

A CHARACTERISATION OF OPEN BISIMILARITY USING AN INTUITIONISTIC MODAL LOGIC

KI YUNG AHN^a, ROSS HORNE^b, AND ALWEN TIU^c

^a Department of Computer Engineering, Hannam University, Daejeon, Korea
e-mail address: kya@hnu.kr

^b Computer Science, University of Luxembourg, Esch-sur-Alzette, Luxembourg
e-mail address: ross.horne@uni.lu

^c Research School of Computer Science, Australian National University, Canberra, Australia
e-mail address: alwen.tiu@anu.edu.au

ABSTRACT. Open bisimilarity is defined for open process terms in which free variables may appear. The insight is, in order to characterise open bisimilarity, we move to the setting of intuitionistic modal logics. The intuitionistic modal logic introduced, called \mathcal{OM} , is such that modalities are closed under substitutions, which induces a property known as *intuitionistic hereditary*. Intuitionistic hereditary reflects in logic the lazy instantiation of free variables performed when checking open bisimilarity. The soundness proof for open bisimilarity with respect to our intuitionistic modal logic is mechanised in Abella. The constructive content of the completeness proof provides an algorithm for generating distinguishing formulae, which we have implemented. We draw attention to the fact that there is a spectrum of bisimilarity congruences that can be characterised by intuitionistic modal logics.

1. INTRODUCTION

This work provides insight into the logical nature of open bisimilarity [San96], but firstly we recall why open bisimilarity itself is important. An asset of open bisimilarity is that it defines a congruence relation for open process terms, i.e., process terms containing free variables. Recall that the original notions of bisimilarity proposed for the π -calculus (early and late bisimilarity [MPW92, MPW93]) do not directly define congruence relations for open process terms. Having a bisimilarity that is a congruence for open process terms improves compositional reasoning, since, having established an algebraic property, we can apply the property with confidence, anywhere inside a process, even under constructs such as input prefixes that bind variables. By providing a notion of bisimilarity that is a congruence for open process terms, open bisimilarity provides a method for the π -calculus that stays true to this desirable property of a processes algebra.

Besides improved algebraic properties, open bisimilarity can be used to improve the efficiency of equivalence checking. For example, open bisimilarity is the notion of bisimilarity

Key words and phrases: bisimulation, modal logic, intuitionistic logic.

implemented in the Mobility Workbench [VM94] – the first toolkit for the π -calculus; and in the tool SPEC [TNH16] – an equivalence checker for the spi-calculus, useful for verifying cryptographic protocols. A reason open bisimilarity is efficient to implement is that it allows a lazy approach to instantiating variables. When we perform an input action, we are not required to explore all possible inputs. Instead, we can represent the input value as a variable symbolically representing all possible inputs. This symbolic approach to inputs can avoid unnecessarily exploring hyper-exponentially many inputs; instead, exploring only the state space necessary. This lazy “call-by-need” approach to input transitions is particularly useful when checking bisimilarity for applied extensions of the π -calculus, where infinitely many messages may be received for a single input action [BN06, TD10, HM21]. Thus open bisimilarity has impact beyond the setting of the π -calculus.

The trick for ensuring that open bisimilarity is a congruence, and also for permitting a lazy approach to inputs is as follows: an open bisimulation is closed under all permitted substitutions at every step in the bisimulation game. When we move to the setting of logic, closure under substitutions corresponds to a concept called *intuitionistic hereditary*, which can be used to induce an intuitionistic logic [Kri65]. This observation leads us to the intuitionistic modal logic in this work.

To understand why closing under substitutions results in an intuitionistic modal logic, firstly consider the setting of a **classical modal logic**. In a classical setting, the law of excluded middle holds, hence we expect that $\langle \tau \rangle \mathbf{tt} \vee \neg \langle \tau \rangle \mathbf{tt}$ is a tautology. That is, any process can either perform a τ -transition or it cannot perform a τ -transition.

In contrast, now consider the setting of an **intuitionistic modal logic**. In the intuitionistic setting we close under all substitutions, so $P \models \neg \langle \tau \rangle \mathbf{tt}$ now reads, *under any substitution* σ , process $P\sigma$ cannot perform a τ -transition. Under this interpretation we have the following.

$$\bar{a}b \mid c(x) \not\models \neg \langle \tau \rangle \mathbf{tt}$$

To see why the above is not satisfiable, observe that, by applying substitution $\{c/a\}$ to the above process, we reach process $\bar{c}b \mid c(x)$, which is a π -calculus process for which a communication is enabled on the channel represented by variable c . Since we have demonstrated there is a substitution under which a τ -transition can be performed, process $\bar{a}b \mid c(x)$ cannot satisfy formula $\neg \langle \tau \rangle \mathbf{tt}$ in the intuitionistic setting.

