Characteristic formulae for fixed-point semantics: a general framework

LUCA ACETO†, ANNA INGOLFSDOTTIR‡, PAUL BLAIN LEVY‡ and JOSHUA SACK†

†School of Computer Science, Reykjavik University, IS-101 Reykjavik, Iceland
Email: {luca; annai; joshua}@ru.is
‡University of Birmingham, Birmingham B15 2TT, UK
Email: pbl@cs.bham.ac.uk

Received 6 February 2010; revised 18 August 2010

The concurrency theory literature 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 paper provides a general view of characteristic formulae that are expressed in terms of logics that have a facility for the recursive definition of formulae. We show how several examples of characteristic-formula constructions in 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 provide 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) (Keller 1976). LTSs underlie Plotkin’s Structural Operational Semantics (Plotkin 2004) and, following Milner’s pioneering work on CCS (Milner 1989), have become 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 formally describes when (states of) LTSs provide 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) one LTS provides at least as many ‘observations’ as another does. 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 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. Van Glabbeek in his now classic paper van Glabbeek (2001) presented a taxonomy of existing 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 model checking (Clarke et al. 1999; Clarke and Emerson 1981; Queille and Sifakis 1981). In this approach, automata are still the formalism of choice for the description of the actual behaviour of a concurrent system, but specifications of the expected behaviour of a system are now expressed using a suitable logic, for instance, a modal or temporal logic (Emerson 1990; Pnueli 1997). 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 characterisation theorem of bisimulation equivalence (Milner 1989; Park 1981) in terms of Hennessy–Milner logic (HML), which is due to Hennessy and Milner (Hennessy and Milner 1985). 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 they are not. This distinguishing formula is useful for debugging purposes, and can be algorithmically constructed for finite processes – see, for example, Korver (1991) and Margaria and Steffen (1993).

However, Hennessy and Milner’s characterisation theorem is less useful if we are interested in using it directly to establish when two processes are behaviourally equivalent using model checking. Indeed, the theorem seems to indicate that in order to show that two processes are equivalent, we need to check that they satisfy all the formulae expressible in the logic, and there are infinitely many such formulae, even modulo logical equivalence. So can we find a single formula that characterises the bisimulation equivalence class of a process \( p \) (in the sense that any process is bisimilar to \( p \) if and only if it has 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. Indeed, to check whether a process \( q \) is bisimilar to \( p \), say, it would suffice to construct the characteristic formula for \( p \) and verify whether \( q \) satisfies it using a model checker. (See, for instance, Boudol and Larsen (1992) for an investigation of the model checking problems that can be reduced to implementation verification.)
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 Graf and Sifakis (1986). In that study, Graf and Sifakis presented a translation from recursion-free terms of Milner's CCS (Milner 1989) into formulae of a modal language representing their equivalence class with respect to observational congruence.

So, can one characterise 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 no because each formula in that logic can only describe a finite fragment of the initial behaviour of a process – see, for instance, Aceto et al. (2007) for a textbook presentation. However, as shown, for example, in Ingolfsdottir et al. (1987) and Steffen and Ingolfsdottir (1994), 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 from the work presented in the original references cited above, the literature on concurrency theory now offers 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, Aceto et al. (2000), Browne et al. (1988), Cleaveland and Steffen (1991), Fecher and Steffen (2005), Laroussinie et al. (1995), Larsen and Skou (1992) and Müller-Olm (1998). Such constructions and their proofs of correctness have been developed independently, but have a common underlying structure. It is therefore natural to ask whether one can devise 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 would such general principles allow us to recover existing 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 $F$ underlying the definition of a behavioural relation, then the greatest interpretation of that collection of formulae characterises the behavioural relation that is the greatest fixed point of $F$ – see Theorem 2.16. Using this result, we can 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 (Park 1981), bisimulation (Milner 1989; Park 1981), ready simulation (Bloom et al. 1995; Larsen and Skou 1991) and prebisimulation semantics (Aceto and Hennessy 1992; Milner 1981). In addition, we show that the framework can be used to yield novel, albeit at times unsurprising, constructions. As an example, we provide characteristic formulae for back and forth bisimilarity, back and forth bisimilarity with indistinguishable states (Dechesne et al. 2007), the boolean precongruences introduced in Levy (2009), conformance simulation semantics (Fábregas et al. 2009) and extended simulation semantics (Thomsen 1987).

The techniques developed in Section 2.3 do not readily yield characteristic-formula constructions for the nested simulation semantics of Groote and Vaandrager (1992) 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 also.

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 a 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 provides. As we show in Section 5, each of the results for characteristic formulae given in 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 paper will offer a framework for the derivation of many more such results and for explaining the reasons underlying the success of the constructions of this kind that have already appeared 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 given in Browne et al. (1988), which showed that any finite Kripke structure can be characterised by a formula in Computation Tree Logic (CTL) (Clarke et al. 1986) up to the natural variation on bisimilarity over Kripke structures. Browne et al. (1988) presented another characteristic formula result for an equivalence between states that takes ‘stuttering’ into account – this equivalence is closely related to van Glabbeek’s and Weijland’s branching bisimilarity (van Glabbeek and Weijland 1996), for which logical characterisations were given in De Nicola and Vaandrager (1995). Browne, Clarke and Grümberg show that equivalence classes of states
in a finite Kripke structure modulo stuttering equivalence are completely characterised by next-time-free CTL formulae – the absence of the next-time operator is expected because of the inability of stuttering equivalence to ‘count’ the number of steps in a stuttering sequence. Kučera and Schnoebelen (2006) presented a refinement of the above classic theorem by Browne, Clarke and Grumberg.

Characteristic formulae for bisimilarity, expressed in terms of infinitary modal logic, are given in Barwise and Moss (1998, Theorem 3.2) over Kripke structures and in Barwise and Moss (1996, 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 van Glabbeek (1987) that if a process \( p \) is image finite and \( p \) and \( q \) are related by the \( \omega \)-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 Barwise and Moss (1996, Theorem 11.12) relies on their extension of the aforementioned result of van Glabbeek’s to infinite regular cardinals other than \( \omega \): see Barwise and Moss (1996, Lemma 11.13) – Barwise and Moss attribute the proof they present to Baltag.

In Section 6, we will present a general account of the above-mentioned characteristic formula constructions in terms of both infinitary modal logics and some novel logics.

The theoretical significance of characteristic formulae in infinitary modal logic is exemplified by the developments in Moss (2007), 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, it would be remiss of us 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 to the equivalence over structures induced by formulae whose quantifier depth is at most \( m \) – see, for example, Ebbinghaus et al. (1994) and Thomas (1993) for overviews.

**Roadmap of the paper**

In Section 2, we describe the theoretical background and the main technical results the paper relies on: in particular, we present our main technical result (Theorem 2.16), which lies 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. As mentioned 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 develop characteristic formulae for those relations. Section 5 presents developments related to the above-mentioned notion of a co-characteristic formula. In Section 6, we present a general account of characteristic
formula constructions in terms of infinitary modal logics. Finally, in Section 7 we provide some concluding remarks and suggest some directions for future research.

2. Fixed points and logic

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

2.1. Posets, homomorphisms and fixed points

Definition 2.1.

(1) A partially ordered set, or poset, \((A,\subseteq_A)\) consists of a set \(A\) and a partial order \(\subseteq_A\) over it. This poset is often denoted simply by the set \(A\), and we write \(\subseteq\) instead of \(\subseteq_A\) if the meaning is clear from the context.

(2) For posets \(A\) and \(B\), a function \(\phi : A \rightarrow B\) is:

— monotone if it preserves the order over \(A\), that is, if \(x \subseteq_A y\) implies \(\phi(x) \subseteq_B \phi(y)\) for all \(x, y \in A\);

— an isomorphism if it is bijective and both \(\phi\) and its inverse \(\phi^{-1}\) 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 \(f(x) \subseteq x\);

— a post-fixed point of \(f\) when \(x \subseteq f(x)\); and

— a fixed point of \(f\) when \(f(x) = x\), that is, when \(x\) is both a pre-fixed point and a post-fixed point of \(f\).

We write \(\nu f\) for the greatest post-fixed point and \(\mu f\) 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 \(\nu f\) exists, then it is the greatest fixed point of \(f\).

(2) If \(\mu f\) exists, then it is the least fixed point of \(f\).

The theorem below, although 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 \(\phi : A \rightarrow B\) be an isomorphism such that the square

\[
\begin{array}{ccc}
A & \xrightarrow{f} & A \\
\phi\downarrow & & \phi\downarrow \\
B & \xrightarrow{g} & B
\end{array}
\]
commutes. Then the following statements hold:

(1) Let \( x \in A \). Then \( x \) is a post-fixed point (respectively, pre-fixed point, fixed point) of \( f \) if and only if \( \phi(x) \) is a post-fixed point (respectively, pre-fixed point, fixed point) of \( g \).

(2) \( \nu_f \) exists if and only if \( \nu_g \) exists, and we then have \( \phi(\nu_f) = \nu_g \).

(3) \( \mu_f \) exists if and only if \( \mu_g \) exists, and we then have \( \phi(\mu_f) = \mu_g \).

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

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 \( \bigcap U \), 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 (Tarski 1955).

**Theorem 2.5.** If \( A \) is a complete lattice and \( f \) is a monotone endofunction on \( A \), then:

— both \( \nu_f \) and \( \mu_f \) exist;

— \( \nu_f = \bigcup \{a \mid a \subseteq f(a)\} \); and

— \( \mu_f = \bigcap \{a \mid f(a) \subseteq a\} \).

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 functions, in the following we often denote a function 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 \( \sigma_1 \subseteq \sigma_2 \) if and only if \( \sigma_1(i) \subseteq_A \sigma_2(i) \) 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 \( \mathcal{F} \) 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

\[
(\nu_{\mathcal{F}})^{-1} = \nu_{\tilde{\mathcal{F}}},
\]

\[
(\mu_{\mathcal{F}})^{-1} = \mu_{\tilde{\mathcal{F}}},
\]

2.2. Hennessy–Milner logic with variables and declarations

In this section we recall the standard Hennessy–Milner logic (HML) extended with variables (see, for instance, Larsen (1990)). The logic depends on a finite set \( A \), 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 \( \mathcal{L}(I,A) \) for the set of formulae given by the following
grammar:

\[
F ::= X_i \quad (i \in I) \\
| \bigwedge_{k \in K} F_k \quad (K \text{ a finite set}) \\
| \bigvee_{k \in K} F_k \quad (K \text{ a finite set}) \\
| \langle a \rangle F \quad (a \in A) \\
| [a] F \quad (a \in A).
\]

If the meaning is clear from the context, we often omit \(I\), \(A\) or both. As usual, we write \(\top\) for nullary conjunction, \(\land\) for binary conjunction, \(\bot\) for nullary disjunction and \(\lor\) for binary disjunction.

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

**Definition 2.8.** An \(A\)-labelled transition system (LTS) is a pair \(P = (P, \rightarrow)\) consisting of:

- a finite set \(P\); and
- a transition relation \(\rightarrow \subseteq P \times A \times P\).

Typically, \(A\) will be clear from context, and we thus often refer to an \(A\)-labelled transition system as simply a labelled transition system. As usual, we write \(p \xrightarrow{a} p'\) for \((p, a, p') \in \rightarrow\). We often think of \(P\) as a set of processes, \(A\) as a set of actions and \(p \xrightarrow{a} p'\) as a transition from process \(p\) to process \(p'\) through 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 \(I\), \(A\) and \(P\) 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 \(A\) and \(I\) to be arbitrary as well. In Section 6, we will explicitly consider the version of the logic \(\mathcal{L}(\emptyset, A)\) with conjunctions and disjunctions over arbitrary cardinals and over a countable set \(A\).

Fixing \(A\), a formula is interpreted over an \(A\)-labelled transition system \((P, \rightarrow)\) as the set of elements from \(P\) 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 is more readable when using longer formulae, and the function involves a notation that should help to 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 \(\sigma \in \mathcal{P}(P)^I\) (possibly with a subscript) that associates to each \(i \in I\) the set of processes in \(P\) 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 \(\mathcal{P}(P)^I \times P \to \mathcal{L}(I)\). This relation tells us when a process \(p\) satisfies the formula \(F\) under the interpretation \(\sigma\), and is defined by structural recursion on \(F\) as follows:

\[
\sigma, p \models X_i \iff p \in \sigma(i) \\
\sigma, p \models \bigwedge_{k \in K} F_k \iff \sigma, p \models F_k \text{ for all } k \in K \\
\sigma, p \models \bigvee_{k \in K} F_k \iff \sigma, p \models F_k \text{ for some } k \in K \\
\sigma, p \models \langle a \rangle F \iff \text{there is some } p' \in P \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p \models [a] F \iff \text{for all } p' \in P \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F.
\]
Now, for each $F \in \mathcal{L}(I)$, we define $[[F]] : \mathcal{P}(\mathcal{P})^I \rightarrow \mathcal{P}(\mathcal{P})$ by

$$[[F]] \sigma = \{ p \in \mathcal{P} \mid \sigma, p \models F \}.$$  

In particular

$$[[X_i]] \sigma = \sigma(i)$$
$$[[f]] \sigma = \emptyset$$
$$[[\pi]] \sigma = \mathcal{P}$$

for all $\sigma \in \mathcal{P}(\mathcal{P})^I$. Moreover,

$$\left( \bigwedge_{k \in K} F_k \right) \sigma = \bigcap_{k \in K} ([[F_k]] \sigma)$$
$$\left( \bigvee_{k \in K} F_k \right) \sigma = \bigcup_{k \in K} ([[F_k]] \sigma).$$

**Lemma 2.9.**

(1) For any $F \in \mathcal{L}(I)$, the function $[[F]] : \mathcal{P}(\mathcal{P})^I \rightarrow \mathcal{P}(\mathcal{P})$ is monotone.

