a version of the back and forth bisimulation from the previous section where some of the states are considered indistinguishable by some external agents. For this purpose we augment our notion of labelled transition systems with a set \mathcal{I} of identities (or agents) and a family of equivalence relations $\{\cdot^i \subseteq \mathbf{P} \times \mathbf{P} \mid i \in \mathcal{I}\}$. Intuitively $p \cdot^i q$ means that agent i cannot distinguish p from q . Such a structure is called an annotated labelled transition system [17].

Given such a structure, let $\mathcal{F}_{bfbid}(S)$ be defined such that

$$(p, q) \in \mathcal{F}_{bfbid}(S) \text{ iff } (p, q) \in \mathcal{F}_{bfb}(S) \text{ and for every } a \in \mathbf{A} \text{ and } i \in \mathcal{I},$$

1. $\forall p' \in \mathbf{P}. p \cdot^i p' \Rightarrow \exists q' \in \mathbf{P}. q \cdot^i q' \text{ and } (p', q') \in S$ and
2. $\forall q' \in \mathbf{P}. q \cdot^i q' \Rightarrow \exists p' \in \mathbf{P}. p \cdot^i p' \text{ and } (p', q') \in S.$

We denote the greatest fixed point of \mathcal{F}_{bfbid} by \sim_{bfbid} . We use the logical language for back and forth bisimulation from Section 3.2.3 and add to it the operators $\langle i \rangle$ and $[i]$ for each $i \in \mathcal{I}$. The semantics for these operators is given by

$$\begin{aligned} \sigma, p \models \langle i \rangle F_1 & \text{ iff } \sigma, p' \models F_1 \text{ for some } p' \text{ for which } p \cdot^i p' \text{ and} \\ \sigma, p \models [i] F_1 & \text{ iff } \sigma, p' \models F_1 \text{ for all } p' \text{ for which } p \cdot^i p'. \end{aligned}$$

Clearly $\llbracket F \rrbracket$ is monotone for each F in the extended language.

Toward finding an endodeclaration that expresses \sim_{bfbid} , we reason similarly to the way we did for finding an endodeclarations that express \sim_{bfb} and \sim_{bism} , and observe the following equivalence:

$$(p, q) \in \mathcal{F}_{bfb}(S) \Leftrightarrow \sigma_{S, q} \models E_{bfb}(p) \wedge \left(\bigwedge_{i \in \mathcal{I}} \bigwedge_{p' \in \mathbf{P}. p \cdot \cdot^i p'} \langle i \rangle X_{p'} \right) \wedge \left(\bigwedge_{i \in \mathcal{I}} [i] \bigvee_{p' \in \mathbf{P}. p \cdot \cdot^i p'} X_{p'} \right).$$

Then the endodeclaration given by

$$E_{bfbid} : p \mapsto E_{bfb}(p) \wedge \left(\bigwedge_{i \in \mathcal{I}} \bigwedge_{p' \in \mathbf{P}. p \cdot \cdot^i p'} \langle i \rangle X_{p'} \right) \wedge \left(\bigwedge_{i \in \mathcal{I}} [i] \bigvee_{p' \in \mathbf{P}. p \cdot \cdot^i p'} X_{p'} \right)$$

expresses \sim_{bfbid} , and hence we have the following proposition.

Proposition 3.10 *The endodeclaration E_{bfbid} characterizes the equivalence \sim_{bfbid} .*

As an immediate consequence of the existence of this characteristic formula, we obtain a behavioural characterization of the equivalence over states in an annotated labelled transition system induced by the epistemic logic studied in [17]. (More precisely, the logic we study in this section may be seen as the positive version of the one studied in [17], where we use the modal operator $[i]$ in lieu of K_i , read “agent i knows”, and its dual.) This solves a problem that was left open in the aforementioned reference.

Theorem 3.11 *Let $p, q \in \mathbf{P}$. Then $p \sim_{bfbid} q$ if, and only if, p and q satisfy the same formulae expressible in the logic considered in this section.*

Proof The implication from left to right may be shown using standard lines. The one from right to left is an immediate consequence of Proposition 3.10. \square

3.3 Transition Systems with Divergence

We frequently wish to consider systems in which some processes may diverge. We write $p \uparrow$ to indicate that p may diverge.

Definition 3.12 *An \mathbf{A} -labelled transition system with divergence consists of*

- a set \mathbf{P} of processes,
- a transition relation $\longrightarrow \subseteq \mathbf{P} \times \mathbf{P}$
- and a predicate $\uparrow \subseteq \mathbf{P}$

We write $p \uparrow$ for $p \in \uparrow$ and $p \not\uparrow$ for $p \notin \uparrow$.

We define *Hennessy-Milner logic with divergence* as follows. For a set I , we write $\mathcal{L}(I)$ for the set of formulae given by

$$\begin{aligned}
F ::= & X_i && (i \in I) \\
& | \bigwedge_{k \in K} F_k && (K \text{ a finite set}) \\
& | \bigvee_{k \in K} F_k && (K \text{ a finite set}) \\
& | \uparrow \\
& | \Downarrow \\
& | \langle a \rangle F && (a \in \mathbf{A}) \\
& | \langle a \rangle^{\wedge \uparrow} F && (a \in \mathbf{A}) \\
& | \langle a \rangle^{\wedge \Downarrow} F && (a \in \mathbf{A}) \\
& | \langle a \rangle^{\vee \uparrow} F && (a \in \mathbf{A}) \\
& | \langle a \rangle^{\vee \Downarrow} F && (a \in \mathbf{A}) \\
& | [a] F && (a \in \mathbf{A}) \\
& | [a]^{\wedge \uparrow} F && (a \in \mathbf{A}) \\
& | [a]^{\wedge \Downarrow} F && (a \in \mathbf{A}) \\
& | [a]^{\vee \uparrow} F && (a \in \mathbf{A}) \\
& | [a]^{\vee \Downarrow} F && (a \in \mathbf{A})
\end{aligned}$$

Let $P = (\mathbf{P}, \longrightarrow, \uparrow)$ be an \mathbf{A} -labelled transition system with divergence. We then define a satisfaction relation \models from $\mathcal{P}(\mathbf{P})^I \times \mathbf{P}$ to $\mathcal{L}(I)$ in the following inductive manner. Here $\sigma \in \mathcal{P}(\mathbf{P})^I$ and $p \in \mathbf{P}$.

$$\begin{aligned}
\sigma, p \models X_i & \Leftrightarrow p \in \sigma(i) \\
\sigma, p \models \bigwedge_{k \in K} F_k & \Leftrightarrow \sigma, p \models F_k \text{ for all } k \in K \\
\sigma, p \models \bigvee_{k \in K} F_k & \Leftrightarrow \sigma, p \models F_k \text{ for some } k \in K \\
\sigma, p \models \uparrow & \Leftrightarrow p \uparrow \\
\sigma, p \models \Downarrow & \Leftrightarrow p \Downarrow \\
\sigma, p \models \langle a \rangle F & \Leftrightarrow \text{there is } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p \models \langle a \rangle^{\wedge \uparrow} F & \Leftrightarrow p \uparrow \text{ and there is } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p \models \langle a \rangle^{\wedge \Downarrow} F & \Leftrightarrow p \Downarrow \text{ and there is } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p \models \langle a \rangle^{\vee \uparrow} F & \Leftrightarrow p \uparrow \text{ or there is } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p \models \langle a \rangle^{\vee \Downarrow} F & \Leftrightarrow p \Downarrow \text{ or there is } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p \models [a] F & \Leftrightarrow \text{for all } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F \\
\sigma, p \models [a]^{\wedge \uparrow} F & \Leftrightarrow p \uparrow \text{ and for all } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F \\
\sigma, p \models [a]^{\wedge \Downarrow} F & \Leftrightarrow p \Downarrow \text{ and for all } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F \\
\sigma, p \models [a]^{\vee \uparrow} F & \Leftrightarrow p \uparrow \text{ or for all } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F \\
\sigma, p \models [a]^{\vee \Downarrow} F & \Leftrightarrow p \Downarrow \text{ or for all } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F
\end{aligned}$$

Occasionally, we want to consider transition systems in which not only divergence is possible but also other behaviours such as deadlock or crash [1]. In general, we have a finite set \mathbf{E} which contains all these erroneous behaviours.

Definition 3.13 Let \mathbf{E} be a finite set. An \mathbf{A} -labelled transition system with \mathbf{E} -errors consists of

- a finite set \mathbf{P} of processes,
- a transition relation $\longrightarrow \subseteq \mathbf{P} \times \mathbf{A} \times \mathbf{P}$, and
- an error relation $\not\downarrow \subseteq \mathbf{P} \times \mathbf{E}$.

An ordinary transition system corresponds to the case $\mathbf{E} = \emptyset$. A transition system with divergence corresponds to the case $\mathbf{E} = \{\uparrow\}$.

We define *Hennessey-Milner logic with \mathbf{E} -errors* as follows. For a set I , we write $\mathcal{L}(I)$ for the set of formulae given by

$$\begin{aligned}
F & ::= X_i && (i \in I) \\
& \mid \bigwedge_{k \in K} F_k && (K \text{ a finite set}) \\
& \mid \bigvee_{k \in K} F_k && (K \text{ a finite set}) \\
& \mid \nabla^{\mathcal{D}} && (\mathcal{D} \subseteq \mathcal{P}(\mathbf{E})) \\
& \mid \langle a \rangle_{\mathcal{D}}^{\mathcal{D}'} F && (a \in \mathbf{A} \text{ and } \mathcal{D}, \mathcal{D}' \text{ disjoint subsets of } \mathcal{P}(\mathbf{E})) \\
& \mid [a]_{\mathcal{D}}^{\mathcal{D}'} F && (a \in \mathbf{A} \text{ and } \mathcal{D}, \mathcal{D}' \text{ disjoint subsets of } \mathcal{P}(\mathbf{E}))
\end{aligned}$$

Let $P = (\mathbf{P}, \longrightarrow, \not\downarrow)$ be an \mathbf{A} -labelled transition system with \mathbf{E} -errors. We then define a relation \models from $\mathcal{P}(\mathbf{P})^I \times \mathbf{P}$ to $\mathcal{L}(I)$ in the following inductive manner. Here $\sigma \in \mathcal{P}(\mathbf{P})^I$ and $p \in \mathbf{P}$, and we write $\text{Errors}(p) = \{e \in \mathbf{E} \mid p \not\downarrow e\}$.

$$\begin{aligned}
\sigma, p \models X_i & \Leftrightarrow p \in \sigma(i) \\
\sigma, p \models \bigwedge_{k \in K} F_k & \Leftrightarrow \sigma, p \models F_k \text{ for all } k \in K \\
\sigma, p \models \bigvee_{k \in K} F_k & \Leftrightarrow \sigma, p \models F_k \text{ for some } k \in K \\
\sigma, p \models \nabla^{\mathcal{D}} & \Leftrightarrow \text{Errors}(p) \in \mathcal{D} \\
\sigma, p \models \langle a \rangle_{\mathcal{D}}^{\mathcal{D}'} F & \Leftrightarrow \text{Errors}(p) \in \mathcal{D}, \text{ or} \\
& \text{Errors}(p) \notin \mathcal{D}' \text{ and there is } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p \models [a]_{\mathcal{D}}^{\mathcal{D}'} F & \Leftrightarrow \text{Errors}(p) \in \mathcal{D}, \text{ or} \\
& \text{Errors}(p) \notin \mathcal{D}' \text{ and for all } p' \in \mathbf{P} \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F
\end{aligned}$$

Informally, for the modalities $\langle a \rangle_{\mathcal{D}}^{\mathcal{D}'}$ and $[a]_{\mathcal{D}}^{\mathcal{D}'}$, the superscript indicates which processes automatically ‘go to heaven’ and the subscript indicates which processes automatically ‘go to hell’. If p is a process which is not automatically assigned, then its fate depends on its actions.

It is easy to see that Hennessey-Milner logic with \mathbf{E} -errors reduces to ordinary Hennessey-Milner logic if \mathbf{E} is empty, and to Hennessey-Milner logic with divergence if \mathbf{E} is a singleton.

Dualization is defined as follows:

$$\begin{aligned}
\overline{X_i} & = X_i \\
\overline{\bigwedge_{k \in K} F_k} & = \bigvee_{k \in K} \overline{F_k} \\
\overline{\bigvee_{k \in K} F_k} & = \bigwedge_{k \in K} \overline{F_k} \\
\overline{\nabla^{\mathcal{D}}} & = \nabla^{(\mathcal{P}(\mathbf{E})) \setminus \mathcal{D}} \\
\overline{\langle a \rangle_{\mathcal{D}}^{\mathcal{D}'} F} & = [a]_{\mathcal{D}}^{\mathcal{D}'} \overline{F} \\
\overline{[a]_{\mathcal{D}}^{\mathcal{D}'} F} & = \langle a \rangle_{\mathcal{D}}^{\mathcal{D}'} \overline{F}
\end{aligned}$$

All the previously mentioned results about substitution and dualization, and about characteristic and co-characteristic formulae, adapt without difficulty to Hennessy-Milner logic with **E**-errors.

3.4 Divergence and Simulation

Let $P = (\mathbf{P}, \longrightarrow, \uparrow)$ be an **A**-labelled transition system with divergence. There are several different kinds of simulation on P . For a relation $S \subseteq \mathbf{P} \times \mathbf{P}$, we define the following relations.