As in the classical case, in the intuitionistic case we have the following, since there is a substitution under which no communication can be performed (the identity substitution).

$$\bar{a}b \mid c(x) \not\models \langle \tau \rangle \mathbf{tt}$$

Putting the above together, we have the following in our intuitionistic modal logic, since we have just shown that neither branch of the disjunction is satisfiable.

$$\bar{a}b \mid c(x) \not\models \langle \tau \rangle \mathbf{tt} \vee \neg \langle \tau \rangle \mathbf{tt}$$

Notice the above we claimed was a tautology in the classical case, since it is an instance of the law of excluded middle. Hence the above example demonstrates that, by closing operators of the modal logic under substitutions, the law of excluded middle does not hold. The absence of the law of excluded middle is a key criterion for any intuitionistic logic.

Intuitively, the absence of the law of excluded middle for the example above can be interpreted as follows. For $\bar{a}b \mid c(x)$, we have not yet decided whether the process can perform a τ -transition or not perform a τ -transition. It is possible that a and c could be the same channel but, since they are variables, we have not yet decided whether this is the case.

So, inducing the key feature of an open bisimulation, closure under substitutions, in a modal logic gives rise to an intuitionistic modal logic. Furthermore, we establish in this work that such an intuitionistic modal logic, called \mathcal{OM} , characterises open bisimilarity. In the tradition pioneered by Hennessy and Milner [HM85], a modal logic characterises a bisimilarity whenever, given two processes, they are bisimilar if and only if there is no *distinguishing formula* separating them. A distinguishing formula is a formula that holds for one process but does not hold for the other process. Such distinguishing formulae are useful for explaining why two processes are not bisimilar, since when processes are not bisimilar we can always exhibit a distinguishing formula.

As an example of a distinguishing formula, consider the following two processes.

$$R \triangleq \tau.(\bar{a}b.a(x) + a(x).\bar{a}b + \tau) + \tau.(\bar{a}b.c(x) + c(x).\bar{a}b) \qquad S \triangleq R + \tau.(\bar{a}b \mid c(x))$$

The above processes are not open bisimilar. Process R satisfies $[\tau](\langle \tau \rangle \mathbf{tt} \vee \neg \langle \tau \rangle \mathbf{tt})$, where the box modality indicates that $\langle \tau \rangle \mathbf{tt} \vee \neg \langle \tau \rangle \mathbf{tt}$ holds for *all* processes reachable by applying a substitution and then a τ -transition. However, process S does not satisfy $[\tau](\langle \tau \rangle \mathbf{tt} \vee \neg \langle \tau \rangle \mathbf{tt})$, since there is a τ -transition to process $\bar{a}b \mid c(x)$ that we just agreed does not satisfy $\langle \tau \rangle \mathbf{tt} \vee \neg \langle \tau \rangle \mathbf{tt}$. In this example, the absence of the law of excluded middle is necessary in order for a formula distinguishing these processes to exist in \mathcal{OM} .

Modal logics characterising late bisimilarity and early bisimilarity were developed early in the literature on the π -calculus, by Milner, Parrow and Walker [MPW93], as part of the motivation for the π -calculus itself. However, proving that a modal logic can characterise open bisimilarity was an open problem until a solution was provided in the conference version of this paper [AHT17a]. This extended version includes more details on proofs, new examples, details on the mechanisation of soundness, and further insight into the spectrum of bisimilarity congruences that can be characterised by variants of our intuitionistic modal logic. We also show that soundness and completeness results for \mathcal{OM} extend from finite processes to infinite but finitely branching processes, without changing the logic, since finite distinguishing strategies are sufficient to distinguish such processes.

A key novelty of this work is the constructive proof of completeness of this logical characterisation. Due to the intuitionistic nature of the modal logic, the completeness proof cannot appeal to certain classical principles, such as de Morgan dualities. This forces the proof to follow a strategy quite different to corresponding completeness proofs for classical modal logics. The proof directly constructs a pair of distinguishing formulae for every pair of processes that are not open bisimilar.

Outline. Section 2 introduces the semantics of intuitionistic modal logic \mathcal{OM} . Section 3 recalls open bisimilarity and states the soundness and completeness results. Section 4 presents the proof of the correctness of an algorithm for generating distinguishing formulae, which is used to establish completeness of \mathcal{OM} with respect to open bisimilarity. Section 5 situates \mathcal{OM} with respect to other modal logics in the spectrum of classical and intuitionistic notions of bisimilarity, highlighting that open bisimilarity is not a canonical notion of bisimilarity congruence for the π -calculus and that other bisimilarity congruences can also be characterised by intuitionistic modal logics, including a new notion of late bisimilarity congruence, called intermediate bisimilarity, introduced for this discussion. Section 6 describes how the proof assistant Abella [BCG⁺14] was used to mechanically prove soundness of \mathcal{OM} with respect to open bisimulation. The soundness theorem (Section 6) and selected examples (Section 2 and Section 4) have been mechanised in the Abella theorem prover, and are