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

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

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 $\mu$-calculus (Kozen 1983), 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 $\mu$-calculus and involve systems of equations that help induce a unique variable interpretation. Following Larsen (1990), we define systems of equations implicitly using declarations.

**Definition 2.10.**

— Given index sets $I,J$ and a language $\mathcal{L}(J)$, we say a function $D : I \rightarrow \mathcal{L}(J)$ (or, equivalently, $D \in \mathcal{L}(J)^I$) is an $I$-indexed declaration for the language $\mathcal{L}(J)$.

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

Semantically, a declaration $D : I \rightarrow \mathcal{L}(J)$ induces a function $[[D]] : \mathcal{P}(\mathcal{P})^I \rightarrow \mathcal{P}(\mathcal{P})^I$ as follows.

**Definition 2.11.** If $D : I \rightarrow \mathcal{L}(J)$ is a declaration, we define $[[D]] : \mathcal{P}(\mathcal{P})^I \rightarrow \mathcal{P}(\mathcal{P})^I$ by

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

**Lemma 2.12.** If \( E \) is an endodeclaration, then \([E]\) is a monotone endofunction on \( \mathcal{P}(P)^I \) and, therefore, \( \nu[E] \) and \( \mu[E] \) exist.

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

\[ [E] \sigma_1 \subseteq [E] \sigma_2, \]

or, equivalently, that

\[ (\{[E] \sigma_1\})(i) \subseteq (\{[E] \sigma_2\})(i) \text{ for all } i \in I. \]

The claim then follows from Lemma 2.9(1) since, by definition, \( (\{E\} \sigma_j)(i) = [E(i)] \sigma_j \) for \( j = 1, 2 \).

Recall that the greatest and least fixed points of the endofunction \([E]\) induced by an endodeclaration \( E \) are elements of \( \mathcal{P}(P)^I \), that is, variable interpretations for the logic.

### 2.3. Characteristic endodeclarations

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

**Definition 2.13.** An endodeclaration \( E \) for the logic \( \mathcal{L}(P) \) characterises \( S \subseteq P \times P \) if and only if for each \( p, q \in P \),

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

In many of our examples, we seek declarations that characterise a relation, such as similarity or bisimilarity, that is of the form \( \nu F \), where \( F \) is a monotone endofunction on \( \mathcal{P}(P \times P) \). In the following, we will describe how we can devise a characterising declaration for a relation that is obtained as a fixed point of a monotone endofunction and that can be expressed in the logic. In Definition 2.15, we will use the notation introduced in the following definition.

**Definition 2.14.** If \( S \subseteq P \times P \), we define the variable interpretation \( \sigma_S \in \mathcal{P}(P)^P \) associated with \( S \) by

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

Thus, \( \sigma_S \) assigns to \( p \) all those processes \( q \) that are related to it through \( S \).
**Definition 2.15.** We say that an endodeclaration $E$ for $\mathcal{L}(P)$ expresses a monotone endofunction $F$ on $\mathcal{P}(P \times P)$ when

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

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

We are now ready to state the main theorem of this paper, which says that if a collection of recursively defined logical formulae expresses an endofunction $F$, then the greatest interpretation of that collection of formulae characterises the greatest fixed point of $F$.

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

We will use Lemma 2.18 to prove the theorem, but begin with a definition.

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

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

We are now ready to prove Theorem 2.16.

**Proof of Theorem 2.16.** To prove the theorem, we first show that the following diagram commutes.

$$
\begin{array}{ccc}
\mathcal{P}(P \times P) & \xrightarrow{\Phi} & \mathcal{P}(P)^P \\
F & \downarrow & \downarrow [E] \\
\mathcal{P}(P \times P) & \xrightarrow{\Phi} & \mathcal{P}(P)^P
\end{array}
$$

(1)

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

$$([E] \circ \Phi)(S) = [E](\Phi(S)) = [E]\sigma_S$$

and

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

To prove (1), it is therefore sufficient to prove that $[E]\sigma_S = \sigma_{F(S)}$. To do this, we assume that $p \in P$. Then, since $E$ expresses $F$ (in the sense of Definition 2.15),

$$([E]\sigma_S)(p) = ([E](p))\sigma_S = \{q \mid \sigma_S, q \models E(p)\} = \{q \mid (p, q) \in F(S)\} = \sigma_{F(S)}(p).$$

By Lemma 2.18, $\Phi$ is an isomorphism. Hence, Theorem 2.3 gives $\Phi(\nu F) = \nu [E]$. As $(p, q) \in \nu F$ if and only if $q \in \sigma_{\nu F}(p)$ and $\sigma_{\nu F}(p) = \Phi(\nu F)(p)$, this implies that for any $p, q \in P$,

$$(p, q) \in \nu F \iff q \in (\nu [E])(p).$$

Therefore $E$ characterises $\nu F$ (in the sense of Definition 2.13).

3. Applications

In this section we will apply Theorem 2.16 to obtain characteristic formulae for various behavioural relations that are defined as greatest fixed points of monotone endofunctions
over variations on labelled transition systems. Most of the results we present in this section already appear in the literature on concurrency theory, but 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 describe some examples of characteristic formulae for various types of similarity.

3.1.1. Simulation (Park 1981). Given an A-labelled transition system $P = (P, \rightarrow)$ and a relation $S \subseteq P \times P$, let $F_{\text{sim}}(S)$ be defined by:

$$(p, q) \in F_{\text{sim}}(S) \text{ if and only if for every } a \in A \text{ and } p' \in P:$$

- if $p \xrightarrow{a} p'$, then there exists a $q' \in P$ such that $q \xrightarrow{a} q'$ and $(p', q') \in S$.

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

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

$$(p, q) \in F_{\text{sim}}(S) \iff \forall a \in A \forall p' \in P. (p \xrightarrow{a} p' \Rightarrow \exists q' \in P. q \xrightarrow{a} q' \& (p', q') \in S)$$

This shows that the endodeclaration

$$E_{\text{sim}} : p \mapsto \bigwedge_{a \in A} \bigwedge_{p' \in P. p \xrightarrow{a} p'} \langle a \rangle X_{p'}$$

expresses $F_{\text{sim}}$. Therefore the following proposition follows from Theorem 2.16.

**Proposition 3.1.** The endodeclaration $E_{\text{sim}}$ characterises the preorder $\sqsubseteq_{\text{sim}}$.

We will now look for characteristic formulae for $\sqsubseteq_{\text{opsim}} = (\sqsubseteq_{\text{sim}})^{-1}$. Lemma 2.7 implies that $\sqsubseteq_{\text{opsim}} = \nu F_{\text{opsim}}$ where

$$F_{\text{opsim}} : S \mapsto (F_{\text{sim}}(S^{-1}))^{-1}.$$ 

Then, for all $p, q \in P$,

$$(p, q) \in F_{\text{opsim}}(S)$$

if and only if

$$(q, p) \in F_{\text{sim}}(S^{-1})$$

if and only if for every $a \in A$ and $q' \in P$,

if $q \xrightarrow{a} q'$, then there exists some $p' \in P$ such that $p \xrightarrow{a} p'$ and $(q', p') \in S^{-1}$,
or, equivalently,

\[(p, q) \in \mathcal{F}_{\text{opsim}}(S)\]

if and only if for every \(a \in A\) and \(q' \in P\),

if \(q \xrightarrow{a} q'\), then there exists some \(p' \in P\) such that \(p \xrightarrow{a} p'\) and \((p', q') \in S\).

This can be expressed in the logic \(\mathcal{L}(P)\) as follows:

\[(p, q) \in \mathcal{F}_{\text{opsim}}(S) \iff \forall a \in A. \forall p' \in P. (q \xrightarrow{a} q' \Rightarrow \exists p' \in P. p \xrightarrow{a} p' \& (p', q') \in S)\]

This shows that the endodeclaration \(E_{\text{opsim}}:\ P \mapsto \bigwedge_{a \in A} \bigvee_{p' \in P. p \xrightarrow{a} p'} X_{p'}\)

expresses \(\mathcal{F}_{\text{opsim}}\). Therefore, Theorem 2.16 gives us the following proposition.

**Proposition 3.2.** The endodeclaration \(E_{\text{opsim}}\) characterises the preorder \(\sqsubseteq_{\text{opsim}}\).

3.1.2. **Ready simulation** (Bloom et al. 1995; Larsen and Skou 1991). Given an \(A\)-labelled transition system \(P = (P, \rightarrow)\) and a relation \(S \subseteq P \times P\), let \(\mathcal{F}_{RS}(S)\) be defined by

\[(p, q) \in \mathcal{F}_{RS}(S) \iff \forall a \in A. \forall p' \in P. (p \xrightarrow{a} p' \Rightarrow \exists q' \in P. q \xrightarrow{a} q' \& (p', q') \in S)\]

This shows that the endodeclaration

\[E_{\text{RS}} : p \mapsto \bigwedge_{a \in A} \bigvee_{p' \in P. p \xrightarrow{a} p'} [a]_{\text{ff}}\]

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

**Proposition 3.3.** The endodeclaration \(E_{\text{RS}}\) characterises the preorder \(\sqsubseteq_{\text{RS}}\).

3.1.3. **Conformance simulation** (Fábregas et al. 2009). Given an \(A\)-labelled transition system \(P = (P, \rightarrow)\) and \(S \subseteq P \times P\), let \(\mathcal{F}_{CS}(S)\) be defined by

\[(p, q) \in \mathcal{F}_{CS}(S) \iff \forall a \in A. \forall q' \in P. (q \xrightarrow{a} q' \Rightarrow \exists p' \in P. p \xrightarrow{a} p' \& (p', q') \in S)\]

This shows that the endodeclaration \(E_{\text{CS}}:\ P \mapsto \bigwedge_{a \in A} \bigvee_{p' \in P. p \xrightarrow{a} p'} \langle a \rangle X_{p'} \land \bigwedge_{a \in A. p \xrightarrow{a}} [a]_{\text{ff}}\)

expresses \(\mathcal{F}_{CS}\). Therefore, Theorem 2.16 gives us the following proposition.

**Proposition 3.3.** The endodeclaration \(E_{\text{CS}}\) characterises the preorder \(\sqsubseteq_{\text{CS}}\).
(1) if \( q \xrightarrow{a} q' \) and \( p \xrightarrow{a} \), then there exists \( p' \in P \) such that \( p \xrightarrow{a} p' \) and \((p', q') \in S \); and

(2) if \( p \xrightarrow{a} \), then \( q \xrightarrow{a} \).

Note that \( \mathcal{F}_{CS} \) is monotone – its greatest fixed point is called the conformance simulation preorder and is denoted by \( \sqsupseteq_{CS} \).

In order to get a \( P \)-indexed declaration that expresses \( \mathcal{F}_{CS} \), we reason as follows:

\[
(p, q) \in \mathcal{F}_{CS}(S) \iff \forall a \in A. \forall q' \in P. (q \xrightarrow{a} q' \\
& \quad \& p \xrightarrow{a} \Rightarrow \exists p' \in P. (p \xrightarrow{a} p' \\
& \quad \& (p', q') \in S))
\]

\[
\& \forall a \in A. (p \xrightarrow{a} \Rightarrow q \xrightarrow{a})
\]

\[
\iff \sigma_{S,q} \models \bigwedge_{a \in A. p \xrightarrow{a}} \bigvee_{p' \in P. p \xrightarrow{a} p'} [a] \bigwedge_{a \in A. p \xrightarrow{a}} \langle a \rangle \Pi
\]

\[
\iff \sigma_{S,q} \models \bigwedge_{a \in A. p \xrightarrow{a}} \bigwedge_{p' \in P. p \xrightarrow{a} p'} ([a] \bigvee_{p' \in P. p \xrightarrow{a} p'} X_{p'})
\]

Thus the endodeclaration

\[
E_{CS} : p \mapsto \bigwedge_{a \in A. p \xrightarrow{a}} \bigwedge_{p' \in P. p \xrightarrow{a} p'} ([a] \bigvee_{p' \in P. p \xrightarrow{a} p'} X_{p'})
\]

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

**Proposition 3.4.** The endodeclaration \( E_{CS} \) characterises the preorder \( \sqsupseteq_{CS} \).

### 3.1.4. Extended simulation (Thomsen 1987).

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

\[
(p, q) \in \mathcal{F}_{ext}(S) \text{ if and only if for every } a \in A:
\]

\[
\text{— if } p \xrightarrow{a} p', \text{ then there exist } q' \in P \text{ and } b \in A \text{ such that } a \sqsubseteq_A b, q \xrightarrow{b} q' \text{ and } (p', q') \in S.
\]

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

In order to get a \( P \)-indexed endodeclaration that expresses \( \mathcal{F}_{ext} \), we reason as follows. For each \( S \subseteq P \times P \) and \( p, q \in P \),

\[
(p, q) \in \mathcal{F}_{ext}(S) \iff \forall a \in A. \forall p' \in P. (p \xrightarrow{a} p' \Rightarrow \exists b \in A. \exists q' \in P. (a \sqsubseteq_A b \\
& \quad \& q \xrightarrow{b} q' \\
& \quad \& (p', q') \in S))
\]

\[
\iff \sigma_{S,q} \models \bigwedge_{a \in A} \bigwedge_{p' \in P. p \xrightarrow{a} p'} \bigwedge_{b \in A. a \sqsubseteq_A b} \langle b \rangle X_{p'}.
\]
Thus the endodeclaration

\[ E_{\text{ext}} : p \mapsto \bigwedge_{a \in A} \bigwedge_{p' \in P} p \xrightarrow{a} p' \lor \bigvee (b) X_{p'} \]

expresses \( \mathcal{F}_{\text{ext}} \), and, by Theorem 2.16, we have the following proposition.