- $(p, p') \in \mathcal{F}_{lower}(S)$ when for every $a \in \mathbf{A}$ and $q \in \mathbf{P}$ such that $p \xrightarrow{a} q$, there exists some $q' \in \mathbf{P}$ such that $p' \xrightarrow{a} q'$ and $(q, q') \in S$. (This is the same as $\mathcal{F}_{sim}(S)$, for the transition system $(\mathbf{P}, \longrightarrow)$.)
- $(p, p') \in \mathcal{F}_{incl}(S)$ when both the following hold:
 - for every $a \in \mathbf{A}$ and $q \in \mathbf{P}$ such that $p \xrightarrow{a} q$, there exists some $q' \in \mathbf{P}$ such that $p' \xrightarrow{a} q'$ and $(q, q') \in S$ and
 - if $p \uparrow$ then $p' \uparrow$.
- $(p, p') \in \mathcal{F}_{upper}(S)$ when $p \not\uparrow$ implies that both the following hold:
 1. $p' \not\uparrow$ and
 2. for all $a \in \mathbf{A}$ and $q' \in \mathbf{P}$ such that $p' \xrightarrow{a} q'$, there exists some $q \in \mathbf{P}$, such that $p \xrightarrow{a} q$ and $(q, q') \in S$.
- $(p, p') \in \mathcal{F}_{smash}(S)$ when $p \not\uparrow$ implies that all the following hold:
 1. $p' \not\uparrow$,
 2. for all $a \in \mathbf{A}$ and $q \in \mathbf{P}$ such that $p \xrightarrow{a} q$, there exists some $q' \in \mathbf{P}$ such that $p' \xrightarrow{a} q'$ and $(q, q') \in S$, and
 3. for all $a \in \mathbf{A}$ and $q' \in \mathbf{P}$ such that $p' \xrightarrow{a} q'$, there exists some $q \in \mathbf{P}$, such that $p \xrightarrow{a} q$ and $(q, q') \in S$.
- $\mathcal{F}_{convex}(S) = \mathcal{F}_{lower}(S) \cap \mathcal{F}_{upper}(S)$

A *lower simulation* is a post-fixed point for \mathcal{F}_{lower} , and we likewise define *inclusion simulation*, *upper simulation*, *smash simulation* and *convex simulation* (also known as a *prebisimulation* or *partial bisimulation*) [37, 40, 42, 46, 52, 56].

For each of these functions, we can find an endodeclaration that expresses it in Hennessy-Milner logic with divergence. For example, \mathcal{F}_{lower} is expressed—just like \mathcal{F}_{sim} in Section 3.1.1—by

$$E_{lower} : p \mapsto \bigwedge_{a \in \mathbf{A}} \bigwedge_{q \in \mathbf{P}. p \xrightarrow{a} q} \langle a \rangle X_q$$

and \mathcal{F}_{upper} is expressed by

$$\begin{aligned} E_{upper} : p &\mapsto \not\uparrow \wedge \bigwedge_{a \in \mathbf{A}} [a] \vee_{q \in \mathbf{P}. p \xrightarrow{a} q} X_q && (p \not\uparrow) \\ &\mapsto \text{tt} && (p \uparrow). \end{aligned}$$

Supposing that **A** is nonempty, we can equivalently write this endodeclaration as follows

$$\begin{aligned} E_{upper} : p &\mapsto \bigwedge_{a \in \mathbf{A}} [a] \wedge \not\uparrow \vee_{q \in \mathbf{P}. p \xrightarrow{a} q} X_q && (p \not\uparrow) \\ &\mapsto \text{tt} && (p \uparrow). \end{aligned}$$

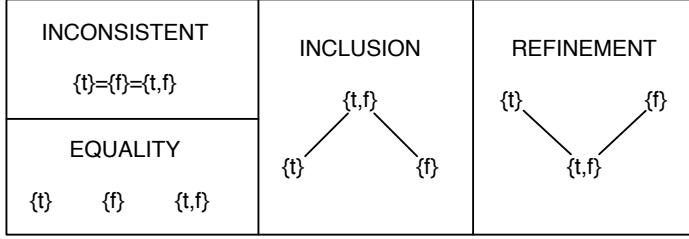


Figure 1: All the NDBPs for no errors

Lemma 3.6 then tells us that \mathcal{F}_{convex} is expressed by

$$\begin{aligned}
 E_{convex} : p &\mapsto \bigwedge_{a \in \mathbf{A}} \bigwedge_{q \in \mathbf{P}. p \xrightarrow{a} q} \langle a \rangle X_q \\
 &\quad \wedge \bigwedge_{a \in \mathbf{A}} [a]^{\wedge \uparrow} \bigvee_{q \in \mathbf{P}. p \xrightarrow{a} q} X_q \quad (p \uparrow \uparrow) \\
 &\mapsto \bigwedge_{a \in \mathbf{A}} \bigwedge_{q \in \mathbf{P}. p \xrightarrow{a} q} \langle a \rangle X_q \quad (p \uparrow)
 \end{aligned}$$

We can treat these examples, and others such as those in [1], in a systematic manner by using the following concepts, taken from [38].

Notation for Disjoint Union Let $\{A_i\}_{i \in I}$ be a family of sets. Then we write the disjoint union as follows:

$$\sum_{i \in I} A_i = \{(i, a) \mid i \in I, a \in A_i\}$$

For the binary case, let A and B be sets. Then we write

$$A + B = \{inl\ a \mid a \in A\} \cup \{inr\ b \mid b \in B\}$$

where we define $inl\ a = (1, a)$ and $inr\ b = (2, b)$.

Definition 3.14 Let E be a finite set. We write $\mathbb{B} = \{\text{t}, \text{f}\}$ for the set of booleans.

1. We define the conditional on $\mathcal{P}(\mathbb{B} + E)$ to be the following ternary operation:

$$\begin{aligned}
 \mathcal{P}(\mathbb{B} + E) \times (\mathcal{P}(\mathbb{B} + E))^{\mathbb{B}} &\longrightarrow \mathcal{P}(\mathbb{B} + E) \\
 K + D, \quad f &\mapsto \bigcup_{b \in K} f(b) \cup \{inr\ e \mid e \in D\}
 \end{aligned}$$

2. A nondeterministic boolean precongruence (NDBP) for E -errors is a preorder on $\{U \subseteq \mathbb{B} + E \mid U \neq \emptyset\}$ making the conditional operation monotone.

It is shown in [38] that, if \sqsubseteq is a NDBP for E -errors, then

- \cup is monotone with respect to \sqsubseteq , and
- \sqsubseteq has a unique extension to $\mathcal{P}(\mathbb{B} + E)$ making the conditional operation monotone.

Some examples of NDBPs are displayed in Figure 1–3.

INCONSISTENT	EQUALITY						
$\{t\}=\{f\}=\{t,f\}=\{d\}=\{t,d\}=\{f,d\}=\{t,f,d\}$	$\{t\}$	$\{f\}$	$\{t,f\}$	$\{d\}$	$\{t,d\}$	$\{f,d\}$	$\{t,f,d\}$
<p>LOWER</p>	<p>UPPER</p>		<p>SMASH</p>				
<p>OP-LOWER</p>	<p>OP-UPPER</p>		<p>OP-SMASH</p>				
<p>CONVEX</p>	<p>INCLUSION</p>		<p>SESQUI INCLUSION</p>				
<p>OP-CONVEX</p>	<p>REFINEMENT</p>		<p>SESQUI REFINEMENT</p>				
<p>PLUCKED</p>	<p>STUNTED</p>		<p>LOWER CONGRUENCE</p>				
<p>OP-PLUCKED</p>	<p>OP-STUNTED</p>		<p>UPPER CONGRUENCE</p>				

Figure 2: All the NDBPs for divergence (d)

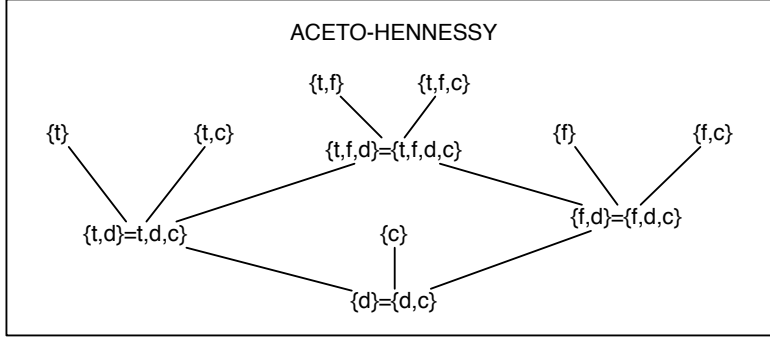


Figure 3: NDBP for divergence (d) and deadlock (c) used in [1]

Any NDBP gives rise to a notion of simulation. Let $P = (\mathbf{P}, \longrightarrow, \not\rightarrow)$ be an \mathbf{A} -labelled transition system with \mathbf{E} -errors.

Definition 3.15 Let E be a finite set, and let \sqsubseteq be a NDBP for \mathbf{E} -errors. For any relation $S \subseteq \mathbf{P} \times \mathbf{P}$, we define $(p, p') \in \mathcal{F}_{\sqsubseteq}(S)$ when

- $\{t\} + \text{Errors}(p) \sqsubseteq \{t\} + \text{Errors}(p')$
- if $\{t, f\} + \text{Errors}(p) \not\sqsubseteq \{t\} + \text{Errors}(p')$ then for every $a \in \mathbf{A}$ and $q \in \mathbf{P}$ such that $p \xrightarrow{a} q$, there exists some $q' \in \mathbf{P}$ such that $p' \xrightarrow{a} q'$ and $(q, q') \in S$
- if $\{t\} + \text{Errors}(p) \not\sqsubseteq \{t, f\} + \text{Errors}(p')$ then for every $a \in \mathbf{A}$ and $q' \in \mathbf{P}$ such that $p' \xrightarrow{a} q'$, there exists some $q \in \mathbf{P}$ such that $p \xrightarrow{a} q$ and $(q, q') \in S$.

A post-fixed point of $\mathcal{F}_{\sqsubseteq}$ is called a \sqsubseteq simulation, and the greatest one is called \sqsubseteq similarity.

Theorem 3.16 Let E be a finite set and let \sqsubseteq be a NDBP for \mathbf{E} -errors. The \mathbf{P} -indexed endodeclaration

$$\begin{aligned}
 E_{\sqsubseteq} : p \mapsto & \bigvee \{C \subseteq E \mid \{t\} + \text{Errors}(p) \sqsubseteq \{t\} + C\} \\
 & \wedge \bigwedge_{a \in \mathbf{A}} \bigwedge_{q \in \mathbf{P}. p \xrightarrow{a} q} \langle a \rangle_{\{C \subseteq E \mid \{t, f\} + \text{Errors}(p) \sqsubseteq \{t\} + C\}}^{\{C \subseteq E \mid \{t, f\} + \text{Errors}(p) \sqsubseteq \{t\} + C\}} X_q \\
 & \wedge \bigwedge_{a \in \mathbf{A}} [a]_{\{C \subseteq E \mid \{t\} + \text{Errors}(p) \sqsubseteq \{t, f\} + C\}}^{\{C \subseteq E \mid \{t\} + \text{Errors}(p) \not\sqsubseteq \{t, f\} + C\}} \bigvee_{q \in \mathbf{P}. p \xrightarrow{a} q} X_q
 \end{aligned}$$

expresses $\mathcal{F}_{\sqsubseteq}$.

Proof The proof is straightforward. □

It follows that E_{\sqsubseteq} characterizes \sqsubseteq similarity.

Remark Although there are various equivalent ways of presenting E_{\sqsubseteq} , we have chosen a formulation that uses only the *positive modalities* of \sqsubseteq , in the sense of [38].

4 Extensions of the approach

All the applications of the general theory developed so far in the paper dealt with relations that were defined as greatest fixed points of monotone endofunctions over the complete lattice of binary relations over \mathbf{P} . In this setting, Theorem 2.16 allowed us to characterize those relations logically by exhibiting an endodeclaration that expressed the relevant endofunction. Sometimes, however, Theorem 2.16 is not readily applicable in order to yield characteristic-formula constructions for behavioral relations. In this section, we present two such examples of relations, namely mutual similarity (i.e., simulation equivalence) and the 2-nested simulation preorder. We also extend our approach, providing the theoretical tools needed to offer characteristic formulae for them. The notion of formula with endodeclaration, which we now proceed to define, provides the underlying framework for the subsequent developments in the section.

4.1 Formula with endodeclaration

Definition 4.1 1. A formula with endodeclaration is a pair (F, E) , where $F \in \mathcal{L}(I)$ is a formula and $E \in \mathcal{L}(I)^I$ is an endodeclaration, for some set I .

2. A formula with endodeclaration (F, E) characterizes process $p \in \mathbf{P}$ up to a relation $S \subseteq \mathbf{P} \times \mathbf{P}$ if

$$(p, q) \in S \text{ iff } \nu \llbracket E \rrbracket, q \models \llbracket F \rrbracket$$

or equivalently, if

$$(p, q) \in S \text{ iff } q \in \llbracket F \rrbracket(\nu \llbracket E \rrbracket).$$

Note that, for a declaration E , the following are equivalent:

- E characterizes S
- for each $p \in \mathbf{P}$, the formula with endodeclaration (X_p, E) characterizes p up to S .

4.1.1 Substitution

We describe the notion and properties of substitution, which will be used in Section 4.2 and extensively in Section 6.