**Proposition 3.5.** The endodeclaration \( E_{\text{ext}} \) characterises the preorder \( \sqsubseteq_{\text{ext}} \).

### 3.2. Types of bisimilarity

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

#### 3.2.1. Strong bisimulation (Park 1981; Milner 1989)

Given an \( A \)-labelled transition system \( P = (P, \rightarrow) \) and a relation \( S \subseteq P \times P \), let \( F_{\text{bisim}}(S) \) be defined by:

\[
(p, q) \in F_{\text{bisim}}(S) \text{ if and only if for every } a \in A:
\]

1. If \( p \xrightarrow{a} p' \), then there exists some \( q' \in P \) such that \( q \xrightarrow{a} q' \) and \( (p', q') \in S \); and
2. If \( q \xrightarrow{a} q' \), then there exists some \( p' \in P \) such that \( p \xrightarrow{a} p' \) and \( (p', q') \in S \).

\( F_{\text{bisim}} \) is a monotone endofunction on \( \mathcal{P}(P \times P) \) – its greatest fixed point is called the bisimulation equivalence and is denoted by \( \sim_{\text{bisim}} \).

In order to find a declaration that expresses \( F_{\text{bisim}} \), we observe in the following lemma that if we have declarations expressing the monotone endofunctions \( F_1 \) and \( F_2 \), we can obtain one expressing \( S \mapsto F_1(S) \cap F_2(S) \). This generalises to arbitrary intersections as follows.

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

Note that

\[
F_{\text{bisim}} : S \mapsto F_{\text{sim}}(S) \cap F_{\text{opsim}}(S)
\]

and that since \( E_{\text{sim}} \) and \( E_{\text{opsim}} \) express \( F_{\text{sim}} \) and \( F_{\text{opsim}} \), respectively, we have \( p \mapsto E_{\text{sim}}(p) \land E_{\text{opsim}}(p) \) expresses \( S \mapsto F_{\text{sim}}(S) \cap F_{\text{opsim}}(S) \), so the endodeclaration for \( \mathcal{L}(P) \) given by

\[
E_{\text{bisim}} : p \mapsto \left( \bigwedge_{a \in A} \bigwedge_{p' \in P} \langle a \rangle X_{p'} \right) \land \left( \bigwedge_{a \in A} \bigwedge_{p' \in P} \langle a \rangle X_{p'} \right)
\]

expresses \( F_{\text{bisim}} \). The following proposition now follows from Theorem 2.16.

**Proposition 3.7.** The endodeclaration \( E_{\text{bisim}} \) characterises the equivalence \( \sim_{\text{bisim}} \).

The endodeclaration \( E_{\text{bisim}} \) is exactly the one proposed in Ingolfsdottir et al. (1987).
3.2.2. Weak bisimulation (Milner 1989). Let \( P = (P, \rightarrow) \) be an \( A \)-labelled transition system with one label \( \tau \in A \) to be viewed as a silent step. We derive a family \( \{ \overset{a}{\Rightarrow} \}_{a \in A} \) of transition relations over \( P \) as follows:

\[ \overset{\tau}{\Rightarrow} \] is the reflexive transitive closure \( \overset{\tau}{\Rightarrow}^* \) of \( \overset{\tau}{\Rightarrow} \); and

\[ \overset{a}{\Rightarrow} \] is the composition \( \overset{\tau}{\Rightarrow} \circ \overset{a}{\Rightarrow} \circ \overset{\tau}{\Rightarrow} \), for \( a \neq \tau \).

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 \( \llbracket a \rrbracket \) for the derived relations \( \overset{a}{\Rightarrow} \), with \( a \in A \). Thus the semantics is:

\[ \sigma, p \models \langle\langle a\rangle\rangle F_1 \iff \sigma, p' \models F_1 \text{ for some } p' \text{ for which } p \overset{a}{\Rightarrow} p' \]

\[ \sigma, p \models \llbracket a \rrbracket F_1 \iff \sigma, p' \models F_1 \text{ for all } p' \text{ for which } p \overset{a}{\Rightarrow} p'. \]

Let \( F_{wbsm}(S) \) be defined by

\[(p, q) \in F_{wbsm}(S) \text{ if and only if for every } a \in A:\]

1. if \( p \overset{a}{\Rightarrow} p' \), then there exists \( q' \in P \) such that \( q \overset{\tau}{\Rightarrow} q' \) and \((p', q') \in S\); and
2. if \( q \overset{\tau}{\Rightarrow} q' \), then there exists \( p' \in P \) such that \( p \overset{a}{\Rightarrow} p' \) and \((p', q') \in S\).

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

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

\[ E_{wbsm} : p \mapsto \left( \bigwedge_{a \in A} \bigwedge_{p' \in P, p \overset{a}{\Rightarrow} p'} \langle\langle a\rangle\rangle X_{p'} \right) \land \left( \bigwedge_{a \in A} \bigvee_{p' \in P, p \overset{a}{\Rightarrow} p'} \llbracket a \rrbracket X_{p'} \right) \]

expresses \( F_{wbsm} \), and hence, by Theorem 2.16, we have the following proposition.

**Proposition 3.8.** The endodeclaration \( E_{wbsm} \) characterises the equivalence \( \sim_{wbsm} \).

Note that since \( q \overset{\tau}{\Rightarrow} q \) holds for each \( q \), unlike in the case for \( E_{bisim} \), the formula for \( E_{wbsm} \) can never have conjuncts of the form \( \llbracket [\tau] \rrbracket \).

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 De Nicola et al. (1990) that allows for several possible past states. The semantics introduced in De Nicola et al. (1990) 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 next section.

Given \( P = (P, \rightarrow) \) and \( S \subseteq P \times P \), let \( F_{bfb}(S) \) be defined by

\[ \overset{\tau}{\Rightarrow} \text{ is the composition } \overset{\tau}{\Rightarrow} \circ \overset{a}{\Rightarrow} \circ \overset{\tau}{\Rightarrow} \], for \( a \neq \tau \).

\[ \overset{\tau}{\Rightarrow} \] is the reflexive transitive closure \( \overset{\tau}{\Rightarrow}^* \) of \( \overset{\tau}{\Rightarrow} \).

\[ \sim_{wbsm} \text{ is the composition } \sim_{wbsm} \circ \overset{a}{\Rightarrow} \circ \sim_{wbsm} \text{, for } a \neq \tau \].
(p, q) ∈ \mathcal{F}_{bfb}(S) if and only if (p, q) ∈ \mathcal{F}_{bisim}(S) and for every a ∈ A:

1. \forall p' ∈ P. p' \xrightarrow{a} p ⇒ \exists q' ∈ P. q' \xrightarrow{a} q and (p', q') ∈ S; and
2. \forall q' ∈ P. q' \xrightarrow{a} q ⇒ \exists p' ∈ P. p' \xrightarrow{a} p and (p', q') ∈ S.

\mathcal{F}_{bfb} is a monotone endofunction on \mathcal{P}(P × P) – its greatest fixed point is called the back and forth bisimulation equivalence and is denoted by \sim_{bfb}.

To express this behaviour in the logical language considered so far, we add to it two operators \langle a \rangle and \lfloor a \rfloor for every a ∈ A. The semantics for these is given by:

\[ \sigma, p \models \langle a \rangle F \iff \sigma, p' \models F \text{ for some } p' \text{ for which } p' \xrightarrow{a} p; \text{ and} \]
\[ \sigma, p \models \lfloor a \rfloor F \iff \sigma, p' \models F \text{ for all } p' \text{ for which } p' \xrightarrow{a} p. \]

\([F]\) is clearly monotone for each F in the extended language.

In order to find an endodeclaration that expresses \sim_{bfb}, we reason as we did for finding an endodeclaration that express \sim_{bisim}, and observe the following equivalence:

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

Then the endodeclaration given by

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

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

**Proposition 3.9.** The endodeclaration \( E_{bfb} \) characterises the equivalence \sim_{bfb}.

### 3.2.4. Back and forth bisimulation with indistinguishable states (Dechesne et al. 2007)

In this section we consider a version of the back and forth bisimulation from the previous section where some of the states are considered to be indistinguishable to some external agents. To this end, we augment our notion of labelled transition systems with a set \mathcal{I} of identities (or agents) and a family of equivalence relations \{ i \cdot \}_{i ∈ \mathcal{I}}. Intuitively, \( p \cdot i \cdot q \) means that agent i cannot distinguish p from q. Such a structure is called an annotated labelled transition system (Dechesne et al. 2007).

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

(p, q) ∈ \mathcal{F}_{bfbid}(S) if and only if (p, q) ∈ \mathcal{F}_{bfb}(S) and for every a ∈ A and i ∈ \mathcal{I}:

1. \forall p' ∈ P. p \cdot i \cdot p' ⇒ \exists q' ∈ P. q \cdot i \cdot q' and (p', q') ∈ S; and
2. \forall q' ∈ P. q \cdot i \cdot q' ⇒ \exists p' ∈ P. p \cdot i \cdot p' and (p', q') ∈ S.
We denote the greatest fixed point of $\mathcal{F}_{bfid}$ by $\sim_{bfid}$. 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 $\left[ i \right]$ for each $i \in I$. The semantics for these operators is given by

$$\sigma, p \models \langle i \rangle F_1 \iff \sigma, p' \models F_1 \text{ for some } p' \text{ for which } p \cdot \cdot \cdot p';$$

$$\sigma, p \models \left[ i \right] F_1 \iff \sigma, p' \models F_1 \text{ for all } p' \text{ for which } p \cdot \cdot \cdot p'.$$

$[F]$ is clearly monotone for each $F$ in the extended language.

In order to find an endodeclaration that expresses $\sim_{bfid}$, we reason as we did for finding an endodeclarations that express $\sim_{bf}$ and $\sim_{bisim}$, and observe the following equivalence:

$$(p, q) \in \mathcal{F}_{bf}(S) \iff \sigma_S, q \models E_{bf}(p) \land \left( \bigwedge_{i \in I} \bigwedge_{p' \in P} \langle i \rangle X_{p'} \right) \land \left( \bigwedge_{i \in I} \bigvee_{p' \in P} X_{p'} \right).$$

Hence the endodeclaration given by

$$E_{bfid} : p \mapsto E_{bf}(p) \land \left( \bigwedge_{i \in I} \bigwedge_{p' \in P} \langle i \rangle X_{p'} \right) \land \left( \bigwedge_{i \in I} \bigvee_{p' \in P} X_{p'} \right)$$

expresses $\sim_{bfid}$, so we have the following proposition.

**Proposition 3.10.** The endodeclaration $E_{bfid}$ characterises the equivalence $\sim_{bfid}$.

As an immediate consequence of the existence of this characteristic formula, we obtain a behavioural characterisation of the equivalence over states in an annotated labelled transition system induced by the epistemic logic studied in Dechesne et al. (2007) – more precisely, the logic we study in this section may be seen as the positive version of the one studied in Dechesne et al. (2007), where we use the modal operator $\left[ i \right]$ instead of $K_i$, which is read ‘agent $i$ knows’, and its dual. This solves a problem that was left open in the aforementioned reference.

**Theorem 3.11.** Let $p, q \in P$. Then $p \sim_{bfid} q$ 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 an argument along standard lines. The right to left implication is an immediate consequence of Proposition 3.10.

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. We write $p \Downarrow$ to indicate that $p$ may diverge. We write $p \uparrow$ to indicate that $p$ may diverge. We write $p \Downarrow$ to indicate that $p$ may diverge. We write $p \Downarrow$ to indicate that $p$ may diverge. We write $p \Downarrow$ to indicate that $p$ may diverge. We write $p \Downarrow$ to indicate that $p$ may diverge.