For any formula $F \in \mathcal{L}(I)$ and declaration $D : I \rightarrow \mathcal{L}(J)$, we write $F[D] \in \mathcal{L}(J)$ for the *substitution* of D within F , defined by induction on F as follows:

$$\begin{aligned} X_{\hat{i}}[D] &= D(\hat{i}) \\ (\bigwedge_{k \in K} F_k)[D] &= \bigwedge_{k \in K} (F_k[D]) \\ (\bigvee_{k \in K} F_k)[D] &= \bigvee_{k \in K} (F_k[D]) \\ \langle a \rangle F [D] &= \langle a \rangle (F[D]) \\ [a] F [D] &= [a](F[D]). \end{aligned}$$

Likewise, given declarations $D : I \rightarrow \mathcal{L}(J)$ and $D' : J \rightarrow \mathcal{L}(H)$ we define the substitution $D[D'] : I \rightarrow \mathcal{L}(H)$ to be the declaration $i \mapsto D(i)[D']$.

Lemma 4.2 *Substitution satisfies the following properties.*

1. For any formula $F \in \mathcal{L}(I)$, we have $F[i \mapsto X_i] = F$.
2. For any formula $F \in \mathcal{L}(I)$ and declarations $D : I \rightarrow \mathcal{L}(J)$ and $D' : J \rightarrow \mathcal{L}(H)$, we have $(F[D])[D'] = F[D[D']]$.
3. For any formula $F \in \mathcal{L}(I)$ and declaration $D : I \rightarrow \mathcal{L}(J)$, the following diagram commutes:

$$\begin{array}{ccc}
 \mathcal{P}(\mathbf{P})^J & \xrightarrow{\llbracket D \rrbracket} & \mathcal{P}(\mathbf{P})^I \\
 & \searrow \llbracket F[D] \rrbracket & \downarrow \llbracket F \rrbracket \\
 & & \mathcal{P}(\mathbf{P})
 \end{array}$$

Proof All by induction on F . □

4.2 Adding variables

It is useful to note that a characteristic formula remains so when we add variables to the language and extend the declaration in an arbitrary manner. Lemma 4.4 below will be used in the proof of this fact, as well as in the proof of Theorem 4.10 in Section 4.4 to follow.

Definition 4.3 Let A , B , and C be sets, $F : A \times B \rightarrow C$, and $a \in A$. Then we write $F_a : B \rightarrow C$ for the function mapping b to $F(a, b)$.

It is clear that if A , B and C are posets and $F : A \times B \rightarrow C$ is monotone, then F_a is also monotone for each $a \in A$.

Lemma 4.4 Let A and B be posets, and let $F : A \rightarrow A$ and $G : A \times B \rightarrow B$ be monotone functions. Suppose νF exists. Define $H : A \times B \rightarrow A \times B$ by

$$(a, b) \mapsto (F(a), G(a, b))$$

Then νH exists iff $\nu G_{\nu F}$ does, in which case

$$\nu H = (\nu F, \nu G_{\nu F})$$

Proof Suppose νH exists, and put $(\hat{a}, \hat{b}) = \nu H$. Then

$$\begin{aligned}
 (\hat{a}, \hat{b}) &= H(\hat{a}, \hat{b}) \\
 &= (F(\hat{a}), G(\hat{a}, \hat{b})).
 \end{aligned}$$

So \hat{a} is a fixed point of F , giving $\hat{a} \sqsubseteq \nu F$, and

$$\begin{aligned}
 \hat{b} &= G(\hat{a}, \hat{b}) \\
 &\sqsubseteq G(\nu F, \hat{b}) \\
 &= G_{\nu F}(\hat{b}),
 \end{aligned}$$

yielding that \hat{b} is a post-fixed point of $G_{\nu F}$. If b is a post-fixed point of $G_{\nu F}$ then

$$\begin{aligned}
 (\nu F, b) &\sqsubseteq (F(\nu F), G_{\nu F}(b)) \\
 &= (F(\nu F), G(\nu F, b)) \\
 &= H(\nu F, b).
 \end{aligned}$$

So $(\nu F, b)$ is a post-fixed point of H , giving $b \sqsubseteq \hat{b}$ because $(\hat{a}, \hat{b}) = \nu H$ is the greatest post-fixed point of H . Thus \hat{b} is the greatest post-fixed point of $G_{\nu F}$.

Conversely, suppose $\nu G_{\nu F}$ exists. Then

$$\begin{aligned} H(\nu F, \nu(G_{\nu F})) &= (F(\nu F), G(\nu F, \nu G_{\nu F})) \\ &= (F(\nu F), G_{\nu F}(\nu G_{\nu F})) \\ &= (\nu F, \nu(G_{\nu F})), \end{aligned}$$

so $(\nu F, \nu G_{\nu F})$ is a fixed point of H . If (a, b) is a post-fixed point of H , then

$$\begin{aligned} (a, b) &\sqsubseteq H(a, b) \\ &= (F(a), G(a, b)) \end{aligned}$$

so a is a post-fixed point of f giving $a \sqsubseteq \nu f$, and

$$\begin{aligned} b &\sqsubseteq G(a, b) \\ &\sqsubseteq G(\nu F, b) \\ &= G_{\nu F}(b) \end{aligned}$$

so b is a post-fixed point of $G_{\nu F}$ giving $b \sqsubseteq \nu G_{\nu F}$. Thus $(\nu f, \nu G_{\nu F})$ is a greatest post-fixed point of H , which was to be shown. \square

Remark We apply Lemma 4.4—as well as Lemma 2.2 and Theorem 2.3—only in the setting of complete lattices, where νf and μf exist automatically. We have presented these statements in the general setting of posets, to illustrate the fact that they are ‘elementary’ results that do not depend on induction. In the complete lattice setting, Lemma 4.4 is a consequence of a classic result about fixed points due to Bekic. (See, for instance, page 130, equation [g], in [7] for a formulation in terms of iteration theories.)

We now proceed to define formally the extension of a declaration.

Definition 4.5 *Let I and J be sets, and $m : I \rightarrow J$ an injection. Let E be an I -indexed endodeclaration and $D : (J \setminus \text{range}(m)) \rightarrow \mathcal{L}(J)$ a declaration. The extension of E by D , written $E_m D$, is the J -indexed endodeclaration*

$$\begin{aligned} m(i) &\mapsto E(i)[i \mapsto X_{m(i)}] & (i \in I) \\ j &\mapsto D(j) & (j \in J \setminus \text{range}(m)) \end{aligned}$$

Lemma 4.6 *Let (F, E) be a formula with endodeclaration. Let $m : I \rightarrow J$ an injection and $D : (J \setminus \text{range}(m)) \rightarrow \mathcal{L}(J)$ a declaration. For any relation $S \subseteq \mathbf{P} \times \mathbf{P}$ and process $p \in \mathbf{P}$, the following statements are equivalent:*

- (F, E) characterizes p up to S .
- $(F[i \mapsto X_{m(i)}], E_m D)$ characterizes p up to S .

Proof We abbreviate $I' = J \setminus \text{range}(m)$. Let $\theta : \mathcal{P}(\mathbf{P})^I \times \mathcal{P}(\mathbf{P})^{I'} \rightarrow \mathcal{P}(\mathbf{P})^J$ be the isomorphism mapping (σ, σ') to

$$\begin{aligned} m(i) &\mapsto \sigma(i) & (i \in I) \\ j &\mapsto \sigma'(j) & (j \in I') \end{aligned}$$

and define the endofunction $G_{\mathbf{v}[[E]]}$ on $\mathcal{P}(\mathbf{P})^{I'}$ to map σ to $[[D]]\theta(\mathbf{v}[[E]], \sigma)$, and the endofunction H on $\mathcal{P}(\mathbf{P})^I \times \mathcal{P}(\mathbf{P})^{I'}$ to map (σ, σ') to $([[E]]\sigma, [[D]]\theta(\sigma, \sigma'))$. Since $[[D]]$ is monotone, so is $G_{\mathbf{v}[[E]]}$, and hence $\mathbf{v}G_{\mathbf{v}[[E]]}$ exists. Then by Lemma 4.4, $\mathbf{v}H = (\mathbf{v}[[E]], \mathbf{v}G_{\mathbf{v}[[E]])}$.

By the definitions, the following commutes:

$$\begin{array}{ccc} \mathcal{P}(\mathbf{P})^I \times \mathcal{P}(\mathbf{P})^{I'} & \xrightarrow{H} & \mathcal{P}(\mathbf{P})^I \times \mathcal{P}(\mathbf{P})^{I'} \\ \theta \downarrow & & \downarrow \theta \\ \mathcal{P}(\mathbf{P})^J & \xrightarrow{[[E_m D]]} & \mathcal{P}(\mathbf{P})^J \end{array} \quad (2)$$

as $\theta(H(\sigma, \sigma')) = \theta([[E]]\sigma, [[D]]\theta(\sigma, \sigma'))$ is the function that

$$\begin{array}{ll} m(i) & \mapsto ([[E]]\sigma)(i) \quad (i \in I) \\ j & \mapsto [[D(j)]]\theta(\sigma, \sigma') \quad (j \in I') \end{array} ,$$

and $[[E_m D]]\theta(\sigma, \sigma')$ is the function that

$$\begin{array}{ll} m(i) & \mapsto [[E_m D(m(i))]]\theta(\sigma, \sigma') = \\ & [[E(i)[i \mapsto X_{m(i)}]]]\theta(\sigma, \sigma') = ([[E]]\sigma)(i) \quad (i \in I) \\ j & \mapsto [[E_m D(j)]]\theta(\sigma, \sigma') = [[D(j)]]\theta(\sigma, \sigma') \quad (j \in I') \end{array} .$$

Thus Theorem 2.3(2) gives us $\mathbf{v}[[E_m D]] = \theta(\mathbf{v}[[E]], \mathbf{v}G_{\mathbf{v}[[E]])}$. We calculate

$$\begin{aligned} [[F][i \mapsto X_{m(i)}]](\mathbf{v}[[E_m D]]) &= [[F]](i \mapsto [[X_{m(i)}]](\mathbf{v}[[E_m D]]) \\ &= [[F]](i \mapsto (\mathbf{v}[[E_m D]])m(i)) \\ &= [[F]](i \mapsto (\theta(\mathbf{v}[[E]], \mathbf{v}G_{\mathbf{v}[[E]])})m(i)) \\ &= [[F]](i \mapsto (\mathbf{v}[[E]])i) \\ &= [[F]](\mathbf{v}[[E]]) \end{aligned}$$

giving the required result. \square

To illustrate the use of Lemma 4.6, we use the following notions.

Definition 4.7 Let $p \in \mathbf{P}$ be a process.

1. A subsystem of P is a subset $\mathbf{Q} \subseteq \mathbf{P}$ such that if $p \in \mathbf{Q}$ and $p \xrightarrow{a} p'$ then $p' \in \mathbf{Q}$.
2. We write $\text{reach}(p)$ for the set of processes $q \in \mathbf{P}$ reachable from p , i.e. the least subsystem containing p .
3. p is image finite (resp. image countable) when for each $q \in \mathbf{P}$ and $a \in \mathbf{A}$, the set $\{r \in \mathbf{P} \mid q \xrightarrow{a} r\}$ is finite (resp. countable).

Let $p \in \mathbf{P}$ and let E be a \mathbf{P} -indexed declaration. We write $E \upharpoonright \text{reach}(p)$ for the restriction of E to a $\text{reach}(p)$ -indexed declaration. By way of example, take $E = E_{\text{sim}}$. Then Lemma 4.6 tells us that $(X_p, E_{\text{sim}} \upharpoonright \text{reach}(p))$ is characteristic for p up to $\underline{\sqsubseteq}_{\text{sim}}$. To see why this is advantageous, suppose that p is image countable. Then, since \mathbf{A} is assumed countable, $\text{reach}(p)$ must be countable, and each formula in the declaration $E_{\text{sim}} \upharpoonright \text{reach}(p)$ uses only conjunctions and disjunctions of countable arity. This will be the case even if \mathbf{P} contains other processes that are not image countable.

4.3 Mutual simulation

We now proceed to develop the general theory that will allow us to provide characteristic-formula constructions for relations that, like mutual similarity, are obtained as intersections of relations for which we already have characteristic formulae with declarations.

If we know how to find characteristic formulae up to S_1 and up to S_2 , then we can find them up to $S_1 \cap S_2$. This generalizes to arbitrary intersections, in the following manner.

Lemma 4.8 *Let $\{S_j\}_{j \in J}$ be a family of binary relations from \mathbf{P} to \mathbf{P} , and let $p \in \mathbf{P}$. For each $j \in J$, let E_j be an I_j -indexed endodeclaration, and let F_j be such that (F_j, E_j) characterizes p up to S_j . Let E be the $\sum_{j \in J} I_j$ -indexed endodeclaration given by*

$$(j, i) \mapsto E_j(i)[i \mapsto X_{(j,i)}]$$

Then

$$\left(\bigwedge_{j \in J} F_j[i \mapsto X_{(j,i)}], E \right)$$

characterizes p up to $\bigcap_{j \in J} S_j$.