**Definition 3.12.** An $A$-labelled transition system with divergence consists of:

- a set $P$ of processes;
- a transition relation $\rightarrow \subseteq P \times P$;
- a predicate $\Downarrow \subseteq P$; and
We write \( p \vdash \) for \( p \in \dagger \) and \( p \nvdash \) for \( p \notin \dagger \).

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

\[
F ::= X_i \quad (i \in I) \\
\neg \bigwedge_{k \in K} F_k \quad (K \text{ a finite set}) \\
\bigvee_{k \in K} F_k \quad (K \text{ a finite set}) \\
\dagger F \\
\narrow F \\
\langle a \rangle F \quad (a \in A) \\
\langle a \rangle \neg F \quad (a \in A) \\
\langle a \rangle \neg \dagger F \quad (a \in A) \\
\langle a \rangle \neg \narrow F \quad (a \in A) \\
[a] F \quad (a \in A) \\
[a] \neg F \quad (a \in A) \\
[a] \neg \dagger F \quad (a \in A) \\
[a] \neg \narrow F \quad (a \in A).
\]

Let \( P = (P, \rightarrow, \dagger) \) be an \( A \)-labelled transition system with divergence. We then inductively define a satisfaction relation \( \models \) from \( \mathcal{P}(P)^I \times P \) to \( \mathcal{L}(I) \) as follows, where \( \sigma \in \mathcal{P}(P)^I \) and \( p \in P \):

\[
\begin{align*}
\sigma, p & \models X_i \quad \Leftrightarrow \quad p \in \sigma(i) \\
\sigma, p & \models \bigwedge_{k \in K} F_k \quad \Leftrightarrow \quad \forall k \in K. \sigma, p \models F_k \\
\sigma, p & \models \bigvee_{k \in K} F_k \quad \Leftrightarrow \quad \exists k \in K. \sigma, p \models F_k \\
\sigma, p & \models \dagger \quad \Leftrightarrow \quad p \dagger \\
\sigma, p & \models \narrow \quad \Leftrightarrow \quad p \narrow \\
\sigma, p & \models \langle a \rangle F \quad \Leftrightarrow \quad \text{there is } p' \in P \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p & \models \langle a \rangle \neg F \quad \Leftrightarrow \quad p \dagger \text{ and there is } p' \in P \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p & \models \langle a \rangle \neg \dagger F \quad \Leftrightarrow \quad p \narrow \text{ and there is } p' \in P \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p & \models \langle a \rangle \neg \narrow F \quad \Leftrightarrow \quad p \dagger \text{ or there is } p' \in P \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F \\
\sigma, p & \models [a] F \quad \Leftrightarrow \quad \text{for all } p' \in P \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F \\
\sigma, p & \models [a] \neg F \quad \Leftrightarrow \quad p \dagger \text{ and for all } p' \in P \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F \\
\sigma, p & \models [a] \neg \dagger F \quad \Leftrightarrow \quad p \narrow \text{ and for all } p' \in P \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F \\
\sigma, p & \models [a] \neg \narrow F \quad \Leftrightarrow \quad p \dagger \text{ or for all } p' \in P \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F \\
\sigma, p & \models [a] F \quad \Leftrightarrow \quad p \narrow \text{ or for all } p' \in P \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F.
\end{align*}
\]

Occasionally, we want to consider transition systems in which other erroneous behaviours in addition to divergence are possible, such as deadlock or crash (Aceto and Hennessy 1992). In general, we have a finite set \( E \) containing all these erroneous behaviours.
Definition 3.13. Let $E$ be a finite set. An $A$-labelled transition system with $E$-errors consists of:

— a finite set $P$ of processes;
— a transition relation $\longrightarrow \subseteq P \times A \times P$; and
— an error relation $\frac{1}{E} \subseteq P \times E$.

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

We define the Hennessy–Milner logic with $E$-errors as follows. For a set $I$, we write $L(I)$ for the set of formulae given by

$$F ::= X_i \quad (i \in I)$$
$$\bigwedge_{k \in K} F_k \quad (K \text{ a finite set})$$
$$\bigvee_{k \in K} F_k \quad (K \text{ a finite set})$$
$$\nabla D \quad (D \subseteq \mathcal{P}(E))$$
$$\langle a \rangle D F \quad (a \in A \text{ and } D, D' \text{ disjoint subsets of } \mathcal{P}(E))$$
$$[a] D F \quad (a \in A \text{ and } D, D' \text{ disjoint subsets of } \mathcal{P}(E))$$

Let $P = (P, \longrightarrow, \frac{1}{E})$ be an $A$-labelled transition system with $E$-errors. We then inductively define a relation $\models$ from $\mathcal{P}(P)^I \times P$ to $L(I)$ as follows, where $\sigma \in \mathcal{P}(P)^I$ and $p \in P$, and we write $\text{Errors}(p) = \{e \in E \mid p \frac{1}{E} e\}$:

$$\sigma, p \models X_i \iff p \in \sigma(i)$$
$$\sigma, p \models \bigwedge_{k \in K} F_k \iff \sigma, p \models F_k \text{ for all } k \in K$$
$$\sigma, p \models \bigvee_{k \in K} F_k \iff \sigma, p \models F_k \text{ for some } k \in K$$
$$\sigma, p \models \nabla D \iff \text{Errors}(p) \in D$$
$$\sigma, p \models \langle a \rangle D F \iff \text{Errors}(p) \in D, \text{ or}$$
$$\text{Errors}(p) \notin D' \text{ and there is } p' \in P \text{ such that } p \xrightarrow{a} p' \text{ and } \sigma, p' \models F$$
$$\sigma, p \models [a] D F \iff \text{Errors}(p) \in D, \text{ or}$$
$$\text{Errors}(p) \notin D' \text{ and for all } p' \in P \text{ such that } p \xrightarrow{a} p', \text{ we have } \sigma, p' \models F.$$
Dualisation is defined as follows:

\[ \overline{X_i} = X_i \]
\[ \bigwedge_{k \in K} F_k = \bigvee_{k \in K} \overline{F_k} \]
\[ \bigvee_{k \in K} F_k = \bigwedge_{k \in K} \overline{F_k} \]
\[ \overline{\bigvee F} = \bigvee (\mathcal{P}(E)) \setminus \mathcal{D} \]
\[ \overline{[a]_E F} = [a]_E \overline{F} \]
\[ [a]_E \overline{f} = \langle a \rangle_E \overline{F} \]

All the previously mentioned results about substitution and about characteristic formulae adapt without difficulty to the Hennessy–Milner logic with E-errors – and the same holds true for the results on dualisation and co-characteristic formulae to be presented in Section 5.

3.4. Divergence and simulation

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

- \((p, p') \in \mathcal{F}_{\text{lower}}(S)\) when for every \( a \in A \) and \( q \in P \) such that \( p \xrightarrow{a} q \), there exists some \( q' \in P \) such that \( p' \xrightarrow{a} q' \) and \((q, q') \in S\) – this is the same as \( \mathcal{F}_{\text{sim}}(S) \), for the transition system \((P, \rightarrow)\).
- \((p, p') \in \mathcal{F}_{\text{incl}}(S)\) when both the following hold:
  1. for every \( a \in A \) and \( q \in P \) such that \( p \xrightarrow{a} q \), there exists some \( q' \in P \) such that \( p' \xrightarrow{a} q' \) and \((q, q') \in S\); and
  2. if \( p \hat{\cdot} \), then \( p' \hat{\cdot} \).
- \((p, p') \in \mathcal{F}_{\text{upper}}(S)\) when \( p \hat{\cdot} \) implies that both the following hold:
  1. \( p' \hat{\cdot} \); and
  2. for all \( a \in A \) and \( q' \in P \) such that \( p' \xrightarrow{a} q' \), there exists some \( q \in P \) such that \( p \xrightarrow{a} q \) and \((q, q') \in S\).
- \((p, p') \in \mathcal{F}_{\text{smash}}(S)\) when \( p \hat{\cdot} \) implies that all the following hold:
  1. \( p' \hat{\cdot} \);
  2. for all \( a \in A \) and \( q \in P \) such that \( p \xrightarrow{a} q \), there exists some \( q' \in P \) such that \( p' \xrightarrow{a} q' \) and \((q, q') \in S\); and
  3. for all \( a \in A \) and \( q' \in P \) such that \( p' \xrightarrow{a} q' \), there exists some \( q \in P \) such that \( p \xrightarrow{a} q \) and \((q, q') \in S\).
- \( \mathcal{F}_{\text{convex}}(S) = \mathcal{F}_{\text{lower}}(S) \cap \mathcal{F}_{\text{upper}}(S) \).

A lower simulation is a post-fixed point for \( \mathcal{F}_{\text{lower}} \), and we likewise define inclusion simulation, upper simulation, smash simulation and convex simulation (also known as a
prebisimulation or partial bisimulation) – see Lassen (1998), Milner (1981), Moran (1998), Pitcher (2001), Stirling (1987) and Ulidowski (1992).

For each of these functions, we can find an endodeclaration expressing it in Hennessy–Milner logic with divergence. For example, $F_{\text{lower}}$ is expressed (just like $F_{\text{sim}}$ in Section 3.1.1) by

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

and $F_{\text{upper}}$ is expressed by

$$E_{\text{upper}} : p \mapsto \tilde{\emptyset} \land \bigwedge_{a \in A} \bigvee_{q \in P, p \xrightarrow{a} q} X_q \quad (p \emptyset)$$

$$\quad \mapsto \top \quad (p \tilde{\emptyset}).$$

If $A$ is non-empty, we can equivalently write this endodeclaration as follows:

$$E_{\text{upper}} : p \mapsto \bigwedge_{a \in A} \bigvee_{q \in P, p \xrightarrow{a} q} X_q \quad (p \emptyset)$$

$$\quad \mapsto \top \quad (p \tilde{\emptyset}).$$

Lemma 3.6 then tells us that $F_{\text{convex}}$ is expressed by

$$E_{\text{convex}} : p \mapsto \bigwedge_{a \in A} \bigwedge_{q \in P, p \xrightarrow{a} q} \langle a \rangle X_q$$

$$\quad \land \bigwedge_{a \in A} \bigvee_{q \in P, p \xrightarrow{a} q} X_q \quad (p \emptyset)$$

$$\quad \mapsto \bigwedge_{a \in A} \bigwedge_{q \in P, p \xrightarrow{a} q} \langle a \rangle X_q \quad (p \tilde{\emptyset}).$$

We can treat these examples, and others such as those in Aceto and Hennessy (1992), in a systematic manner by using the following concepts, which are taken from Levy (2009).

**Notation for disjoint union.** If $\{A_i\}_{i \in I}$ is a family of sets, 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

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

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

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

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

$$\mathcal{P}(\mathbb{B} + E) \times (\mathcal{P}(\mathbb{B} + E))^\mathbb{B} \longrightarrow \mathcal{P}(\mathbb{B} + E)$$

$$K + D, \quad f \quad \mapsto \bigcup_{b \in K} f(b) \cup \{\text{inr } e \mid e \in D\}.$$ 

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

Levy (2009) showed 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 given in Figures 1–3, which are taken from Levy (2009).
There are many different notions of simulation on an $A$-labelled transition system $P = (P, \rightarrow, \xi)$. In order for $p'$ to simulate $p$, we need three kinds of conditions:

1. The errors of $p$ and the errors of $p'$ must be suitably related. For example, in smash simulation, if $p$ does not diverge, then neither does $p'$.
2. In some circumstances (depending on the errors), whatever transition $p$ can do can also be done by $p'$, up to simulation. For example, in smash simulation, this is the case if $p$ does not diverge.
3. In some circumstances (depending on the errors), whatever transition $p'$ can do can also be done by $p$, up to simulation. For example, in smash simulation, this is the case if $p$ does not diverge.

The precise conditions are determined by an NDBP in the following manner.

**Definition 3.15.**

Let $E$ be a finite set, and let $\sqsubseteq$ be a NDBP for $E$-errors. For any relation $S \subseteq P \times P$, we define a relation $\mathcal{F}_{\sqsubseteq}(S)$ as follows:

$$(p, p') \mathcal{F}_{\sqsubseteq}(S)$$

when:

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

Note that each of the five kinds of simulation from the start of Section 3.4 arises from the corresponding NDBP shown in Figure 2.
Fig. 2. All the NDBPs for divergence (d)
Theorem 3.16. Let $E$ be a finite set and $\sqsubseteq$ be an NDBP for $E$-errors. The $P$-indexed endodeclaration

$$
E_{\sqsubseteq} : p \mapsto \nabla \{ C \subseteq E \mid \text{Errors}(p) \sqsubseteq [t] + C \} \land \bigwedge_{a \in A} \bigwedge_{q \in P, p \stackrel{a}{\rightarrow} q} \langle a \rangle \{ C \subseteq E \mid \text{Errors}(p) \sqsubseteq [t] + C \} X_q
$$

$$
\land \bigwedge_{a \in A} \bigvee_{q \in P, p \stackrel{a}{\rightarrow} q} \langle a \rangle \{ C \subseteq E \mid \text{Errors}(p) \sqsubseteq [t] + C \} X_q \land \bigwedge_{a \in A} \bigvee_{q \in P, p \stackrel{a}{\rightarrow} q} \langle a \rangle \{ C \subseteq E \mid \text{Errors}(p) \sqsubseteq [t] + C \} X_q
$$

expresses $F_{\sqsubseteq}$.