Proof For each $\hat{j} \in J$, Lemma 4.6 tells us that $(F_{\hat{j}}[i \mapsto X_{(j,i)}], E)$ characterizes p up to $S_{\hat{j}}$. The result follows. \square

We write \sim_{sim} for *mutual similarity*, i.e. the intersection of \sqsubseteq_{sim} and \sqsubseteq_{opsim} . This can be treated using Lemma 4.8. We define E_{simeq} to be the $\mathbf{P} + \mathbf{P}$ indexed declaration

$$\begin{aligned} inl\ p &\mapsto \bigwedge_{a \in \mathbf{A}} \bigwedge_{q \in \mathbf{P}. p \xrightarrow{a} q} \langle a \rangle X_{inl\ q} \\ inr\ p &\mapsto \bigwedge_{a \in \mathbf{A}} [a] \bigvee_{q. p \xrightarrow{a} q} X_{inr\ q} \end{aligned}$$

Then the formula with endodeclaration $(X_{inl\ p} \wedge X_{inr\ p}, E_{simeq})$ characterizes p up to \sim_{sim} .

4.4 Nested simulation [26]

Let $P = (\mathbf{P}, \longrightarrow)$ be an \mathbf{A} -labelled transition system. A relation $S \subseteq (\mathbf{P} \times \mathbf{P})$ is a *2-nested simulation* when it is a simulation contained in \sqsubseteq_{opsim} . The greatest 2-nested simulation is called *2-nested similarity* and denoted by \sqsubseteq_{2sim} . Theorem 2.16 is not applicable in this case, so we need to develop a suitable generalization.

Definition 4.9 *Let A be a complete lattice, let f be a monotone endofunction on A and let $a \in A$. We write $a \sqcap f$ for the endofunction $x \mapsto a \wedge f(x)$ on A .*

Theorem 4.10 *Let $P = (\mathbf{P}, \longrightarrow)$ be an \mathbf{A} -labelled transition systems, let \mathcal{F} be a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$, and let $R \subseteq \mathbf{P} \times \mathbf{P}$ be a relation. Let E be an I -indexed endodeclaration and $D : \mathbf{P} \rightarrow \mathcal{L}(I)$ a declaration such that, for each $p \in \mathbf{P}$, the formula with endodeclaration $(D(p), E)$ characterizes p up to R . Let E' be a \mathbf{P} -indexed endodeclaration expressing \mathcal{F} . Let E'' be the $I + \mathbf{P}$ indexed declaration*

$$\begin{aligned} inl\ i &\mapsto E(i)[i' \mapsto X_{inl\ i'}] \\ inr\ p &\mapsto F(p)[i' \mapsto X_{inl\ i'}] \wedge E'(p)[p' \mapsto X_{inr\ p'}] \end{aligned}$$

Let $p \in \mathbf{P}$. Then (X_p, E'') characterizes p up to $\mathbf{v}(R \sqcap \mathcal{F})$.

Proof Let $h : \mathcal{P}(\mathbf{P})^I \rightarrow \mathcal{P}(\mathbf{P} \times \mathbf{P})$ be the monotone function mapping each σ to

$$\{(q, q') \in \mathbf{P} \times \mathbf{P} \mid q' \in \llbracket D(q) \rrbracket \sigma\}.$$

Let H be the monotone endofunction on $\mathcal{P}(\mathbf{P})^I \times \mathcal{P}(\mathbf{P} \times \mathbf{P})$ mapping (σ, S) to $(\llbracket E \rrbracket \sigma, h(\sigma) \cap \mathcal{F}(S))$. Lemma 4.4 gives us

$$\mathbf{v}H = (\mathbf{v}\llbracket E \rrbracket, \mathbf{v}(h(\mathbf{v}\llbracket E \rrbracket) \cap \mathcal{F}))$$

Now

$$\begin{aligned} (p, p') \in h(\mathbf{v}\llbracket E \rrbracket) &\Leftrightarrow p' \in \llbracket D(p) \rrbracket (\mathbf{v}\llbracket E \rrbracket) && \text{by definition} \\ &\Leftrightarrow (p, p') \in R && \text{since } (D(p), E) \text{ characterizes } p \text{ up to } R \end{aligned}$$

Therefore $h(\mathbf{v}\llbracket E \rrbracket) = R$ and so $\mathbf{v}H = (\mathbf{v}\llbracket E \rrbracket, \mathbf{v}(R \cap \mathcal{F}))$. By calculation, the following diagram commutes:

$$\begin{array}{ccc} \mathcal{P}(\mathbf{P})^I \times \mathcal{P}(\mathbf{P} \times \mathbf{P}) & \xrightarrow{\tau} & \mathcal{P}(\mathbf{P})^{I+P} \\ H \downarrow & & \downarrow \llbracket E'' \rrbracket \\ \mathcal{P}(\mathbf{P})^I \times \mathcal{P}(\mathbf{P} \times \mathbf{P}) & \xrightarrow{\tau} & \mathcal{P}(\mathbf{P})^{I+P} \end{array}$$

where τ is the isomorphism mapping (σ, S) to the environment

$$\begin{aligned} \text{inl } i &\mapsto \sigma i \\ \text{inr } p &\mapsto \{p' \in \mathbf{P} \mid (p, p') \in S\} \end{aligned}$$

So Lemma 2.3(2) gives $\tau \mathbf{v}H = \mathbf{v}\llbracket E'' \rrbracket$. Hence

$$\begin{aligned} \llbracket X_{\text{inr } p} \rrbracket (\mathbf{v}\llbracket E'' \rrbracket) &= (\mathbf{v}\llbracket E'' \rrbracket)(\text{inr } p) \\ &= (\tau \mathbf{v}H)(\text{inr } p) \\ &= (\tau(\mathbf{v}\llbracket E \rrbracket, \mathbf{v}(R \cap \mathcal{F}))) (\text{inr } p) \\ &= \{p' \in \mathbf{P} \mid (p, p') \in \mathbf{v}(R \cap \mathcal{F})\} \end{aligned}$$

as required. □

It is easy to see that Theorem 2.16 corresponds to the special case of Theorem 4.10 given by

$$\begin{aligned} R &= \mathbf{P} \times \mathbf{P} \\ I &= \emptyset \\ F &: p \mapsto tt \end{aligned}$$

We are now able to put together the characteristic formulae for $\sqsubseteq_{\text{opsim}}$ and the declaration expressing \mathcal{F}_{sim} , both given in Section 3.1.1. Let $E_{2\text{sim}}$ be the $\mathbf{P} + \mathbf{P}$ indexed declaration

$$\begin{aligned} \text{inl } p &\mapsto \bigwedge_{q \in \mathbf{A}} [a] \bigvee_{q \in \mathbf{P}. p \xrightarrow{a} q} X_{\text{inl } q} \\ \text{inr } p &\mapsto X_{\text{inl } p} \wedge \bigwedge_{a \in \mathbf{A}} \bigwedge_{q \in \mathbf{P}. p \xrightarrow{a} q} \langle a \rangle X_{\text{inr } q}. \end{aligned}$$

For any $p \in \mathbf{P}$, Theorem 4.10 tells us that $(X_{\text{inr } p}, E_{2\text{sim}})$ characterizes p up to $\sqsubseteq_{2\text{sim}}$.

5 Co-characteristic formulae

5.1 Dualization

For any formula $F \in \mathcal{L}(I)$ we write $\overline{F} \in \mathcal{L}(I)$ for the *de Morgan dual* of F , defined by induction on F as follows.

$$\begin{aligned} \overline{X_i} &= X_i \\ \overline{\bigwedge_{k \in K} F_k} &= \bigvee_{k \in K} \overline{F_k} \\ \overline{\bigvee_{k \in K} F_k} &= \bigwedge_{k \in K} \overline{F_k} \\ \overline{\langle a \rangle F} &= [a] \overline{F} \\ \overline{[a] f} &= \langle a \rangle \overline{f} \end{aligned}$$

Likewise, for a declaration $D : I \rightarrow \mathcal{L}(J)$ we define the dual $\overline{D} : I \rightarrow \mathcal{L}(J)$ to be the declaration $i \mapsto \overline{D(i)}$.

Before giving the properties of dualization, it will be helpful to introduce the following notion.

Definition 5.1 For posets A and B , an anti-isomorphism $\psi : A \rightarrow B$ is a bijective function such that $x \sqsubseteq_A y$ iff $\psi(x) \sqsupseteq_B \psi(y)$.

Theorem 5.2 Let A and B be posets, f and g be monotone endofunctions on A and B respectively and $\psi : A \rightarrow B$ an anti-isomorphism such that $\psi \circ f = g \circ \psi$, or equivalently such that the square

$$\begin{array}{ccc} A & \xrightarrow{f} & A \\ \psi \downarrow & & \downarrow \psi \\ B & \xrightarrow{g} & B \end{array}$$

commutes. Then the following statements hold:

1. If $f(x) = x$ then $g(\psi(x)) = \psi(x)$ (ψ maps fixed points for f into fixed points for g).
2. If νf exists then μg exists and $\psi(\nu f) = \mu g$ (ψ maps the greatest fixed point for f into the least fixed point for g).
3. If μf exists then νg exists and $\psi(\mu f) = \nu g$ (ψ maps the least fixed point for f into the greatest fixed point for g).

Proof This theorem is equivalent to Theorem 2.3, because an anti-isomorphism from A to B is just an isomorphism from A to the dual of B . \square

Lemma 5.3 Dualization satisfies the following properties.

1. For any formula $F \in \mathcal{L}(I)$, we have $\overline{\overline{F}} = F$.
2. Let us write $\chi : \mathcal{P}(\mathbf{P}) \rightarrow \mathcal{P}(\mathbf{P})$ for the anti-isomorphism mapping U to $\mathbf{P} \setminus U$, and $\chi' : \mathcal{P}(\mathbf{P})^I \rightarrow \mathcal{P}(\mathbf{P})^I$ for the anti-isomorphism mapping σ to $i \mapsto (\mathbf{P} \setminus \sigma(i))$. For any formula $F \in \mathcal{L}(I)$, the following diagram commutes:

$$\begin{array}{ccc} \mathcal{P}(\mathbf{P})^I & \xrightarrow{[F]} & \mathcal{P}(\mathbf{P}) \\ \chi' \downarrow & & \downarrow \chi \\ \mathcal{P}(\mathbf{P})^I & \xrightarrow{[F]} & \mathcal{P}(\mathbf{P}) \end{array}$$

Proof Both statements can be shown by induction on F . \square

5.2 Co-characteristic formulae: theory and examples

We now introduce the notion of a *co-characteristic formula*, which for each $p \in \mathbf{P}$ is a formula F_p such that for some fixed variable interpretation σ and each $q \in \mathbf{P}$,

$$\sigma, q \models F_p \text{ iff } (q, p) \notin S.$$

Definition 5.4 Let $S \subseteq \mathbf{P} \times \mathbf{P}$ be a relation.

1. An endodeclaration E co-characterizes S iff

$$(q, p) \notin S \text{ iff } q \in (\mu[[E]])(p).$$

2. A formula with endodeclaration (F, E) co-characterizes process $p \in \mathbf{P}$ up to S if

$$(q, p) \notin S \text{ iff } \mu[[E]], p \not\models [[F]]$$

or equivalently if

$$(q, p) \notin S \text{ iff } p \in [[F]](\mu[[E]]).$$

Note that, for a declaration E , the following are equivalent:

- E co-characterizes S
- for each $p \in \mathbf{P}$, the formula with endodeclaration (X_p, E) co-characterizes p up to S .

To find a co-characteristic formula up to a given a binary relation S , we simply take a characteristic formula up to S^{-1} and dualize it.

Lemma 5.5 Let $S \subseteq \mathbf{P} \times \mathbf{P}$ be a relation.

1. Let E be an endodeclaration for $\mathcal{L}(\mathbf{P})$. Then the following statements are equivalent:

- E characterizes S^{-1} , and
- \bar{E} co-characterizes S .

2. Let $p \in \mathbf{P}$, and let (F, E) be a formula with endodeclaration. Then the following statements are equivalent:

- (F, E) characterizes p up to S^{-1} , and
- (\bar{F}, \bar{E}) is co-characterizes p up to S .

Proof We just prove part (2), as part (1) is a special case. Lemma 5.3(2) tells us that

$$\begin{array}{ccc} \mathcal{P}(\mathbf{P})^I & \xrightarrow{[[E]]} & \mathcal{P}(\mathbf{P})^I \\ \chi^I \downarrow & & \downarrow \chi^I \\ \mathcal{P}(\mathbf{P})^I & \xrightarrow{[[\bar{E}]]} & \mathcal{P}(\mathbf{P})^I \end{array}$$

commutes, and Theorem 5.2(2) tells us that $\mu[[\bar{E}]] = \chi^I(v[[E]])$. We apply Lemma 5.3(2) again to obtain

$$\begin{aligned} [[\bar{F}]](\mu[[\bar{E}]]) &= \chi^I([[F]](v[[E]])) \\ &= \mathbf{P} \setminus [[F]](v[[E]]). \end{aligned}$$

The result follows. □

Here are some examples.

1. We obtain co-characteristic formulae for \sqsubseteq_{sim} by dualizing the characteristic formulae for \sqsubseteq_{opsim} . Thus the endodeclaration E'_{sim} , where

$$E'_{sim} : p \mapsto \bigvee_{a \in \mathbf{A}} \langle a \rangle \bigwedge_{q \in \mathbf{P}. p \xrightarrow{a} q} X_q$$

co-characterizes \sqsubseteq_{sim} .