Proof. The proof is straightforward. \hfill \square

It follows that $E_{\sqsubseteq}$ characterises $\sqsubseteq$ 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 Levy (2009).

4. Extensions of the approach

All the applications of the general theory developed so far in this paper have dealt with relations defined as the greatest fixed points of monotone endofunctions over the complete lattice of binary relations over $P$. In this setting, Theorem 2.16 has allowed us to characterise those relations logically by exhibiting an endodeclaration that expresses the relevant endofunction. Sometimes, however, it is not easy to use Theorem 2.16 to yield characteristic-formula constructions for behavioural relations. In this section, we present two such examples of relations, namely, mutual similarity (that is, simulation equivalence) and the 2-nested simulation preorder. We also extend our approach and provide the theoretical tools needed to develop characteristic formulae for them. The notion of a 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 \(E \in \mathcal{L}(I)'\) is an endodeclaration for some set \(I\).
2. A formula with endodeclaration \((F, E)\) *characterises* process \(p \in P\) up to a relation \(S \subseteq P \times P\) if

\[(p, q) \in S \text{ iff } \nu[[E]], q \models [[F]]\]

or, equivalently, if

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

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

— \(E\) characterises \(S\).
— For each \(p \in P\), the formula with endodeclaration \((X_p, E)\) characterises \(p\) up to \(S\).

4.1.1. Substitution. We will now 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 \(F\), which is defined by induction on \(F\) as follows:

\[X_i[D] = D(i)\]
\[(\bigwedge_{k \in K} F_k)[D] = \bigwedge_{k \in K} (F_k[D])\]
\[(\bigvee_{k \in K} F_k)[D] = \bigvee_{k \in K} (F_k[D])\]
\[(\langle a \rangle F)[D] = \langle a \rangle (F[D])\]
\([a F][D] = [a F][D].\]

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:

\[
\begin{array}{ccc}
\mathcal{P}(P)^I & \xrightarrow{[D]} & \mathcal{P}(P)^I \\
|[F[D]| & \downarrow & |[F]| \\
\mathcal{P}(P) & & \mathcal{P}(P)
\end{array}
\]

**Proof.** All the statements are proved by induction on \(F\).
4.2. Adding variables

It is useful to note that a characteristic formula is still a characteristic formula 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.

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

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

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

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

Then $\nu H$ exists if and only if $\nu g \nu f$ does, in which case

$$\nu H = (\nu f, \nu g \nu f).$$

The above result, which is proved in Appendix A, is a special case of Bekic’s lemma.

We will now formally define the extension of a declaration.

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

$$m(i) \mapsto E(i)[i \mapsto X_{m(i)}] \quad (i \in I)$$

$$j \mapsto D(j) \quad (j \in J \setminus \text{range}(m)).$$

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

— $(F, E)$ characterises $p$ up to $S$.
— $(F[i \mapsto X_{m(i)}, EmD])$ characterises $p$ up to $S$.

**Proof.** We will use the abbreviation $I' = J \setminus \text{range}(m)$. Let $\theta : \mathcal{P}(P)^I \times \mathcal{P}(P)^{I'} \to \mathcal{P}(P)^I$ be the isomorphism mapping $(\sigma, \sigma')$ to

$$m(i) \mapsto \sigma(i) \quad (i \in I)$$

$$j \mapsto \sigma'(j) \quad (j \in I')$$

and define the endofunction $G_{\nu [E]}$ on $\mathcal{P}(P)^{I'}$ to map $\sigma$ to $[D]\theta(\nu [E], \sigma)$, and the endofunction $H$ on $\mathcal{P}(P)^I \times \mathcal{P}(P)^{I'}$ to map $(\sigma, \sigma')$ to $(\nu [E] \sigma, [D]\theta(\sigma, \sigma'))$. Since $[D]$ is monotone, so is $G_{\nu [E]}$, and hence $\nu g_{\nu [E]}$ exists. Then, by Lemma 4.4, we have $\nu H = (\nu [E], \nu g_{\nu [E]}).$
By the definitions, the following diagram commutes:

\[
\begin{array}{ccc}
\mathcal{P}(P)^I \times \mathcal{P}(P)^I' & \xrightarrow{H} & \mathcal{P}(P)^I \times \mathcal{P}(P)^I' \\
\theta \downarrow & & \theta \downarrow \\
\mathcal{P}(P)^I & \xrightarrow{\llbracket E_mD \rrbracket} & \mathcal{P}(P)^J \\
\end{array}
\tag{2}
\]

This is because

\[
\theta(H(\sigma, \sigma')) = \theta(\llbracket E \rrbracket \sigma, \llbracket D \rrbracket \theta(\sigma, \sigma'))
\]
is the function such that

\[
m(i) \mapsto (\llbracket E \rrbracket \sigma)(i) \quad (i \in I) \\
j \mapsto \llbracket D(j) \rrbracket \theta(\sigma, \sigma') \quad (j \in I').
\]

and \(\llbracket E_mD \rrbracket \theta(\sigma, \sigma')\) is the function such that

\[
m(i) \mapsto \llbracket E_mD(m(i)) \rrbracket \theta(\sigma, \sigma') = (\llbracket E(i)[i \mapsto X_{m(i)}] \rrbracket \theta(\sigma, \sigma') = (\llbracket E \rrbracket \sigma)(i) \quad (i \in I) \\
j \mapsto \llbracket E_mD(j) \rrbracket \theta(\sigma, \sigma') = \llbracket D(j) \rrbracket \theta(\sigma, \sigma') \quad (j \in I').
\]

Thus Theorem 2.3(2) gives us \(\nu \llbracket E_mD \rrbracket = \theta(\nu \llbracket E \rrbracket, \nu g \nu \llbracket E \rrbracket)\). We calculate

\[
\llbracket F[i \mapsto X_{m(i)}] \rrbracket(\nu \llbracket E_mD \rrbracket) = \llbracket F \rrbracket(i \mapsto \llbracket X_{m(i)} \rrbracket(\nu \llbracket E_mD \rrbracket)) = \llbracket F \rrbracket(i \mapsto (\nu \llbracket E_mD \rrbracket)m(i)) = \llbracket F \rrbracket(i \mapsto (\theta(\nu \llbracket E \rrbracket, \nu g \nu \llbracket E \rrbracket))m(i)) = \llbracket F \rrbracket(i \mapsto (\nu \llbracket E \rrbracket)i) = \llbracket F \rrbracket(\nu \llbracket E \rrbracket),
\]

which gives the required result.

We use the following notions to illustrate the use of Lemma 4.6.

**Definition 4.7.** Let \(p \in P\) be a process.

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

Let \(p \in P\) and \(E\) be a \(P\)-indexed declaration. We write \(E \upharpoonright \text{reach}(p)\) for the restriction of \(E\) to a \(\text{reach}(p)\)-indexed declaration. As an 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 \(\sqsubseteq_{\text{sim}}\). To see why this is advantageous, suppose \(p\) is image countable. Then, since \(A\) is assumed to be 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 \(P\) 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 both $S_1$ and $S_2$, then we can find them up to $S_1 \cap S_2$. This generalises to arbitrary intersections as follows.

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

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

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

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

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

$$\text{inl } p \mapsto \bigwedge_{a \in A} \bigwedge_{q \in P, p \xrightarrow{a} q} \langle a \rangle X_{\text{inl } q}$$

$$\text{inr } p \mapsto \bigwedge_{a \in A} \bigvee_{q, p \xrightarrow{a} q} X_{\text{inr } q}.$$

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

4.4. Nested simulation (Groote and Vaandrager 1992)

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

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

**Theorem 4.10.** Let $P = (P, \rightarrow)$ be an A-labelled transition systems, $F$ be a monotone endofunction on $\mathcal{P}(P \times P)$ and $R \subseteq P \times P$ be a relation. Let $E$ be an $I$-indexed endodeclaration and $D : P \rightarrow \mathcal{L}(I)$ be a declaration such that, for each $p \in P$, the formula with endodeclaration $(D(p), E)$ characterises $p$ up to $R$. Let $E'$ be a $P$-indexed
endodeclaration expressing $\mathcal{F}$. Let $E''$ be the $I + P$ indexed declaration
\[
\begin{align*}
\text{inl } i & \mapsto E(i)[i' \mapsto X_{\text{inl } i}]
\end{align*}
\]
\[
\begin{align*}
\text{inr } p & \mapsto F(p)[i' \mapsto X_{\text{inl } i}] \land E'(p)[p' \mapsto X_{\text{inr } p}].
\end{align*}
\]

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

**Proof.** Let $h : \mathcal{P}(P)^I \to \mathcal{P}(P \times P)$ be the monotone function mapping each $\sigma$ to
\[
\{(q, q') \in P \times P \mid q' \in [D(q)]_{\sigma}\}.
\]
Let $H$ be the monotone endofunction on $\mathcal{P}(P)^I \times \mathcal{P}(P \times P)$ mapping $(\sigma, S)$ to
\[
([E]_{\sigma}, h(\sigma) \cap \mathcal{F}(S)).
\]
Lemma 4.4 gives us
\[
vH = (v([E]), v(h(v([E]) \cap \mathcal{F})).
\]
Now
\[
(p, p') \in h(v([E])) \iff p' \in [D(p)](v([E])) \iff (p, p') \in R
\]
(by definition)

(since $(D(p), E)$ characterises $p$ up to $R$).

Therefore, $h(v([E])) = R$, so $vH = (v([E]), v(R \cap \mathcal{F}))$. By calculation, the following diagram commutes:
\[
\begin{array}{ccc}
\mathcal{P}(P)^I \times \mathcal{P}(P \times P) & \xrightarrow{\tau} & \mathcal{P}(P)^I + P \\
H \downarrow & & \downarrow [E]'' \\
\mathcal{P}(P)^I \times \mathcal{P}(P \times P) & \xrightarrow{\tau} & \mathcal{P}(P)^I + P
\end{array}
\]
where $\tau$ is the isomorphism mapping $(\sigma, S)$ to the environment
\[
\begin{align*}
\text{inl } i & \mapsto \sigma i \\
\text{inr } p & \mapsto \{p' \in P \mid (p, p') \in S\}.
\end{align*}
\]
So Lemma 2.3(2) gives $\tau vH = v([E''])$. Hence
\[
\begin{align*}
\llbracket X_{\text{inr } p} \rrbracket(v([E''])) & = (v([E'']))(\text{inr } p) \\
& = (\tau vH)(\text{inr } p) \\
& = (\tau(v([E]), v(R \cap \mathcal{F}))(\text{inr } p) \\
& = \{p' \in P \mid (p, p') \in v(R \cap \mathcal{F})\},
\end{align*}
\]
as required. □

It is easy to see that Theorem 2.16 corresponds to the special case of Theorem 4.10 given by
\[
R = P \times P \\
I = \emptyset \\
F : p \mapsto \pi.
\]
We can now put together the characteristic formulae for \( \sqsubseteq_{\text{opsim}} \) and the declaration expressing \( \mathcal{F}_{\text{sim}} \), which were both given in Section 3.1.1. Let \( E_{2\text{sim}} \) be the \( P + P \) indexed declaration

\[
\begin{align*}
\text{inl } p &\mapsto \bigwedge_{q \in A} \left[ a \right] \bigvee_{q \in P, \ p \xrightarrow{a} q} X_{\text{inl } q} \\
\text{inr } p &\mapsto X_{\text{inl } p} \land \bigwedge_{a \in A} \bigwedge_{q \in P, \ p \xrightarrow{a} q} \langle a \rangle X_{\text{inr } q}.
\end{align*}
\]

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

5. Co-characteristic formulae

5.1. Dualisation

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

\[
\begin{align*}
\overline{X_i} &= X_i \\
\bigwedge_{k \in K} F_k &= \bigvee_{k \in K} \overline{F_k} \\
\bigvee_{k \in K} F_k &= \bigwedge_{k \in K} \overline{F_k} \\
\langle a \rangle \overline{F} &= [a] \overline{F} \\
[a] \overline{F} &= \langle a \rangle \overline{F}.
\end{align*}
\]

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

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

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

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

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

commutes. Then the following statements hold:

1. If \( f(x) = x \), then \( g(\psi(x)) = \psi(x) \) (\( \psi \) maps fixed points for \( f \) into fixed points for \( g \)).
2. If \( \nu f \) exists, then \( \mu g \) exists and \( \psi(\nu f) = \mu g \) (\( \psi \) maps the greatest fixed point for \( f \) into the least fixed point for \( g \)).
If $\mu f$ exists, then $\nu g$ exists and $\psi(\mu f) = \nu g$ ($\psi$ 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 $B$.

Lemma 5.3. Dualisation satisfies the following properties:

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

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

Proof. Both statements can be shown by induction on $F$.

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 $\sigma$ and each $q \in \mathbf{P}$,

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

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

1. An endodeclaration $E$ co-characterises $S$ if and only if

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

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

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

or, equivalently, if

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

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

— $E$ co-characterises $S$.
— For each $p \in \mathbf{P}$, the formula with endodeclaration $(X_p, E)$ co-characterises $p$ 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 dualise it.