2. We obtain co-characteristic formulae for \sqsubseteq_{opsim} by dualizing the characteristic formulae for \sqsubseteq_{sim} . Thus the endodeclaration E'_{opsim} , where

$$E'_{opsim} : p \mapsto \bigvee_{a \in \mathbf{A}} \bigvee_{q \in \mathbf{P}. p \xrightarrow{a} q} [a]X_q$$

co-characterizes \sqsubseteq_{opsim} .

3. We obtain co-characteristic formulae for \sim_{bisim} by dualizing its characteristic formulae. Thus E'_{bisim} , where

$$E'_{bisim} : p \mapsto \left(\bigvee_{a \in \mathbf{A}} \langle a \rangle \bigwedge_{q \in \mathbf{P}. p \xrightarrow{a} q} X_q \right) \vee \left(\bigvee_{a \in \mathbf{A}} \bigvee_{q \in \mathbf{P}. p \xrightarrow{a} q} [a]X_q \right)$$

co-characterizes \sim_{bisim} .

4. Let \mathbf{E} be a finite set and let \sqsubseteq be a NDBP for \mathbf{E} -errors. We obtain co-characteristic formulae for \sqsubseteq similarity by dualizing the characteristic formulae for its inverse (see Theorem 3.16). The \mathbf{P} -indexed endodeclaration

$$\begin{aligned} E'_{\sqsubseteq} : q \mapsto & \nabla \{C \sqsubseteq E | \{t\} + C \sqsubseteq \{t\} + \text{Errors}(q)\} \\ & \vee \bigvee_{a \in \mathbf{A}} \langle a \rangle \bigwedge_{q' \in \mathbf{P}. q \xrightarrow{a} q'} \bigvee_{\{C \sqsubseteq E | \{t, f\} + C \sqsubseteq \{t\} + \text{Errors}(q)\}} \bigwedge_{\{C \sqsubseteq E | \{t\} + C \sqsubseteq \{t, f\} + \text{Errors}(q)\}} X_{q'} \\ & \vee \bigvee_{a \in \mathbf{A}} \bigvee_{q' \in \mathbf{P}. q \xrightarrow{a} q'} [a] \bigvee_{\{C \sqsubseteq E | \{t\} + C \sqsubseteq \{t, f\} + \text{Errors}(q)\}} \bigwedge_{\{C \sqsubseteq E | \{t, f\} + C \sqsubseteq \{t\} + \text{Errors}(q)\}} X_{q'} \end{aligned}$$

co-characterizes \sqsubseteq_{sim} .

Note that

- for \sqsubseteq_{sim} , both the characteristic and the co-characteristic formulae use only the $\langle a \rangle$ modalities.
- for \sqsubseteq_{opsim} , both the characteristic and the co-characteristic formulae use only the $[a]$ modalities.
- the characteristic and co-characteristic formulae for \sim_{bisim} use both $\langle a \rangle$ and $[a]$ modalities.

For each of our results about characteristic formulae, there is a dual result about co-characteristic formulae. As an illustration, we give the dual of the key results in Section 2.3.

Definition 5.6 (dual of Definition 2.15) *Let \mathcal{F} be a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$. A \mathbf{P} -indexed endodeclaration E is said to co-express \mathcal{F} when*

$$(q, p) \notin \mathcal{F}(S) \Leftrightarrow \sigma_{(\mathbf{P} \times \mathbf{P}) \setminus S^{-1}, q} \models E(p)$$

for every relation $S \subseteq \mathbf{P} \times \mathbf{P}$ and every $p, p' \in \mathbf{P}$.

Lemma 5.7 Write $\varphi' : \mathcal{P}(\mathbf{P} \times \mathbf{P}) \rightarrow \mathcal{P}(\mathbf{P})^{\mathbf{P}}$ for the anti-isomorphism mapping a relation S to the environment $\sigma_{(\mathbf{P} \times \mathbf{P}) \setminus S^{-1}}$. Let \mathcal{F} be a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$, and let E be a \mathbf{P} -indexed endodeclaration. Then E co-expresses \mathcal{F} iff the following commutes:

$$\begin{array}{ccc} \mathcal{P}(\mathbf{P} \times \mathbf{P}) & \xrightarrow{\varphi'} & \mathcal{P}(\mathbf{P})^{\mathbf{P}} \\ \mathcal{F} \downarrow & & \downarrow \llbracket E \rrbracket \\ \mathcal{P}(\mathbf{P} \times \mathbf{P}) & \xrightarrow{\varphi'} & \mathcal{P}(\mathbf{P})^{\mathbf{P}} \end{array} \quad (3)$$

Theorem 5.8 (dual of Theorem 2.16) Let \mathcal{F} be a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$, and let E be a \mathbf{P} -indexed endodeclaration co-expressing \mathcal{F} . Then E co-characterizes $\vee \mathcal{F}$.

As defined in Lemma 2.7, given an endofunction \mathcal{F} on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$, let $\tilde{\mathcal{F}} : S \mapsto (\mathcal{F}(S^{-1}))^{-1}$. To obtain a declaration expressing \mathcal{F} , we take a declaration co-expressing $\tilde{\mathcal{F}}$ and dualize it.

Lemma 5.9 Let \mathcal{F} be a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$. Then the following are equivalent:

- E expresses $\tilde{\mathcal{F}}$.
- \bar{E} co-expresses \mathcal{F} .

In particular,

- the endodeclaration E'_{sim} co-expresses \mathcal{F}_{sim} ,
- the endodeclaration E'_{opsim} co-expresses $\widetilde{\mathcal{F}_{sim}}$, and
- the endodeclaration E'_{bisim} co-expresses \mathcal{F}_{bisim} .
- the endodeclaration E'_{\sqsubseteq} co-expresses $\mathcal{F}_{\sqsubseteq}$.

6 Closed Formulae

In this section, we consider $\mathcal{L}(I, \mathbf{A})$ with \mathbf{A} countable, and with disjunctions and conjunctions over arbitrary cardinals, interpreted over \mathbf{A} -labelled transition systems with sets of processes of arbitrary cardinality. As previously mentioned, all the definitions and results in Sections 2.2 through 4.4 extend easily to this language.

A formula $F \in \mathcal{L}(\emptyset)$ is called a *closed formula*. We call a declaration $E : I \rightarrow \mathcal{L}(\emptyset)$ a *closed declaration*. We write $\varepsilon \in \mathcal{P}(\mathbf{P})^{\emptyset}$ for the empty environment.

Although we have defined what it means for a formula with endodeclaration to characterize a process p , when involving closed formulae, the declaration need not be specified, as in the following definition.

Definition 6.1 Let $S \subseteq \mathbf{P} \times \mathbf{P}$ be a relation. A closed formula F characterizes p up to S when, for all $p' \in \mathbf{P}$, we have $\varepsilon, p' \models F$ iff $(p, p') \in S$.

This is equivalent to the formula with endodeclaration (F, E_{\emptyset}) characterizing p up to S , where E_{\emptyset} is the \emptyset -indexed endodeclaration.

We now proceed to show how to find characteristic closed formulae up to a relation S defined as a greatest fixed point, using transfinite induction. We write On for the class of ordinals.

Definition 6.2 Let f be a monotone endofunction on a complete lattice A . We define a decreasing sequence $(f^\alpha)_{\alpha \in \text{On}}$ in A as follows:

$$\begin{aligned} f^{\alpha+1} &= f(f^\alpha). \\ \text{For } \lambda \text{ a limit ordinal, } f^\lambda &= \prod_{\alpha < \lambda} f^\alpha \quad (\text{the greatest lower bound of the set } \{f^\alpha\}_{\alpha < \lambda}). \\ \text{In particular, } f^0 &= \top \quad (\text{the top element}). \end{aligned}$$

Since A is a set, this sequence reaches its greatest lower bound, which is νf . The corresponding construction for a declaration is as follows.

Definition 6.3 Let E be an I -indexed endodeclaration. For each ordinal α , we define closed declarations $E^\alpha : I \rightarrow \mathcal{L}(\emptyset)$ as follows:

$$\begin{aligned} E^{\alpha+1} : i &\mapsto E(i)[E^\alpha]. \\ \text{For } \lambda \text{ a limit ordinal, } E^\lambda : i &\mapsto \bigwedge_{\alpha < \lambda} E^\alpha(i) \quad (\text{the conjunction of all the } E^\alpha(i) \text{ with } \alpha < \lambda). \\ \text{In particular, } E^0 : i &\mapsto \text{tt}. \end{aligned}$$

The following lemma establishes the connection between the approximants of $\llbracket E \rrbracket$ given in Definition 6.2 and the syntactic approximants to E presented in Definition 6.3.

Lemma 6.4 Let E be an I -indexed endodeclaration. For each ordinal α , we have

$$\llbracket E^\alpha \rrbracket \varepsilon = \llbracket E \rrbracket^\alpha.$$

Proof Induction on α , using Lemma 4.2(3) for the successor case. □

We note that $E^\alpha(i)$ is unchanged when we add more variables to the declaration E , in the manner of Definition 4.5.

Lemma 6.5 Let I and J be sets, and $m : I \rightarrow J$ an injection. Let E be an I -indexed endodeclaration and $D : (J \setminus \text{range}(m)) \rightarrow \mathcal{L}(J)$ a declaration. Let α be an ordinal. Then for any $i \in I$ we have

$$(E_m D)^\alpha(m(i)) = E^\alpha(i). \quad (4)$$

Proof Induction on α : the limit case is trivial and the successor case is as follows. Suppose (4) holds for all $i \in I$. For any $\hat{i} \in I$ we have

$$\begin{aligned} (E_m D)^{\alpha+1}(m(\hat{i})) &= E_m D[(E_m D)^\alpha]m(\hat{i}) && \text{(by definition)} \\ &= E_m D(m(\hat{i}))[(E_m D)^\alpha] && \text{(by definition)} \\ &= E(\hat{i})[i \mapsto X_{m(i)}][(E_m D)^\alpha] && \text{(definition of } E_m D) \\ &= E(\hat{i})[i \mapsto X_{m(i)}[(E_m D)^\alpha]] && \text{(by Lemma 4.2(2))} \\ &= E(\hat{i})[i \mapsto (E_m D)^\alpha(m(i))] && \text{(by definition)} \\ &= E(\hat{i})[i \mapsto E^\alpha(i)] && \text{(by inductive hypothesis)} \\ &= E[i \mapsto E^\alpha(i)](\hat{i}) && \text{(by definition)} \\ &= E^{\alpha+1}(\hat{i}) && \text{(by definition)} \end{aligned}$$

as required. □

Lemma 6.6 *Suppose A and B are complete lattices and ϕ is an isomorphism. Then, for each $X \subseteq A$,*

$$\phi(\bigsqcup X) = \bigsqcup \phi(X) = \bigsqcup \{\phi(x) \mid x \in X\}.$$

Lemma 6.7 *(analog of Theorem 2.3) Let A and B be complete lattices, f and g monotone endofunctions on A and B respectively, and $\phi : A \rightarrow B$ an isomorphism, such that $\phi \circ f = g \circ \phi$. Then for each ordinal α , we have that $\phi(f^\alpha) = g^\alpha$.*

Proof The statement is shown by ordinal induction, using Lemma 6.6 for a limit ordinal. \square

Once again, we assume that $P = (\mathbf{P}, \longrightarrow)$ is an \mathbf{A} -labelled transition system.

Lemma 6.8 *Suppose that \mathcal{F} is a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$, and E is a \mathbf{P} -indexed endodeclaration expressing \mathcal{F} . Let α be an ordinal and $p \in \mathbf{P}$ a process with the following property: for any $p' \in \mathbf{P}$, if $(p, p') \in \mathcal{F}^\alpha$ then $(p, p') \in v\mathcal{F}$ (the converse is automatic). Then the closed formula $E^\alpha(p)$ characterizes p up to $v\mathcal{F}$.*

Proof Since E expresses \mathcal{F} , then following an argument given in the proof of Theorem 2.16, we know that (1) commutes. Then by Lemma 6.7, for any ordinal β , we have $\Phi(\mathcal{F}^\beta) = \llbracket E \rrbracket^\beta$, where Φ is the isomorphism $\mathcal{P}(\mathbf{P} \times \mathbf{P}) \rightarrow \mathcal{P}(\mathbf{P})^{\mathbf{P}}$ mapping a relation S to σ_S (see Definition 2.17). For any $p' \in \mathbf{P}$, we reason as follows:

$$\begin{aligned} \varepsilon, p' \models E^\alpha(p) &\Leftrightarrow p' \in (\llbracket E^\alpha \rrbracket \varepsilon)(p) \\ &\Leftrightarrow p' \in \llbracket E \rrbracket^\alpha(p) && \text{(from Lemma 6.4)} \\ &\Leftrightarrow p' \in (\Phi(\mathcal{F}^\alpha))(p) && \text{(from Lemma 6.7)} \\ &\Leftrightarrow (p, p') \in \mathcal{F}^\alpha \\ &\Leftrightarrow (p, p') \in v\mathcal{F} && \text{(by the property in the statement of the lemma)} \end{aligned}$$

as required. \square

To illustrate how Lemma 6.8 can be applied, we use the following results from the literature.