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

1. Let $E$ be an endodeclaration for $L(\mathbf{P})$. Then the following statements are equivalent:
— \( E \) characterises \( S^{-1} \).
— \( \overline{E} \) co-characterises \( S \).

(2) Let \( p \in P \) and \((F, E)\) be a formula with endodeclaration. Then the following statements are equivalent:
— \((F, E)\) characterises \( p \) up to \( S^{-1} \).
— \((\overline{F}, E)\) co-characterises \( p \) up to \( S \).

**Proof.** We will just prove part (2) as part (1) is just a special case.

Lemma 5.3(2) tells us that
\[
\mathcal{P}(\mathcal{P})(\chi_{\mathcal{I}} \downarrow \downarrow [E]) \rightarrow \mathcal{P}(\mathcal{P})(\chi_{\mathcal{I}} \downarrow \downarrow [E]) \rightarrow \mathcal{P}(\mathcal{P})(\chi_{\mathcal{I}} \downarrow \downarrow [E])
\]
commutes, and Theorem 5.2(2) tells us that \( \mu[[E]] = \chi(\nu[[E]]) \).

We now apply Lemma 5.3(2) again to obtain
\[
[[\overline{F}](\mu[[E]]) = \chi([[F](\nu[[E])) = P \setminus [[F](\nu[[E])],
\]
and the result then follows.

The following are some examples:

(1) We obtain co-characteristic formulae for \( \sqsubseteq_{\text{sim}} \) by dualising the characteristic formulae for \( \sqsubseteq_{\text{opsim}} \). Thus the endodeclaration \( E'_{\text{sim}} \), where
\[
E'_{\text{sim}} : p \mapsto \bigvee_{a \in A} \langle a \rangle \bigwedge_{q \in P. p \xrightarrow{a} q} X_q,
\]
co-characterises \( \sqsubseteq_{\text{sim}} \).

(2) We obtain co-characteristic formulae for \( \sqsubseteq_{\text{opsim}} \) by dualising the characteristic formulae for \( \sqsubseteq_{\text{sim}} \). Thus the endodeclaration \( E'_{\text{opsim}} \), where
\[
E'_{\text{opsim}} : p \mapsto \bigvee_{a \in A} \bigvee_{q \in P. p \xrightarrow{a} q} [a]X_q,
\]
co-characterises \( \sqsubseteq_{\text{opsim}} \).

(3) We obtain co-characteristic formulae for \( \sim_{\text{bisim}} \) by dualising its characteristic formulae. Thus \( E'_{\text{bisim}} \), where
\[
E'_{\text{bisim}} : p \mapsto \left( \bigvee_{a \in A} \langle a \rangle \bigwedge_{q \in P. p \xrightarrow{a} q} X_q \right) \vee \left( \bigvee_{a \in A} \bigvee_{q \in P. p \xrightarrow{a} q} [a]X_q \right),
\]
co-characterises \( \sim_{\text{bisim}} \).

(4) Let \( E \) be a finite set and \( \sqsubseteq \) be an NDBP for \( E \)-errors. We obtain co-characteristic formulae for \( \sqsubseteq \) similarity by dualising the characteristic formulae for its inverse (see
Theorem 3.16. The $\mathcal{P}$-indexed endodeclaration
\[
E'_{\sqsubseteq} : q \mapsto \bigvee \{ C \subseteq E[t] \mid C \sqsubseteq E[t] + E[q] \} \bigvee a \in A \bigvee q' \in \mathcal{P} \cdot q \xrightarrow{a} q' \bigvee \bigwedge Xq'
\]

co-characterises $\sqsubseteq_{\text{sim}}$.

Note that:

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

For each of our results for characteristic formulae, there is also a dual result for 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 $\mathcal{P}(\mathcal{P} \times \mathcal{P})$. A $\mathcal{P}$-indexed endodeclaration $E$ is said to co-express $\mathcal{F}$ when
\[
(q, p) \notin \mathcal{F} (S) \Leftrightarrow \sigma_{(\mathcal{P} \times \mathcal{P}) \cdot S^{-1}, q} \models E(p)
\]

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

**Lemma 5.7.** We write $\phi' : \mathcal{P}(\mathcal{P} \times \mathcal{P}) \rightarrow \mathcal{P}(\mathcal{P})^\mathcal{P}$ for the anti-isomorphism mapping a relation $S$ to the environment $\sigma_{(\mathcal{P} \times \mathcal{P}) \cdot S^{-1}}$ and let $\mathcal{F}$ be a monotone endofunction on $\mathcal{P}(\mathcal{P} \times \mathcal{P})$ and $E$ be a $\mathcal{P}$-indexed endodeclaration. Then $E$ co-expresses $\mathcal{F}$ if and only if the following diagram commutes:

\[
\begin{array}{ccc}
\mathcal{P}(\mathcal{P} \times \mathcal{P}) & \xrightarrow{\phi'} & \mathcal{P}(\mathcal{P})^\mathcal{P} \\
\mathcal{F} \downarrow & & \downarrow [E] \\
\mathcal{P}(\mathcal{P} \times \mathcal{P}) & \xrightarrow{\phi'} & \mathcal{P}(\mathcal{P})^\mathcal{P}
\end{array}
\]

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

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

**Lemma 5.9.** Let $\mathcal{F}$ be a monotone endofunction on $\mathcal{P}(\mathcal{P} \times \mathcal{P})$. Then the following are equivalent:
— $E$ expresses $\mathcal{F}$.
— $E$ co-expresses $\mathcal{F}$.

In particular:
— The endodeclaration $E'_\text{sim}$ co-expresses $\mathcal{F}_\text{sim}$.
— The endodeclaration $E'_\text{opsim}$ co-expresses $\mathcal{F}_\text{sim}$.
— The endodeclaration $E'_\text{bisim}$ co-expresses $\mathcal{F}_\text{bisim}$.
— The endodeclaration $E'_\subseteq$ co-expresses $\mathcal{F}_\subseteq$.

6. Closed formulae

In this section, we consider $\mathcal{L}(I,A)$ with $A$ countable and disjunctions and conjunctions over arbitrary cardinals, and interpret it over $A$-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 a closed formula, and we say a declaration $E : I \rightarrow \mathcal{L}(\emptyset)$ is a closed declaration. We write $\epsilon \in \mathcal{P}(\mathcal{P})$ for the empty environment.

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

**Definition 6.1.** Let $S \subseteq \mathcal{P} \times \mathcal{P}$ be a relation. A closed formula $F$ characterises $p$ up to $S$ when for all $p' \in \mathcal{P}$, we have $\epsilon, p' \models F$ if and only if $(p, p') \in S$.

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

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

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

$$f^{x+1} = f(f^x)$$

and for $\lambda$ a limit ordinal,

$$f^\lambda = \bigcap_{x < \lambda} f^x \quad \text{(the greatest lower bound of the set $\{f^x\}_{x < \lambda}$.)}$$

In particular,

$$f^0 = \top \quad \text{(the top element).}$$

Since $A$ is a set, this sequence reaches its greatest lower bound, which is $\nu f$. The corresponding construction for a declaration is as follows.
**Definition 6.3.** Let $E$ be an $I$-indexed endodeclaration. For each ordinal $\alpha$, we define closed declarations $E^\alpha : I \to \mathcal{P}(\emptyset)$ as follows:

$$E^{\alpha+1} : i \mapsto E(i)[E^\alpha]$$

and for $\lambda$ a limit ordinal,

$$E^\lambda : i \mapsto \bigwedge_{\alpha < \lambda} E^\alpha(i)$$

(the conjunction of all the $E^\alpha(i)$ with $\alpha < \lambda$).

In particular,

$$E^0 : i \mapsto \emptyset.$$ 

The following lemma establishes the connection between the approximants of $[[E]]$ 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 $\alpha$, we have

$$[[E^\alpha]] = [[E]]^\alpha.$$ 

**Proof.** The proof is by induction on $\alpha$, using Lemma 4.2(3) for the successor case. \hfill \square

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

$$(E_mD)^\alpha(m(i)) = E^\alpha(i).$$

**Proof.** The proof is by induction on $\alpha$. The limit case is trivial. For the successor case, we suppose (4) holds for all $i \in I$. Then for any $\hat{i} \in I$, we have

$$(E_mD)^{\alpha+1}(m(\hat{i})) = E_mD[(E_mD)^\alpha(m(\hat{i}))] = E_mD(m(\hat{i}))[E_mD^\alpha]$$

(by definition)

$$= E(\hat{i})[i \mapsto X_{m(\hat{i})}][(E_mD)^\alpha]$$

(by definition of $E_mD$)

$$= E(\hat{i})[i \mapsto X_{m(\hat{i})}][(E_mD)^\alpha]$$

(by Lemma 4.2(2))

$$= E(\hat{i})[i \mapsto (E_mD)^\alpha(m(i))]$$

(by definition)

$$= E(\hat{i})[i \mapsto E^\alpha(i)]$$

(by the induction hypothesis)

$$= E\hat{i}[i \mapsto E^\alpha(i)]$$

(by definition)

$$= E^{\alpha+1}(\hat{i})$$

(by definition)

as required. \hfill \square

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

$$\phi(\bigcap X) = \bigcap \phi(X) = \bigcap \{\phi(x) \mid x \in X\}.$$
Lemma 6.7 (analog of Theorem 2.3). Let $A$ and $B$ be complete lattices, $f$ and $g$ be monotone endofunctions on $A$ and $B$, respectively, and $\phi : A \to B$ be an isomorphism such that $\phi \circ f = g \circ \phi$. Then for each ordinal $\alpha$, we have $\phi(f^\alpha) = g^\alpha$.

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

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

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

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

$\varepsilon, p' \models E^\alpha(p) \iff p' \in ([E^\alpha]_{\varepsilon}(p))$

$\iff p' \in ([E]^\varepsilon(p))$  \hspace{1cm} (from Lemma 6.4)

$\iff p' \in (\Phi(F^\varepsilon))(p)$  \hspace{1cm} (from Lemma 6.7)

$\iff (p, p') \in F^\varepsilon$

$\iff (p, p') \in vF$  \hspace{1cm} (by the property in the statement of the lemma)

as required. \hspace{1cm} \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 $\kappa$ for which if we are given any $I$ with cardinality $< \kappa$ and 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 $\kappa$ be an infinite regular cardinal. Then:

1. A process $p \in P$ is image $\kappa$-bounded when for each $a \in A$ the set $\{q \in P \mid p \xrightarrow{a} q\}$ has cardinality $< \kappa$.

2. $P$ is an image $\kappa$-bounded $A$-labelled transition system when each $p \in P$ is image $\kappa$-bounded.

3. A formula $F \in \mathcal{L}(I)$ is $\kappa$-bounded when all the conjunctions and disjunctions in it are of arity $< \kappa$.

In particular, $\kappa = \omega$ gives image finiteness (Hennessy and Milner 1985), and $\kappa = \omega_1$ (the smallest uncountable ordinal) gives image countability.

Theorem 6.11. Let $\kappa$ be an infinite regular cardinal. Suppose $p \in P$ is image $\kappa$-bounded. Then:

1. If $p' \in P$ is image $\kappa$-bounded, then $p \sqsubseteq_{\text{sim}} p'$ if and only if $(p, p') \in F^\kappa_{\text{sim}}$. 


(2) If $p' \in P$, then $p \sqsubseteq_{\text{opsim}} p'$ if and only if $(p, p') \in \mathcal{F}_{\text{opsim}}^\kappa$.
(3) If $p' \in P$, then $p \sim_{\text{bisim}} p'$ if and only if $(p, p') \in \mathcal{F}_{\text{bisim}}^\kappa$.
(4) If $p' \in P$ is image $\kappa$-bounded, then $p \sqsubseteq_{2\text{sim}} p'$ if and only if $(p, p') \in (\sqsubseteq_{\text{opsim}} \cap \mathcal{F}_{\text{sim}})^\kappa$.

**Proof.** For $\kappa = \omega$, statements (1), (2) and (4) can be proved using the method found in Hennessy and Milner (1985). Statement (3) was proved in van Glabbeek (1987). The general case is similar – a detailed proof is given in Appendix B.

In the rest of the paper, we will write $\kappa^+$ for the successor cardinal of $\kappa$.

**Corollary 6.12.** Let $\kappa$ be an infinite regular cardinal. Then:

1. If $P$ is an image $\kappa$-bounded system and $p \in P$, then the closed formula $(E_{\text{sim}})^\kappa(p)$ is $\kappa^+$-bounded and characterises $p$ up to $\sqsubseteq_{\text{sim}}$.
2. If $p \in P$ is image $\kappa$-bounded, then the closed formula $(E_{\text{opsim}})^\kappa(p)$ is $\kappa^+$-bounded and characterises $p$ up to $\sqsubseteq_{\text{opsim}}$.
3. If $p \in P$ is image $\kappa$-bounded, then the closed formula $(E_{\text{bisim}})^\kappa(p)$ is $\kappa^+$-bounded and characterises $p$ up to $\sim_{\text{bisim}}$.

In each case, the characteristic formula is $\kappa^+$-bounded.

**Proof.** We will only prove statement (3) as this is the most difficult. For each ordinal $\alpha$, Lemma 6.5 gives us

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

Since $A$ is assumed to be countable, the formulae in $E_{\text{bisim}} \uparrow \text{reach}(p)$ are $\kappa^+$-bounded (indeed $\kappa$-bounded if $\kappa > \aleph_0$). Therefore, for $\alpha < \kappa$, we know the formula $(E_{\text{bisim}} \uparrow \text{reach}(p))^\alpha(p)$ is $\kappa^+$-bounded by induction on $\alpha$.

Lemma 6.8 and Theorem 6.11(1) together with parts (2) and (3) tell us that $(E_{\text{bisim}})^\kappa$ characterises $p$ up to $\sim_{\text{bisim}}$.

Corollary 6.12(3) gives Baltag’s construction, which is described in Theorem 11.12 of Barwise and Moss (1996).

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 $R \subseteq P \times P$ and $E : P \rightarrow \mathcal{L}(\emptyset)$ be a closed declaration such that for each $p \in P$, the closed formula $E(p)$ characterises $p$ up to $R$. Let $\hat{E} : P \rightarrow \mathcal{L}(P)$ 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 $\mathcal{P}(P \times P)$.
2. Let $F$ be a monotone endofunction on $\mathcal{P}(P \times P)$ and $D$ be a $P$-indexed endodeclaration expressing $F$. Then the declaration

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

expresses $R \cap F$. 


Proof.

(1) This part is obvious.
(2) This part follows from part (1) and Lemma 4.8.

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

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

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

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

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 just recover characteristic formulae already discovered but others are 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 Groote and Vaandrager (1992) 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 provides.

Overall, this study provides a collection of ‘tools’ that anyone can use to define characteristic formulae for processes with respect to their favourite notion of behavioural relation defined as a fixed point of a monotone endofunction. Formally, the main tools we provide 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 illustrates the usability of the framework we have developed.
There are several possible avenues for further work based on the approach and results we have presented 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 (Milner 1989), branching bisimulation equivalence (van Glabbeek and Weijland 1996), resource bisimulation equivalence (Corradini et al. 1999), g-bisimulation equivalence (de Rijke 2000) and probabilistic bisimulation equivalence (Larsen and Skou 1992). However, a general view of characteristic formulae for behavioural equivalences such as resource bisimulation and probabilistic bisimulation equivalence may require a further generalisation 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 described in Aceto et al. (2000) and Laroussinie et al. (1995), as well as of the characteristic-formula constructions for bisimulation-based relations in terms of temporal logics like CTL, such as those described in Browne et al. (1988). We leave these further developments for future work.

Appendix A. Proof of Lemma 4.4

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

$$(\hat{a}, \hat{b}) = H(\hat{a}, \hat{b})$$

$$= (F(\hat{a}), G(\hat{a}, \hat{b})).$$

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

$$\hat{b} = G(\hat{a}, \hat{b})$$

$$\sqsubseteq G(\nu f, \hat{b})$$

$$= G_{\nu f}(\hat{b}),$$

so $\hat{b}$ is a post-fixed point of $G_{\nu f}$. If $b$ is a post-fixed point of $G_{\nu f}$, then

$$(\nu f, b) \sqsubseteq (F(\nu f), G_{\nu f}(b))$$

$$= (F(\nu f), G(\nu f, b))$$

$$= H(\nu f, b).$$

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

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

$$= (F(\nu f), G_{\nu f}(\nu g_{\nu f}))$$

$$= (\nu f, \nu(G_{\nu f})),$$

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

$$(a, b) \sqsubseteq H(a, b)$$

$$= (F(a), G(a, b)), $$
so \( a \) is a post-fixed point of \( f \), giving \( a \sqsubseteq vf \), and
\[
\begin{align*}
b &\sqsubseteq G(a, b) \\
&\sqsubseteq G(vf, b) \\
&= G_{vf}(b),
\end{align*}
\]
so \( b \) is a post-fixed point of \( G_{vf} \), giving \( b \sqsubseteq vG_{vf} \). Thus \((vf, vG_{vf})\) is a greatest post-fixed point of \( H \), as required. \( \square \)

Appendix B. Proof of Theorem 6.11

Given \( A \)-labelled transition systems \( Q = (Q, \rightarrow) \) and \( Q' = (Q', \rightarrow) \), we define three monotone endofunctions
\[
\mathcal{F}_{\sim}^{Q', Q}, \quad \mathcal{F}_{\text{opsim}}^{Q, Q'}, \quad \mathcal{F}_{\text{bisim}}^{Q, Q'}
\]
on \( \mathcal{P}(Q \times Q') \), just as in Sections 3.1.1 and 3.2.1. Using these, we define relations
\[
\sqsubseteq_{\sim}^{Q, Q'}, \quad \sqsubseteq_{\text{opsim}}^{Q, Q'}, \quad \sqsubseteq_{\text{bisim}}^{Q, Q'}, \quad \sqsubseteq_{\text{2sim}}^{Q, Q'}
\]
each of which is a subset of \( Q \times Q' \), just as in Sections 3.1.1, 3.2.1 and 4.4. We shall show the following results.

**Theorem B.1.** Let \( Q = (Q, \rightarrow) \) and \( Q' = (Q', \rightarrow) \) be \( A \)-labelled transition systems. Let \( \kappa \) be an infinite regular cardinal.

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

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

**Lemma B.2.** Let \( P = (P, \rightarrow) \) be an \( A \)-labelled transition system, and let \( Q, Q' \) be subsystems (in the sense of Definition 4.7(1)) of \( P \), giving systems \( Q = (Q, \rightarrow) \) and \( Q' = (Q', \rightarrow) \). Then we have
\[
\begin{align*}
\sqsubseteq_{\sim}^{Q, Q'} &= \sqsubseteq_{\sim} \cap (Q \times Q') \quad \text{(5)} \\
\sqsubseteq_{\text{opsim}}^{Q, Q'} &= \sqsubseteq_{\text{opsim}} \cap (Q \times Q') \quad \text{(6)} \\
\sqsubseteq_{\text{bisim}}^{Q, Q'} &= \sqsubseteq_{\text{bisim}} \cap (Q \times Q') \quad \text{(7)} \\
\sqsubseteq_{\text{2sim}}^{Q, Q'} &= \sqsubseteq_{\text{2sim}} \cap (Q \times Q') \quad \text{(8)}
\end{align*}
\]
and for any ordinal \( \alpha \) we have
\[
\begin{align*}
(\mathcal{F}_{\sim}^{Q, Q'})^\alpha &= \mathcal{F}_{\sim}^{\alpha} \cap (Q \times Q') \quad \text{(9)} \\
(\mathcal{F}_{\text{opsim}}^{Q, Q'})^\alpha &= \mathcal{F}_{\text{opsim}}^{\alpha} \cap (Q \times Q') \quad \text{(10)} \\
(\mathcal{F}_{\text{bisim}}^{Q, Q'})^\alpha &= \mathcal{F}_{\text{bisim}}^{\alpha} \cap (Q \times Q') \quad \text{(11)} \\
(\sqsubseteq_{\text{opsim}} \cap \mathcal{F}_{\sim}^{Q, Q'})^\alpha &= (\sqsubseteq_{\text{opsim}} \cap \mathcal{F}_{\sim}^{\alpha}) \cap (Q \times Q'). \quad \text{(12)}
\end{align*}
\]
Proof. (9)–(11) are proved by induction on \( \alpha \), and we can then deduce (5)–(7). We then prove (12) by induction on \( \alpha \) using (6), and, finally, we can then deduce (8).

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

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

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

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

**Lemma B.4.** Let \( \kappa \) be an infinite regular cardinal, and \( A \) and \( B \) be complete lattices. For any function \( f : A \rightarrow B \), the following statements are equivalent:

1. For every downwards \( \kappa \)-directed \( U \subseteq A \), we have
   \[
   f \left( \bigcap_{a \in U} a \right) = \bigcap_{a \in U} f(a).
   \]
2. \( f \) is monotone and for every downwards \( \kappa \)-directed \( U \subseteq A \), we have
   \[
   f \left( \bigcap_{a \in U} a \right) \subseteq \bigcap_{a \in U} f(a).
   \]

**Proof.**

(1) \( \Rightarrow \) (2)

For monotonicity, we suppose \( a \sqsubseteq b \in A \). Then \( \{a, b\} \) is downwards \( \kappa \)-directed (as \( a \) is a least element). So

\[
f(a) = f(a \sqcap b) = f(a) \sqcap f(b) \subseteq f(b).
\]

(2) \( \Rightarrow \) (1)

Let \( U \subseteq A \). For each \( a \in U \), we have \( f(\bigcap_{a \in U} a) \subseteq f(a) \) by monotonicity. So \( f(\bigcap_{a \in U} a) \subseteq \bigcap_{a \in U} f(a) \).

We say that a function \( f : A \rightarrow B \) **preserves \( \kappa \)-directed meets** when it satisfies the conditions of Lemma B.4.

The preservation of \( \kappa \)-directed meets has the following application.

**Lemma B.5.** Let \( \kappa \) be an infinite regular cardinal and \( f \) be an endofunction on a complete lattice \( A \) that preserves \( \kappa \)-directed meets. Then \( \nu f = f^\kappa \).
Proof. The set \( \{ f^x \}_{x < \kappa} \) is downwards \( \kappa \)-directed, since \( \kappa \) is regular. So

\[
f(f^\kappa) = f \left( \bigcap_{x < \kappa} f^x \right) \\
= \bigcap_{x < \kappa} f(f^x) \\
= \bigcap_{x < \kappa} f^{x+1} \\
= f^\kappa.
\]

Lemma B.6. Let \( A \) and \( B \) be complete lattices and \( \kappa \) be an infinite regular cardinal.

(1) For each \( i \in I \), let \( f_i : A \to B \) be a \( \kappa \)-directed meet preserving function. Then the function \( A \to B \) mapping \( a \) to \( \bigcap_{i \in I} f_i(a) \) preserves \( \kappa \)-directed meets.

(2) Let \( b \in B \). Then the constant function \( A \to B \) mapping any \( a \) to \( b \) preserves \( \kappa \)-directed meets.

(3) Let \( b \in B \) and \( f : A \to B \) be a function preserving \( \kappa \)-directed meets. Then the function \( b \sqcap f : A \to B \) preserves \( \kappa \)-directed meets.

Proof.

(1) This part is trivial.

(2) This part follows from the fact that a downwards \( \kappa \)-directed set must be non-empty.

(3) This part follows from parts (1) and (2).

Lemma B.7. Let \( C \) and \( D \) be sets and \( \kappa \) be an infinite regular cardinal. For any function \( f : \mathcal{P}(C) \to \mathcal{P}(D) \), the following statements are equivalent:

(1) \( f \) preserves \( \kappa \)-directed meets.

(2) For any \( M \subseteq C \) and \( d \in D \), if \( d \notin f(M) \), 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 \( \kappa \)-directed (because \( \kappa \) is regular), and its intersection is \( M \). So we have

\[
f(M) = f \left( \bigcap_{N \subseteq A} (C \setminus N) \right) \\
= \bigcap_{N \subseteq A} f(C \setminus N)
\]
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 that \( f \) is monotone, suppose \( M' \subseteq M \). 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 \( \kappa \)-directed. If \( d \notin f(\bigcap_{u \in U} u) \), 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 \( a \in N \), we 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 \( a \notin u' \), that is, \( u' \) is disjoint from \( N \), so \( d \notin f(u') \) and hence \( d \notin \bigcap_{u \in U} f(u) \). We can then conclude that \( \bigcap_{u \in U} f(u) \subseteq f(\bigcap_{u \in U} u) \). \( \square \)

**Remark.** Lemma B.7 is an easy equivalent (by duality) of the following well-known result: a function \( g : \mathcal{P}(C) \rightarrow \mathcal{P}(D) \) preserves \( \kappa \)-directed joins if and only if 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 B.8.** Let \( Q = (Q, \rightarrow) \) and \( Q' = (Q', \rightarrow) \) be \( A \)-labelled transition systems and \( \kappa \) be an infinite regular cardinal. Then:

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

**Proof.**

1. We use Lemma B.7. Suppose \( R \subseteq Q \times Q' \) and \( (p, p') \notin \mathcal{F}_{\text{sim}}(R) \). Then there is \( a \in A \) and \( q \in Q \) such that \( p \xrightarrow{a} q \). But there is no \( q' \in Q' \) 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 \( \kappa \)-bounded. For any \( R' \subseteq Q \times Q' \), if \( R' \) is disjoint from \( N \), then there is no \( q' \in Q' \) such that \( p' \xrightarrow{a} q' \) and \( (q, q') \in R' \), so \( (p, p') \notin \mathcal{F}_{\text{sim}}(R') \). This establishes the required condition.
2. This is the dual result.
3. This part follows by a similar argument.
4. This part follows from part (1) and Lemma B.6(3). \( \square \)

Using Lemma B.5, we can deduce parts (1), (2) and (4) of Theorem B.1 from Lemma B.8. But part (3) requires a more subtle argument.