Definition 6.9 *A regular cardinal is a cardinal κ for which given any I with cardinality $< \kappa$ and given any sets $\{\alpha_i\}_{i \in I}$, each with cardinality $< \kappa$, then $\bigcup_{i \in I} \alpha_i$ has cardinality $< \kappa$.*

Definition 6.10 *Let κ be an infinite regular cardinal.*

1. *A process $p \in \mathbf{P}$ is image κ -bounded when for each $a \in \mathbf{A}$ the set $\{q \in \mathbf{P} \mid p \xrightarrow{a} q\}$ has cardinality $< \kappa$.*
2. *P is an image κ -bounded \mathbf{A} -labelled transition system when, each $p \in \mathbf{P}$ is image κ -bounded.*
3. *A formula $F \in \mathcal{L}(I)$ is κ -bounded when all the conjunctions and disjunctions in it are of arity $< \kappa$.*

In particular, $\kappa = \omega$ gives image finiteness [27], and $\kappa = \omega_1$ (the smallest uncountable ordinal) gives image countability.

Theorem 6.11 *Let κ be an infinite regular cardinal. Suppose that $p \in \mathbf{P}$ is image κ -bounded.*

1. *Suppose $p' \in \mathbf{P}$ is image κ -bounded. Then $p \sqsubseteq_{sim} p'$ iff $(p, p') \in \mathcal{F}_{sim}^\kappa$.*
2. *Let $p' \in \mathbf{P}$. Then $p \sqsubseteq_{opsim} p'$ iff $(p, p') \in \mathcal{F}_{opsim}^\kappa$.*

3. Let $p' \in \mathbf{P}$. Then $p \sim_{\text{bisim}} p'$ iff $(p, p') \in \mathcal{F}_{\text{bisim}}^\kappa$.
4. Suppose $p' \in \mathbf{P}$ is image κ -bounded. Then $p \sqsubseteq_{2\text{sim}} p'$ iff $(p, p') \in (\sqsubseteq_{\text{opsim}} \sqcap \mathcal{F}_{\text{sim}})^\kappa$.

Proof In the case $\kappa = \omega$, statements (1)–(2) and (4) are proved by the same method as one found in [27]. Statement (3) was proved in [24]. The general case is similar. A detailed proof may be found in the appendix. \square

In the sequel, we write κ^+ for the successor cardinal of κ .

Corollary 6.12 *Let κ be an infinite regular cardinal.*

1. Suppose that P is an image κ -bounded system, and let $p \in \mathbf{P}$. Then the closed formula $(E_{\text{sim}})^\kappa(p)$ is κ^+ -bounded and characterizes p up to \sqsubseteq_{sim} .
2. Suppose that $p \in \mathbf{P}$ is image κ -bounded. Then the closed formula $(E_{\text{opsim}})^\kappa(p)$ is κ^+ -bounded and characterizes p up to $\sqsubseteq_{\text{opsim}}$.
3. Suppose that $p \in \mathbf{P}$ is image κ -bounded. Then the closed formula $(E_{\text{bisim}})^\kappa(p)$ is κ^+ -bounded and characterizes for p up to \sim_{bisim} .

In each case, the characteristic formula is κ^+ -bounded.

Proof We prove only statement (3) as this is the most difficult. For each ordinal α , Lemma 6.5 gives us

$$(E_{\text{bisim}})^\alpha(p) = (E_{\text{bisim}} \upharpoonright \text{reach}(p))^\alpha(p)$$

Since \mathbf{A} is assumed countable, the formulae in $E_{\text{bisim}} \upharpoonright \text{reach}(p)$ are κ^+ -bounded (indeed κ -bounded if $\kappa > \aleph_0$). Therefore, for $\alpha \leq \kappa$, the formula $(E_{\text{bisim}} \upharpoonright \text{reach}(p))^\alpha(p)$ is κ^+ -bounded, by induction on α .

Lemma 6.8 and Theorem 6.11(1) and (2)–(3) tell us that $(E_{\text{bisim}})^\kappa$ characterizes p up to \sim_{bisim} . \square

Corollary 6.12(3) gives Baltag's construction described in [4], Theorem 11.12.

We now embark on the construction of closed characteristic formulae for $\sqsubseteq_{2\text{sim}}$ using Theorem 6.11(4).

To apply Theorem 6.11(4), we use the following fact.

Lemma 6.13 *Let $R \subseteq \mathbf{P} \times \mathbf{P}$, and let $E : \mathbf{P} \rightarrow \mathcal{L}(\emptyset)$ be a closed declaration such that, for each $p \in \mathbf{P}$, the closed formula $E(p)$ characterizes p up to R . Let $\hat{E} : \mathbf{P} \rightarrow \mathcal{L}(\mathbf{P})$ be endodeclaration defined by $p \mapsto E(p)$. Then the following hold.*

1. The endodeclaration \hat{E} expresses the constant endofunction $S \mapsto R$ on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$.
2. Let \mathcal{F} be a monotone endofunction on $\mathcal{P}(\mathbf{P} \times \mathbf{P})$, and let D be a \mathbf{P} -indexed endodeclaration expressing \mathcal{F} . Then the declaration

$$p \mapsto E(p) \wedge D(p)$$

expresses $R \sqcap \mathcal{F}$.

Proof 1. Obvious.

2. Follows from (1) and Lemma 4.8. \square

Corollary 6.14 *For any ordinal α , let $E_{2\text{sim}(\alpha)}$ be the \mathbf{P} -indexed endodeclaration*

$$p \mapsto (E_{\text{opsim}})^\alpha(p) \wedge E_{\text{sim}}(p)$$

Let κ be an infinite regular cardinal, and suppose that \mathbf{P} is image κ -bounded. Let $p \in \text{Proc}$. Then the closed formula $(E_{2\text{sim}(\kappa)})^\kappa(p)$ is κ^+ -bounded and characterizes p up to $\sqsubseteq_{2\text{sim}}$.

Proof By Corollary 6.12(2) and Lemma 6.13(2), $E_{2\text{sim}(\kappa)}$ expresses $\sqsubseteq_{\text{opsim}} \sqcap \mathcal{F}_{\text{sim}}$. Then Lemma 6.8 and Theorem 6.11(4) tell us that $(E_{2\text{sim}(\kappa)})^\kappa(p)$ characterizes p up to $\sqsubseteq_{2\text{sim}}$. The bound on conjunctions and disjunctions is obvious. \square

7 Conclusion

This paper provides a general view of characteristic formulae for suitable behavioural relations. The relations of interest are those that can be defined by greatest or least fixed points of monotone endofunctions over the complete lattice of binary relations over the set of processes of a labelled transition system, which can be expressed by declarations over a given language. Theorem 2.16 shows that the greatest interpretation of an endodeclaration that expresses such a function can be viewed as the characteristic formula for the greatest fixed point of the function. We have explored a number of applications of this theorem, some in recovering characteristic formulae already discovered, and some being novel constructions. Moreover, we have shown how our technical developments and results can be extended to yield characteristic-formula constructions for the two-nested simulation preorder from [26] and for simulation equivalence (or mutual simulation).

All the behavioural relations we consider in Section 3 are greatest fixed points of suitable monotone endofunctions. However, Theorem 5.8 allows us to define in a principled and general fashion co-characteristic formulae for behavioural relations. (See Section 5.) For a behavioural equivalence, a co-characteristic formula for a process p expresses the property that any process should satisfy in order for it *not* to be equivalent to p . Least-fixed-point interpretations are the appropriate ones for defining such co-characteristic formulae since, in order to show that two processes are not equivalent, we need to find some finite ‘observation’ that only one of them affords.

Overall, this study provides a collection of ‘tools’ that one may use to define characteristic formulae for processes with respect to one’s own favourite notion of behavioural relation defined as a fixed point of a monotone endofunction. Formally, the main tools we offer for constructing characteristic formulae in a principled fashion are given in Theorem 2.16, Lemmas 3.6 and 4.8, and Theorem 4.10. We trust that the variety of examples of applications of the ‘toolbox’ we present in the paper witnesses the usability of the framework we have developed.

There are several possible avenues for further work based on the approach and on the results we present in this paper. For instance, there are still many other relations that are defined as greatest fixed points of monotone endofunctions that may lead to applications of our results. Examples include weak bisimulation congruence [41], branching bisimulation equivalence [23], resource bisimulation equivalence [14], g-bisimulation equivalence [50] and probabilistic bisimulation equivalence [35]. A general view of characteristic formulae for behavioural equivalences such as resource bisimulation and probabilistic bisimulation equivalence may require, however, a further generalization of our results.

It would also be interesting to develop a general view of characteristic formulae for behavioural relations over timed automata, such as those offered in [3, 33], as well as of the characteristic-formula constructions for bisimulation-based relations in terms of temporal logics like CTL, such as those offered in [9]. We leave these further developments for future work.

References

- [1] Luca Aceto & Matthew Hennessy (1992): *Termination, Deadlock, and Divergence*. *Journal of the ACM* 39(1), pp. 147–187. Available at <http://doi.acm.org/10.1145/147508.147527>.
- [2] Luca Aceto, Anna Ingólfssdóttir, Kim G. Larsen & Jiří Srba (2007): *Reactive Systems: Modelling, Specification and Verification*. Cambridge University Press.
- [3] Luca Aceto, Anna Ingólfssdóttir, Mikkel Lykke Pedersen & Jan Poulsen (2000): *Characteristic formulae for timed automata*. *RAIRO, Theoretical Informatics and Applications* 34(6), pp. 565–584. Available at <http://dx.doi.org/10.1051/ita:2000131>.

- [4] Jon Barwise & Lawrence Moss (1996): *Vicious Circles*, CSLI Lecture Notes 60. CSLI Publications, Stanford, CA. On the mathematics of non-wellfounded phenomena.
- [5] Jon Barwise & Lawrence S. Moss (1998): *Modal correspondence for models*. *J. Philos. Logic* 27(3), pp. 275–294. Available at <http://dx.doi.org/10.1023/A:1004268613379>.
- [6] Bard Bloom, Sorin Istrail & Albert Meyer (1995): *Bisimulation can't be Traced*. *Journal of the ACM* 42(1), pp. 232–268.
- [7] Stephen L. Bloom & Zoltán Ésik (1993): *Iteration Theories*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin.
- [8] Gérard Boudol & Kim G. Larsen (1992): *Graphical versus logical specifications*. *Theoretical Computer Science* 106(1), pp. 3–20.
- [9] M.C. Browne, E.M. Clarke & O. Grümberg (1988): *Characterizing finite Kripke structures in propositional temporal logic*. *Theoretical Computer Science* 59(1,2), pp. 115–131.
- [10] E.C. Clarke, E.A. Emerson & A.P. Sistla (1986): *Automatic verification of finite-state concurrent systems using temporal logic specifications*. *ACM Transactions on Programming Languages and Systems* 8(2), pp. 244–263.
- [11] Ed Clarke, Orna Grumberg & Doron Peled (1999): *Model Checking*. MIT Press.
- [12] E.M. Clarke & E.A. Emerson (1981): *Design and Synthesis of Synchronization Skeletons using Branching Time Temporal Logic*. In: D. Kozen, editor: *Proceedings of the Workshop on Logics of Programs, Lecture Notes in Computer Science* 131. Springer-Verlag, pp. 52–71.
- [13] Rance Cleaveland & Bernhard Steffen (1991): *Computing Behavioural Relations, Logically*. In: J. Leach Albert, B. Monien & M. Rodríguez, editors: *Proceedings 18th ICALP, Madrid, Lecture Notes in Computer Science* 510. Springer-Verlag, pp. 127–138.
- [14] Flavio Corradini, Rocco De Nicola & Anna Labella (1999): *Graded Modalities and Resource Bisimulation*. In: C. Pandu Rangan, Venkatesh Raman & Ramaswamy Ramanujam, editors: *Foundations of Software Technology and Theoretical Computer Science, 19th Conference, Chennai, India, December 13-15, 1999, Proceedings, Lecture Notes in Computer Science* 1738. Springer-Verlag, pp. 381–393. Available at <http://link.springer.de/link/service/series/0558/bibs/1738/17380381.htm>.
- [15] Rocco De Nicola, Ugo Montanari & Frits Vaandrager (1990): *Back and forth bisimulations*. In: *CONCUR'90 (Amsterdam, 1990), Lecture Notes in Comput. Sci.* 458. Springer, Berlin, pp. 152–165.
- [16] Rocco De Nicola & Frits W. Vaandrager (1995): *Three Logics for Branching Bisimulation*. *Journal of the ACM* 32(2), pp. 458–487.
- [17] Francien Dechesne, MohammadReza Mousavi & Simona Orzan (2007): *Operational and Epistemic Approaches to Protocol Analysis: Bridging the Gap*. In: *Proceedings of the 14th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR'07), Lecture Notes in Artificial Intelligence* 4790. Springer-Verlag, pp. 226–241.
- [18] H.-D. Ebbinghaus, J. Flum & W. Thomas (1994): *Mathematical Logic*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, second edition. Translated from the German by Margit Meßmer.
- [19] E. Allen Emerson (1990): *Temporal and modal logic*. In: *Handbook of Theoretical Computer Science, Vol. B*. Elsevier, Amsterdam, pp. 995–1072.
- [20] Ignacio Fábregas, David de Frutos-Escrig & Miguel Palomino (2009): *Non-strongly Stable Orders Also Define Interesting Simulation Relations*. In: Alexander Kurz, Marina Lenisa & Andrzej Tarlecki, editors: *Algebra and Coalgebra in Computer Science, Third International Conference, CALCO 2009, Udine, Italy, September 7–10, 2009. Proceedings, Lecture Notes in Computer Science* 5728. Springer-Verlag, pp. 221–235. Available at http://dx.doi.org/10.1007/978-3-642-03741-2_16.
- [21] Harald Fecher & Martin Steffen (2005): *Characteristic μ -Calculus Formula for an Underspecified Transition System*. In: *EXPRESS'04, Electronic Notes in Theoretical Computer Science* 128. Elsevier Science Publishers, pp. 103–116. Available at <http://www.informatik.uni-kiel.de/~hf/papers/Fecher04express.pdf>.
- [22] R. van Glabbeek (2001): *The linear time–branching time spectrum. I. The semantics of concrete, sequential*