**Definition B.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.
Note that an intersection of difunctional relations is also difunctional, so \( \text{Difun}(A, B) \) is a complete lattice.

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

**Proof.** Suppose \( (p_1, p_1'), (p_2, p_2') \in \mathcal{F}_{\text{bisim}}^R \). If \( p_2 \xrightarrow{a} q_2 \), then:

- \( (p_2, p_1') \in \mathcal{F}_{\text{bisim}}^R \) gives \( q_1' \in Q \) such that \( p_1' \xrightarrow{a} q_1' \) and \( (q_2, q_1') \in R \).
- \( (p_1, p_1') \in \mathcal{F}_{\text{bisim}}^R \) gives \( q_1 \in Q \) such that \( p_1 \xrightarrow{a} q_1 \) and \( (q_1, q_1') \in R \).
- \( (p_1, p_2') \in \mathcal{F}_{\text{bisim}}^R \) gives \( q_2' \in Q \) 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 \( (p_2, p_2') \in \mathcal{F}_{\text{bisim}}^R \), and the result then follows straightforwardly.

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

**Lemma B.11.** Let \( \kappa \) be an infinite regular cardinal and \( C \) be a set of size \( < \kappa \). Then any \( U \subseteq P(C) \) that is downwards \( \kappa \)-directed has a least element.

**Proof.** For each \( c \in C \setminus \bigcap_{u \in U} u \), we pick \( u_c \subseteq U \) 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 \( c \subsetneq u' \setminus \bigcap_{u \in U} u \), then \( c \notin u_c \), which contradicts \( u \subseteq u_c \). Hence \( u' = \bigcap_{u \in U} u \).

**Lemma B.12.** If \( \kappa \) is an infinite regular cardinal and \( Q = (Q, \rightarrow) \) and \( Q' = (Q', \rightarrow) \) are \( A \)-labelled transition systems, with \( Q \) image \( \kappa \)-bounded, then \( \mathcal{G}^Q \) preserves \( \kappa \)-directed meets.

**Proof.** The monotonicity of \( \mathcal{G} \) is inherited from that of \( \mathcal{F}_{\text{bisim}}^Q \).

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

\[
\bigcap_{R \in U} \mathcal{G}(R) \subseteq \bigcap_{R \in U} \mathcal{F}_{\text{opsim}}(R) = \mathcal{F}_{\text{opsim}}(\bigcap_{R \in U} R) \quad \text{(by Lemma B.8(2))}
\]

We only need to prove

\[
\bigcap_{R \in U} \mathcal{G}(R) \subseteq \mathcal{F}_{\text{sim}} \left( \bigcap_{R \in U} R \right)
\]

since we can then conclude

\[
\bigcap_{R \in U} \mathcal{G}(R) \subseteq \mathcal{F}_{\text{sim}} \left( \bigcap_{R \in U} R \right) \cap \mathcal{F}_{\text{opsim}} \left( \bigcap_{R \in U} R \right) = \mathcal{G} \left( \bigcap_{R \in U} R \right).
\]
Suppose \((p, p') \in \bigcap_{R \in U} \mathcal{G}(R)\) and \(p \xrightarrow{a} q\). Let \(B = \{r \in P \mid p \xrightarrow{a} r\}\). We define a function \(H : U \rightarrow \mathcal{P}(B)\) mapping \(R\) to

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

\(H\) is monotone, so the set \(L = \{H(R) \mid R \in U\}\) is downwards \(\kappa\)-directed, and thus, by Lemma B.11, it 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 \(q' \in Q'\) such that \(p' \xrightarrow{a} q'\) and \((q, q') \in S\).

For any \(R \in U\), we pick a lower bound \(R' \in U\) for \(R\) and \(S\). Now, 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 \(r' \in Q\) such that \((q, r') \in R'\) and \((r, r') \in R'\). Difunctionality of \(R'\) then gives \((q, q') \in R'\), so \((q, q') \in R\). Thus \((q, q') \in \bigcap_{R \in U} R\) and we can conclude that \((p, p') \in F_{\text{sim}}(\bigcap_{R \in U} R)\).

\(\square\)

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

**Acknowledgements**

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

**References**

Aceto, L. and Hennessy, M. (1992) Termination, deadlock, and divergence. *Journal of the ACM* 39 (1) 147–187.

Aceto, L., Ingolfsdottir, A., Larsen, K.G. and Srba, J. (2007) *Reactive Systems: Modelling, Specification and Verification*, Cambridge University Press.

Aceto, L., Ingolfsdottir, A., Pedersen, M. L. and Poulsen, J. (2000) Characteristic formulae for timed automata. *RAIRO, Theoretical Informatics and Applications* 34 (6) 565–584.

Barwise, J. and Moss, L. (1996) *Vicious Circles: On the mathematics of non-wellfounded phenomena*, CSLI Lecture Notes 60, CSLI Publications.

Barwise, J. and Moss, L. S. (1998) Modal correspondence for models. *J. Philos. Logic* 27 (3) 275–294.

Bloom, B., Istrail, S. and Meyer, A. (1995) Bisimulation can’t be traced. *Journal of the ACM* 42 (1) 232–268.

Boudol, G. and Larsen, K.G. (1992) Graphical versus logical specifications. *Theoretical Computer Science* 106 (1) 3–20.

Brown, M. C., Clarke, E. M. and Grumberg, O. (1988) Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science* 59 (1-2) 115–131.

Clarke, E. C., Emerson, E. A. and Sistla, A. P. (1986) Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems* 8 (2) 244–263.
Characteristic formulae for fixed-point semantics: a general framework

Clarke, E., Grumberg, O. and Peled, D. (1999) Model Checking, MIT Press.

Clarke, E. M. and Emerson, E. A. (1981) Design and synthesis of synchronization skeletons using branching time temporal logic. In: Kozen, D. (ed.) Proceedings of the Workshop on Logics of Programs. Springer-Verlag Lecture Notes in Computer Science 131 52–71.

Cleaveland, R. and Steffen, B. (1991) Computing behavioural relations, logically. In: Albert, J. L., Monien, B. and Rodriguez, M. (eds.) Proceedings 18th ICALP, Madrid. Springer-Verlag Lecture Notes in Computer Science 510 127–138.

Corradini, F., De Nicola, R. and Labella, A. (1999) Graded modalities and resource bisimulation. In: Rangan, C. P., Raman, V. and Ramanujam, R. (eds.) Proceedings Foundations of Software Technology and Theoretical Computer Science, 19th Conference, Chennai, India. Springer-Verlag Lecture Notes in Computer Science 1738 381–393.

De Nicola, R., Montanari, U. and Vaandrager, F. (1990) Back and forth bisimulations. In: Proceedings CONCUR’90 (Amsterdam, 1990). Springer-Verlag Lecture Notes in Computer Science 458 152–165.

De Nicola, R. and Vaandrager, F. W. (1995) Three logics for branching bisimulation. Journal of the ACM 32 (2) 458–487.

de Rijke, M. (2000) A note on graded modal logic. Studia Logica 64 (2) 271–283.

Dechesne, F., Mousavi, M. R. and Orzan, S. (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). Springer-Verlag Lecture Notes in Artificial Intelligence 4790 226–241.

Ebbinghaus, H.-D., Flum, J. and Thomas, W. (1994) Mathematical Logic, second edition, Undergraduate Texts in Mathematics, Springer-Verlag. (Translated from the German by Margit Meßmer.)

Emerson, E. A. (1990) Temporal and modal logic. In: Handbook of Theoretical Computer Science, Vol. B, Elsevier 995–1072.

Fábregas, I., de Frutos-Escrig, D. and Palomino, M. (2009) Non-strongly stable orders also define interesting simulation relations. In: Kurz, A., Lenisa, M. and Tarlecki, A. (eds.) Proceedings: Algebra and Coalgebra in Computer Science, Third International Conference, CALCO 2009, Udine, Italy. Springer-Verlag Lecture Notes in Computer Science 5728 221–235.

Fecher, H. and Steffen, M. (2005) Characteristic μ-calculus formula for an underspecified transition system. In: EXPRESS’04. Electronic Notes in Theoretical Computer Science 128 103–116.

van Glabbeek, R. (1987) Bounded nondeterminism and the approximation induction principle in process algebra. In: Brandenburg, F.-J., Vidal-Naquet, G. and Wirsing, M. (eds.) Proceedings: STACS 87, 4th Annual Symposium on Theoretical Aspects of Computer Science, Passau, Germany. Springer-Verlag Lecture Notes in Computer Science 247 336–347.

van Glabbeek, R. (2001) The linear time–branching time spectrum. I. The semantics of concrete, sequential processes. In: Bergstra, J., Ponse, A. and Smolka, S. A. (eds.) Handbook of Process Algebra, Elsevier 3–99.

van Glabbeek, R. and Weijland, W. P. (1996) Branching time and abstraction in bisimulation semantics. Journal of the ACM 43 (3) 555–600.

Graf, S. and Sifakis, J. (1986) A modal characterization of observational congruence on finite terms of CCS. Information and Control 68 (1-3) 125–145.

Groote, J.-F. and Vaandrager, F. W. (1992) Structured operational semantics and bisimulation as a congruence. Information and Computation 100 (2) 202–260.

Hennessy, M. and Milner, R. (1985) Algebraic laws for nondeterminism and concurrency. Journal of the ACM 32 (1) 137–161.
Ingolfsdottir, A., Godskesen, J.C. and Zeeberg, M. (1987) Fra Hennessy–Milner logik til CCS-processer. Master’s thesis, Department of Computer Science, Aalborg University. (In Danish.)

Keller, R.M. (1976) Formal verification of parallel programs. *Communications of the ACM* **19** (7) 371–384.

Korver, H. (1991) Computing distinguishing formulas for branching bisimulation. In: Larsen, K.G. and Skou, A. (eds.) Proceedings of the Third Workshop on Computer Aided Verification, Aalborg, Denmark. *Springer-Verlag Lecture Notes in Computer Science* **575** 13–23.

Kozen, D. (1983) Results on the propositional mu-calculus. *Theoretical Computer Science* **27** 333–354.

Kucera, A. and Schnoebelen, Ph. (2006) A general approach to comparing finite-state systems with their finite-state specifications. *Theoretical Computer Science* **358** (2-3) 315–333.

Kozen, D. (1983) Results on the propositional mu-calculus. *Theoretical Computer Science* **27** 333–354.

Levy, P.B. (2009) Boolean precongruences.

Margaria, T. and Steffen, B. (1993) Distinguishing formulas for free. In: *Proceedings EDAC–EUROASIC’93: IEEE European Design Automation Conference, Paris, France*, IEEE Computer Society Press.

Milner, R. (1981) A modal characterisation of observable machine behaviour. In: Astesiano, E. and Böhm, C. (eds.) CAAP ’81: Trees in Algebra and Programming, 6th Colloquium. *Springer-Verlag Lecture Notes in Computer Science* **630** 456–471.

Moss, L.S. (2007) Finite models constructed from canonical formulas. *J. Philos. Logic* **36** (6) 605–640.

Müller-Olm, M. (1998) Derivation of characteristic formulae. In: MFCS’98 Workshop on Concurrency, Brno, Czech Republic. *Electronic Notes in Theoretical Computer Science* **18**.

Park, D. (1981) Concurrency and automata on infinite sequences. In: Deussen, P. (ed.) 5th GI Conference, Karlsruhe, Germany. *Springer-Verlag Lecture Notes in Computer Science* **104** 167–183.

Pitcher, C. (2001) *Functional Programming and Erratic Non-Determinism*, Ph.D. thesis, Oxford University.

Plotkin, G.D. (2004) A structural approach to operational semantics. *Journal of Logic and Algebraic Programming* (60-61) 17–139.

Pnueli, A. (1997) The temporal logic of programs. In: *Proceedings 18th Annual Symposium on Foundations of Computer Science*, IEEE 46–57.
Queille, J.P. and Sifakis, J. (1981) Specification and verification of concurrent systems in Cesar. In: Proceedings of the 5th International Symposium on Programming. Springer-Verlag Lecture Notes in Computer Science 137 337–351.

Steffen, B. and Ingolfsdottir, A. (1994) Characteristic formulae for processes with divergence. Information and Computation 110 (1) 149–163.

Stirling, C. (1987) Modal logics for communicating systems. In: Twelfth international colloquium on automata, languages and programming. Theoretical Computer Science 49 (2-3) 311–347.

Tarski, A. (1955) A lattice-theoretical fixpoint theorem and its applications. Pacific Journal of Mathematics 5 285–309.

Thomas, W. (1993) On the Ehrenfeucht–Fraïssé game in theoretical computer science. In: Gaudel, M.-C. and Jouannaud, J.-P. (eds.) Proceedings TAPSOFT. Springer-Verlag Lecture Notes in Computer Science 668 559–568.

Thomsen, B. (1987) An extended bisimulation induced by a preorder on actions. Master’s thesis, Aalborg University Centre.

Ulidowski, I. (1992) Equivalences on observable processes. In: Scedrov, A. (ed.) Proceedings of the 7th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press 148–161.