- processes. In: Jan Bergstra, Alban Ponse & Scott A. Smolka, editors: *Handbook of Process Algebra*. Elsevier, pp. 3–99.
- [23] R. van Glabbeek & W.P. Weijland (1996): *Branching Time and Abstraction in Bisimulation Semantics*. *Journal of the ACM* 43(3), pp. 555–600.
- [24] Rob J. van Glabbeek (1987): *Bounded Nondeterminism and the Approximation Induction Principle in Process Algebra*. In: Franz-Josef Brandenburg, Guy Vidal-Naquet & Martin Wirsing, editors: *STACS 87, 4th Annual Symposium on Theoretical Aspects of Computer Science, Passau, Germany, February 19–21, 1987, Proceedings, Lecture Notes in Computer Science 247*. Springer-Verlag, pp. 336–347. Available at <http://dx.doi.org/10.1007/BFb0039617>.
- [25] S. Graf & J. Sifakis (1986): *A Modal Characterization of Observational Congruence on Finite Terms of CCS*. *Information and Control* 68(1–3), pp. 125–145.
- [26] Jan Friso Groote & Frits W. Vaandrager (1992): *Structured Operational Semantics and Bisimulation as a Congruence*. *Information and Computation* 100(2), pp. 202–260.
- [27] M. Hennessy & R. Milner (1985): *Algebraic laws for nondeterminism and concurrency*. *Journal of the ACM* 32(1), pp. 137–161.
- [28] Anna Ingólfssdóttir, Jens Christian Godskesen & Michael Zeeberg (1987): *Fra Hennessy-Milner Logik til CCS-Processor*. Master’s thesis, Department of Computer Science, Aalborg University. In Danish.
- [29] R.M. Keller (1976): *Formal verification of parallel programs*. *Communications of the ACM* 19(7), pp. 371–384.
- [30] H. Korver (1992): *Computing Distinguishing Formulas for Branching Bisimulation*. In: K.G. Larsen & A. Skou, editors: *Proceedings of the Third Workshop on Computer Aided Verification, Aalborg, Denmark, July 1991, Lecture Notes in Computer Science 575*. Springer-Verlag, pp. 13–23.
- [31] D. Kozen (1983): *Results on the propositional mu-calculus*. *Theoretical Computer Science* 27, pp. 333–354.
- [32] Antonín Kucera & Ph. Schnoebelen (2006): *A general approach to comparing infinite-state systems with their finite-state specifications*. *Theoretical Computer Science* 358(2–3), pp. 315–333. Available at <http://dx.doi.org/10.1016/j.tcs.2006.01.021>.
- [33] F. Laroussinie, K. G. Larsen & C. Weise (1995): *From Timed Automata to Logic - and Back*. In: Jiri Wiedermann & Petr Hájek, editors: *Mathematical Foundations of Computer Science 1995, 20th International Symposium, Lecture Notes in Computer Science 969*. Springer, Prague, Czech Republic, pp. 529–539.
- [34] Kim Guldstrand Larsen (1990): *Proof Systems for Satisfiability in Hennessy–Milner Logic with Recursion*. *Theoretical Computer Science* 72(2–3), pp. 265–288.
- [35] Kim Guldstrand Larsen & A. Skou (1992): *Compositional Verification of Probabilistic Processes*. In: Rance Cleaveland, editor: *Proceedings CONCUR 92, Stony Brook, NY, USA, Lecture Notes in Computer Science 630*. Springer-Verlag, pp. 456–471.
- [36] Kim Gulstrand Larsen & A. Skou (1991): *Bisimulation through Probabilistic Testing*. *Information and Computation* 94(1), pp. 1–28.
- [37] S B Lassen (1998): *Relational Reasoning about Functions and Nondeterminism*. Ph.D. thesis, Univ. of Aarhus.
- [38] P. B. Levy (2009): *Boolean Precongruences*. Journal submission.
- [39] Tiziana Margaria & Bernhard Steffen (1993): *Distinguishing Formulas for Free*. In: *Proc. EDAC–EUROASIC’93: IEEE European Design Automation Conference, Paris (France)*. IEEE Computer Society Press.
- [40] R. Milner (1981): *A modal characterisation of observable machine behaviour*. In: E. Astesiano & C. Böhm, editors: *CAAP ’81: Trees in Algebra and Programming, 6th Colloquium, Lecture Notes in Computer Science 112*. Springer-Verlag, pp. 25–34.
- [41] R. Milner (1989): *Communication and Concurrency*. Prentice-Hall International, Englewood Cliffs.
- [42] A. K. Moran (1998): *Call-by-name, Call-by-need, and McCarthy’s Amb*. Ph.D. thesis, Department of Computing Science, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden.

- [43] Lawrence S. Moss (2007): *Finite models constructed from canonical formulas*. *J. Philos. Logic* 36(6), pp. 605–640. Available at <http://dx.doi.org/10.1007/s10992-007-9052-4>.
- [44] Markus Müller-Olm (1998): *Derivation of Characteristic Formulae*. In: *MFCS'98 Workshop on Concurrency (Brno, 1998)*, *Electron. Notes Theor. Comput. Sci.* 18. Elsevier, Amsterdam, p. 12 pp. (electronic).
- [45] D. Park (1981): *Concurrency and automata on infinite sequences*. In: P. Deussen, editor: *5th GI Conference, Karlsruhe, Germany*, *Lecture Notes in Computer Science* 104. Springer-Verlag, pp. 167–183.
- [46] C Pitcher (2001): *Functional Programming and Erratic Non-Determinism*. Ph.D. thesis, Oxford Univ.
- [47] Gordon D. Plotkin (2004): *A Structural Approach to Operational Semantics*. *Journal of Logic and Algebraic Programming* 60–61, pp. 17–139.
- [48] Amir Pnueli (1977): *The Temporal Logic of Programs*. In: *Proceedings 18th Annual Symposium on Foundations of Computer Science*. IEEE, pp. 46–57.
- [49] J. P. Queille & J. Sifakis (1981): *Specification and Verification of Concurrent Systems in Cesar*. In: *Proceedings of the 5th International Symposium on Programming, Lecture Notes in Computer Science* 137. Springer-Verlag, pp. 337–351.
- [50] M. de Rijke (2000): *A Note on Graded Modal Logic*. *Studia Logica* 64(2), pp. 271–283.
- [51] Bernhard Steffen & Anna Ingolfsdottir (1994): *Characteristic Formulae for Processes with Divergence*. *Information and Computation* 110(1), pp. 149–163.
- [52] Colin Stirling (1987): *Modal logics for communicating systems*. *Theoret. Comput. Sci.* 49(2-3), pp. 311–347. Twelfth international colloquium on automata, languages and programming (Nafplion, 1985).
- [53] A. Tarski (1955): *A Lattice-Theoretical Fixpoint Theorem and its Applications*. *Pacific Journal of Mathematics* 5, pp. 285–309.
- [54] Wolfgang Thomas (1993): *On the Ehrenfeucht-Fraïssé Game in Theoretical Computer Science*. In: Marie-Claude Gaudel & Jean-Pierre Jouannaud, editors: *TAPSOFT, Lecture Notes in Computer Science* 668. Springer, pp. 559–568. Available at http://dx.doi.org/10.1007/3-540-56610-4_89.
- [55] B. Thomsen (1987): *An extended bisimulation induced by a preorder on actions*. Master's thesis, Aalborg University Centre.
- [56] I. Ulidowski (1992): *Equivalences on Observable Processes*. In: Andre Scedrov, editor: *Proceedings of the 7th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, Santa Cruz, CA, pp. 148–161.

A Proof of Theorem 6.11

Given \mathbf{A} -labelled transition systems $Q = (\mathbf{Q}, \longrightarrow)$ and $Q' = (\mathbf{Q}', \longrightarrow)$, we define three monotone endofunctions

$$\mathcal{F}_{sim}^{Q,Q'}, \quad \mathcal{F}_{opsim}^{Q,Q'}, \quad \mathcal{F}_{bisim}^{Q,Q'}$$

on $\mathcal{P}(\mathbf{Q} \times \mathbf{Q}')$, just as in Sections 3.1.1 and 3.2.1. Using these, we define relations

$$\sqsubseteq_{sim}^{Q,Q'}, \quad \sqsubseteq_{opsim}^{Q,Q'}, \quad \sqsubseteq_{bisim}^{Q,Q'}, \quad \sqsubseteq_{2sim}^{Q,Q'}$$

each a subset of $\mathbf{Q} \times \mathbf{Q}'$, just as in Sections 3.1.1, 3.2.1 and 4.4. We shall show the following:

Theorem A.1 *Let $Q = (\mathbf{Q}, \longrightarrow)$ and $Q' = (\mathbf{Q}', \longrightarrow)$ be \mathbf{A} -labelled transition systems. Let κ be an infinite regular cardinal.*

1. *If Q' is κ -bounded then $\sqsubseteq_{sim}^{Q,Q'} = (\mathcal{F}_{sim}^{Q,Q'})^\kappa$.*
2. *If Q is κ -bounded then $\sqsubseteq_{opsim}^{Q,Q'} = (\mathcal{F}_{opsim}^{Q,Q'})^\kappa$.*
3. *If Q is κ -bounded then $\sqsubseteq_{bisim}^{Q,Q'} = (\mathcal{F}_{bisim}^{Q,Q'})^\kappa$.*

4. If Q' is κ -bounded then $\sqsubseteq_{2sim}^{Q,Q'} = (\sqsubseteq_{opsim}^{Q,Q'} \cap \mathcal{F}_{sim}^{Q,Q'})^\kappa$.

To see that this implies Theorem 6.11, we reason as follows.

Lemma A.2 *Let $P = (\mathbf{P}, \longrightarrow)$ be an \mathbf{A} -labelled transition system, and let \mathbf{Q}, \mathbf{Q}' be subsystems (in the sense of Definition 4.7(1)) of P , giving systems $Q = (\mathbf{Q}, \longrightarrow)$ and $Q' = (\mathbf{Q}', \longrightarrow)$. Then we have*

$$\sqsubseteq_{sim}^{Q,Q'} = \sqsubseteq_{sim} \cap (\mathbf{Q} \times \mathbf{Q}') \quad (5)$$

$$\sqsubseteq_{opsim}^{Q,Q'} = \sqsubseteq_{opsim} \cap (\mathbf{Q} \times \mathbf{Q}') \quad (6)$$

$$\sqsubseteq_{bisim}^{Q,Q'} = \sqsubseteq_{bisim} \cap (\mathbf{Q} \times \mathbf{Q}') \quad (7)$$

$$\sqsubseteq_{2sim}^{Q,Q'} = \sqsubseteq_{2sim} \cap (\mathbf{Q} \times \mathbf{Q}') \quad (8)$$

and for any ordinal α we have

$$(\mathcal{F}_{sim}^{Q,Q'})^\alpha = \mathcal{F}_{sim}^\alpha \cap (\mathbf{Q} \times \mathbf{Q}') \quad (9)$$

$$(\mathcal{F}_{opsim}^{Q,Q'})^\alpha = \mathcal{F}_{opsim}^\alpha \cap (\mathbf{Q} \times \mathbf{Q}') \quad (10)$$

$$(\mathcal{F}_{bisim}^{Q,Q'})^\alpha = \mathcal{F}_{bisim}^\alpha \cap (\mathbf{Q} \times \mathbf{Q}') \quad (11)$$

$$(\sqsubseteq_{opsim}^{Q,Q'} \cap \mathcal{F}_{sim}^{Q,Q'})^\alpha = (\sqsubseteq_{opsim} \cap \mathcal{F}_{sim})^\alpha \cap (\mathbf{Q} \times \mathbf{Q}') \quad (12)$$

Proof (9)–(11) are proved by induction on α , and we deduce (5)–(7). Then (12) is proved by induction on α , using (6), and we deduce (8). \square

(We could also prove (5)–(8) directly from Lemma 4.4, in the same way that we proved Lemma 4.6.)

Using Lemma A.2, Theorem 6.11 follows immediately from Theorem A.1, by setting $\mathbf{Q} = \text{reach}(p)$ and $\mathbf{Q}' = \text{reach}(p')$. We shall now prove Theorem A.1, and we shall often omit the superscripts Q, Q' .

Definition A.3 *Let κ be an infinite regular cardinal. A poset U is downwards κ -directed when every $V \subseteq U$ of size $< \kappa$ has a lower bound.*

For example, U is downwards \aleph_0 -directed when it is nonempty and any two elements have a lower bound.

Lemma A.4 *Let κ be an infinite regular cardinal, and let A and B be complete lattices. For any function $f : A \longrightarrow B$, the following statements are equivalent.*

1. For every downwards κ -directed $U \subseteq A$, we have $f(\prod_{a \in U} a) = \prod_{a \in U} f(a)$.
2. f is monotone; and for every downwards κ -directed $U \subseteq A$, we have $f(\prod_{a \in U} a) \sqsupseteq \prod_{a \in U} f(a)$.

Proof (1) \Rightarrow (2) For monotonicity, suppose $a \sqsubseteq b \in A$. Then $\{a, b\}$ is downwards κ -directed (as a is a least element). So

$$f(a) = f(a \sqcap b) = f(a) \sqcap f(b) \sqsubseteq f(b)$$

(2) \Rightarrow (1) Let $U \subseteq A$. For each $a \in U$ we have $f(\prod_{a \in U} a) \sqsubseteq f(a)$ by monotonicity. So $f(\prod_{a \in U} a) \sqsubseteq \prod_{a \in U} f(a)$. \square

We say that a function $f : A \longrightarrow B$ *preserves κ -directed meets* when it satisfies the conditions of Lemma A.4.

Preservation of κ -directed meets has the following application.

Lemma A.5 *Let κ be an infinite regular cardinal. Let f be an endofunction on a complete lattice A that preserves κ -directed meets. Then $\forall f = f^\kappa$.*

Proof The set $\{f^\alpha\}_{\alpha < \kappa}$ is downwards κ -directed, since κ is regular. So

$$\begin{aligned} f(f^\kappa) &= f\left(\bigcap_{\alpha < \kappa} f^\alpha\right) \\ &= \bigcap_{\alpha < \kappa} f(f^\alpha) \\ &= \bigcap_{\alpha < \kappa} f^{\alpha+1} \\ &= f^\kappa \end{aligned}$$

□

Lemma A.6 Let A and B be complete lattices, and let κ be an infinite regular cardinal.

1. For each $i \in I$, let $f_i : A \rightarrow B$ be a κ -directed meet preserving function. Then the function $A \rightarrow B$ mapping a to $\bigcap_{i \in I} f_i(a)$ preserves κ -directed meets.
2. Let $b \in B$. Then the constant function $A \rightarrow B$ mapping any a to b preserves κ -directed meets.
3. Let $b \in B$ and let $f : A \rightarrow B$ be a function preserving κ -directed meets. Then the function $b \sqcap f : A \rightarrow B$ preserves κ -directed meets.

Proof 1. Trivial.

2. This follows from the fact that a downwards κ -directed set must be nonempty.

3. This follows from parts (1)–(2). □

Lemma A.7 Let C and D be sets, and let κ be an infinite regular cardinal. For any function $f : \mathcal{P}(C) \rightarrow \mathcal{P}(D)$, the following statements are equivalent.

1. f preserves κ -directed meets.
2. For any $M \subseteq C$ and $d \in D$, if $d \notin f(M)$ then there is some $N \subseteq C$, disjoint from M and of size $< \kappa$, such that, for any $M' \subseteq C$ disjoint from N , we have $d \notin f(M')$.

Proof (1) \Rightarrow (2) The set $\{C \setminus N \mid N \subseteq A, N \cap M = \emptyset, |N| < \kappa\}$ is downwards κ -directed (because κ is regular), and its intersection is M . So we have

$$\begin{aligned} f(M) &= f\left(\bigcap_{\substack{N \subseteq A \\ N \cap M = \emptyset, |N| < \kappa}} (C \setminus N)\right) \\ &= \bigcap_{\substack{N \subseteq A \\ N \cap M = \emptyset, |N| < \kappa}} f(C \setminus N) \end{aligned}$$

Suppose $d \notin f(M)$. Then there is $N \subseteq A$, disjoint from M and of size $< \kappa$, such that $d \notin f(C \setminus N)$. For any $M' \subseteq A$ disjoint from N , we have $M' \subseteq C \setminus N$, so monotonicity of f gives $f(M') \subseteq f(C \setminus N)$. Hence $d \notin f(M')$.

(2) \Rightarrow (1) To show f monotone, suppose $M' \subseteq M \subseteq C$. If $d \notin f(M)$, then there exists $N \subseteq C$, disjoint from M and of size $< \kappa$, such that, for any $M'' \subseteq C$ disjoint from N , we have $d \notin f(M'')$. In particular M' is disjoint from N , so $d \notin f(M)$. We conclude that $f(M') \subseteq f(M)$.

Let $U \subseteq \mathcal{P}(C)$ be downwards κ -directed. If $d \notin f(\bigcap_{u \in U} u)$ then there is $N \subseteq C$, disjoint from $\bigcap_{u \in U} u$ and of size $< \kappa$, such that, for any $M' \subseteq C$ disjoint from N , we have $d \notin f(M')$. For each $a \in N$ pick $u_a \in U$ such that $a \notin u_a$. We know that $\{u_a \mid a \in N\}$ has a lower bound $u' \in U$, so for each $a \in N$ we have $a \notin u'$, i.e. u' is disjoint from N , so $d \notin f(u')$ and hence $d \notin \bigcap_{u \in U} f(u)$. We conclude that $\bigcap_{u \in U} f(u) \subseteq f(\bigcap_{u \in U} u)$. \square

Remark Lemma A.7 is an easy equivalent (by duality) of the following well-known result: a function $g: \mathcal{P}(C) \rightarrow \mathcal{P}(D)$ preserves κ -directed joins iff, for every $M \subseteq C$ and $d \in D$, if $d \in g(M)$ then there is $N \subseteq M$ of size $< \kappa$ such that, for every $M' \subseteq C$ such that $N \subseteq M'$, we have $d \in g(M')$.

Lemma A.8 Let $Q = (\mathbf{Q}, \rightarrow)$ and $Q' = (\mathbf{Q}', \rightarrow)$ be \mathbf{A} -labelled transition systems. Let κ be an infinite regular cardinal.

1. If Q' is image κ -bounded, then $\mathcal{F}_{sim}^{Q, Q'}$ preserves κ -directed meets.
2. If Q is image κ -bounded, then $\mathcal{F}_{opsim}^{Q, Q'}$ preserves κ -directed meets.
3. If Q and Q' are both image κ -bounded, then \mathcal{F}_{bisim} preserves κ -directed meets.
4. If Q' is image κ -bounded, then $\sqsubseteq_{opsim}^{Q, Q'} \cap \mathcal{F}_{sim}^{Q, Q'}$ preserves κ -directed meets.

Proof 1. We use Lemma A.7. Suppose $R \subseteq \mathbf{Q} \times \mathbf{Q}'$ and $(p, p') \notin \mathcal{F}_{sim}(R)$. Then there is $a \in \mathbf{A}$ and $q \in \mathbf{Q}$ such that $p \xrightarrow{a} q$ but there does not exist any $q' \in \mathbf{Q}'$ such that $p' \xrightarrow{a} q'$ and $(q, q') \in R$. Put $N = \{(q, q') \mid p' \xrightarrow{a} q'\}$. We know that N is disjoint from R , and it has size $< \kappa$ because Q' is κ -bounded. For any $R' \subseteq \mathbf{Q} \times \mathbf{Q}'$, if R' is disjoint from N then there does not exist $q' \in \mathbf{Q}'$ such that $p' \xrightarrow{a} q'$ and $(q, q') \in R'$, so $(p, p') \notin \mathcal{F}_{sim}(R')$. This establishes the required condition.

2. The dual result.
3. Similar argument.
4. This follows from part (1) and Lemma A.6(3). \square

Using Lemma A.5, we deduce parts (1), (4) and (4) of Theorem A.1 from Lemma A.8. But part (3) requires a more subtle argument.

Definition A.9 Let A and B be sets. A relation $R \subseteq A \times B$ is difunctional when, for all $a_1, a_2 \in A$ and $b_1, b_2 \in B$, if $(a_1, b_1) \in R$ and $(a_1, b_2) \in R$ and $(a_2, b_1) \in R$ we have $(a_2, b_2) \in R$. We write $\text{Difun}(A, B)$ for the poset of difunctional relations from A to B , ordered by inclusion.

We note that an intersection of difunctional relations is also difunctional, so $\text{Difun}(A, B)$ is a complete lattice.

Lemma A.10 Let $Q = (\mathbf{Q}, \rightarrow)$ and $Q' = (\mathbf{Q}', \rightarrow)$ be \mathbf{A} -labelled transition systems. Let R be a difunctional relation from \mathbf{Q} to \mathbf{Q}' . Then $\mathcal{F}_{bisim}^{Q, Q'}(R)$ is also difunctional.

Proof Suppose $(p_1, p'_1), (p_1, p'_2), (p_2, p'_1) \in \mathcal{F}_{bisim}(R)$. If $p_2 \xrightarrow{a} q_2$, then

- $(p_2, p'_1) \in \mathcal{F}_{bisim}(R)$ gives $q'_1 \in \mathbf{Q}'$ such that $p'_1 \xrightarrow{a} q'_1$ and $(q_2, q'_1) \in R$
- $(p_1, p'_1) \in \mathcal{F}_{bisim}(R)$ gives $q_1 \in \mathbf{Q}$ such that $p_1 \xrightarrow{a} q_1$ and $(q_1, q'_1) \in R$

- $(p_1, p'_1) \in \mathcal{F}_{bisim}(R)$ gives $q'_2 \in \mathbf{Q}$ such that $p'_2 \xrightarrow{a} q'_2$ and $(q_1, q'_2) \in R$

so difunctionality of R gives $(q_2, q'_2) \in R$. We conclude that $(p_2, p'_2) \in \mathcal{F}_{bisim}(R)$. Straightforward. \square

We write $\mathcal{G}^{Q, Q'}$ for the restriction of $\mathcal{F}_{bisim}^{Q, Q'}$ to an endofunction on $\text{Difun}(\mathbf{Q}, \mathbf{Q}')$. This will enable us to replace Lemma A.8(3) with one in which Q' need not be κ -bounded. We first give a useful fact.

Lemma A.11 *Let κ be an infinite regular cardinal, and let C be a set of size $< \kappa$. Then any $U \subseteq \mathcal{P}(C)$ that is downwards κ -directed has a least element.*

Proof For each $c \in C \setminus \bigcap_{u \in U} u$, pick $u_c \in U$ such that $c \notin u_c$. Then the set $\{u_c \mid c \in C \setminus \bigcap_{u \in U} u\}$ has a lower bound $u' \in U$. If $c \in u' \setminus \bigcap_{u \in U} u$ then $c \notin u_c$, contradicting $u \subseteq u_c$. Hence $u' = \bigcap_{u \in U} u$. \square

Lemma A.12 *Let κ be an infinite regular cardinal. Let $Q = (\mathbf{Q}, \longrightarrow)$ and $Q' = (\mathbf{Q}', \longrightarrow)$ be \mathbf{A} -labelled transition systems, with Q image κ -bounded. Then $\mathcal{G}^{Q, Q'}$ preserves κ -directed meets.*

Proof The monotonicity of \mathcal{G} is inherited from that of \mathcal{F}_{bisim} .

Let U be a downwards κ -directed subset of $\text{Difun}(Q, Q')$. We have

$$\begin{aligned} \bigcap_{R \in U} \mathcal{G}(R) &\subseteq \bigcap_{R \in U} \mathcal{F}_{opsim}(R) \\ &= \mathcal{F}_{opsim}(\bigcap_{R \in U} R) \quad (\text{by Lemma A.8(2)}) \end{aligned}$$

We need only prove

$$\bigcap_{R \in U} \mathcal{G}(R) \subseteq \mathcal{F}_{sim}(\bigcap_{R \in U} R) \quad (13)$$

for then we can conclude

$$\begin{aligned} \bigcap_{R \in U} \mathcal{G}(R) &\subseteq \mathcal{F}_{sim}(\bigcap_{R \in U} R) \cap \mathcal{F}_{opsim}(\bigcap_{R \in U} R) \\ &= \mathcal{G}(\bigcap_{R \in U} R) \end{aligned}$$

Suppose that $(p, p') \in \bigcap_{R \in U} \mathcal{G}(R)$ and $p \xrightarrow{a} q$. Let $B = \{r \in \mathbf{P} \mid p \xrightarrow{a} r\}$. We define a function $H : U \rightarrow \mathcal{P}(B)$ mapping R to

$$\{r \in B \mid \exists q' \in \mathbf{Q}'. (q, q') \in R, (r, q') \in R\}$$

H is monotone, so the set $L = \{H(R) \mid R \in U\}$ is downwards κ -directed; and by Lemma A.11 has a least element, since B has size $< \kappa$. We pick some $S \in U$ such that $H(S)$ is the least element of L .

Now suppose $p \xrightarrow{a} q$ and $(p, p') \in \bigcap_{R \in U} \mathcal{G}(R)$. We know that there is $q' \in \mathbf{Q}'$ such that $p' \xrightarrow{a} q'$ and $(q, q') \in S$.

For any $R \in U$, pick a lower bound $R' \in U$ for R and S . There is $r \in \text{Proc}$ such that $p \xrightarrow{a} r$ and $(r, q') \in R'$. Since $(r, q') \in S$, we have $r \in H(S) = H(R')$. So there exists $r' \in \mathbf{Q}$ such that $(q, r') \in R'$ and $(r, r') \in R'$. Difunctionality of R' gives $(q, q') \in R'$, hence $(q, q') \in R$. Thus $(q, q') \in \bigcap_{R \in U} R$. We conclude that $(p, p') \in \mathcal{F}_{sim}(\bigcap_{R \in U} R)$. \square

Proof (of Theorem A.1(3)) By induction on $\alpha \in \text{On}$, we have $(\mathcal{F}_{bisim})^\alpha = \mathcal{G}^\alpha$, and Lemma A.5 and Lemma A.12 tell us that $\mathcal{G}^\alpha_{\alpha \in \text{On}}$ reaches its infimum at κ . \square