$\pi ::= \tau$ $\bar{x}z$ $\bar{x}(z)$ $x(z)$	(progress) (free output) (bound output) (input)	$\frac{}{\pi.P \xrightarrow{\pi} P}$ $\frac{P \xrightarrow{\pi} Q}{\nu x.P \xrightarrow{\pi} \nu x.Q} \quad x \notin \mathfrak{n}(\pi)$ $\frac{P \xrightarrow{\bar{x}z} Q}{\nu z.P \xrightarrow{\bar{x}(z)} Q} \quad x \neq z$ $\frac{P \xrightarrow{\pi} Q}{P \mid R \xrightarrow{\pi} Q \mid R} \quad \begin{array}{l} \text{if } x \in \mathfrak{bn}(\pi) \\ \text{then } x \text{ fresh for } R \end{array}$
$P ::= 0$ $\nu x.P$ $\pi.P$ $[x = y]P$ $P \mid P$ $P + P$ $!P$	(deadlock) (new) (action) (match) (par) (choice) (replication)	$\frac{P \xrightarrow{\pi} R}{P + Q \xrightarrow{\pi} R}$ $\frac{P \xrightarrow{\pi} R}{[x = x]P \xrightarrow{\pi} R}$ $\frac{P \xrightarrow{\pi} Q}{!P \xrightarrow{\pi} Q}$ $\frac{P \xrightarrow{\bar{x}(z)} P' \quad Q \xrightarrow{x(z)} Q'}{P \mid Q \xrightarrow{\tau} \nu z.(P' \mid Q')}$ $\frac{P \xrightarrow{\bar{x}y} P' \quad Q \xrightarrow{x(z)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'\{y_z\}}$ $\frac{P \xrightarrow{\bar{x}(z)} P' \quad P \xrightarrow{x(z)} Q'}{!P \xrightarrow{\tau} \nu z.(P' \mid Q')}$ $\frac{P \xrightarrow{\bar{x}y} P' \quad P \xrightarrow{x(z)} Q'}{!P \xrightarrow{\tau} P' \mid Q'\{y_z\}}$

Figure 1: Syntax and semantics of the π -calculus, omitting symmetric rules for choice and parallel composition, where $\mathfrak{n}(x(y)) = \mathfrak{n}(\bar{x}(y)) = \mathfrak{n}(\bar{x}y) = \{x, y\}$, $\mathfrak{bn}(x(y)) = \mathfrak{bn}(\bar{x}(y)) = \{y\}$, $\mathfrak{n}(\tau) = \mathfrak{bn}(\tau) = \mathfrak{bn}(\bar{x}y) = \emptyset$, and $\mathfrak{fn}(\pi) = \mathfrak{n}(\pi) \setminus \mathfrak{bn}(\pi)$. Processes $\nu x.P$, $z(x).P$ and $\bar{z}(x).P$ bind x in P .

available online.¹ Section 7 demonstrates an implementation of the algorithm automatically generating distinguishing formulae, extracted from the proof of Proposition 4.9.

2. INTRODUCING THE INTUITIONISTIC MODAL LOGIC \mathcal{OM}

We recall the syntax and labelled transition system of the π -calculus (Fig. 1). Note all the atomic symbols x, y, \dots are variables. There is no separate syntactic class for channels or names in this presentation of the π -calculus. Distinctions between the roles of variables are made by the use of binders: variables may appear as open variables, be bound by an input binder, or by a new name binder. Notably, the new name binder $\nu x.P$ indicates any occurrence of x in P is a ground name that is distinct from any other ground name and cannot be guessed by an observer unless it is provided explicitly to the observer through an output action². Variables may also be bound by an input binder, say $z(x).P$ where occurrences of variable x in P are treated as placeholders for some message (also represented by a variable) that will be received when an input on a channel represented by variable z occurs. Variables that are not bound, i.e., free variables, are critical for this call-by-need approach to the π -calculus where they are used as symbolic placeholders that range over all possible ways in which they may be instantiated.

Other features include: the deadlocked process that can do nothing, output prefixes that output a free variable or extrude a variable bound by a new name binder on a channel, the silent progress action τ , the match guard that tests for equality, parallel composition, and

¹Via <https://github.com/alwentiu/abella/tree/master/pic>

²An explanation, coming from computational security, is the new name binder represents a nonce selected from a set of nonces that is exponential in size w.r.t. some sufficiently large parameter, hence cannot be guessed by an observer.

non-deterministic choice. We also include a replication operator, which creates unboundedly many parallel copies of a process.

Transitions are labelled with four types of action ranged over by π : free outputs, bound outputs, inputs and internal progress (τ). A *free output* represents sending a free variable, whereas a *bound output* represents extruding a bounded name. We employ a late labelled transition system for the π -calculus, where the variable on the *input* action is a symbolic placeholder that need not be instantiated until after an input transition. The action τ represents some internal communication, resulting from the synchronisation of an input and output action. We use the notations $\text{bn}(E)$ and $\text{fn}(E)$ to represent the bound variables and, respectively, free variables in a given expression E (processes, actions, formulae, etc.). We assume α -conversion for bound variables.

Histories are used in the definitions of both the intuitionistic modal logic and open bisimilarity. Histories are lists representing what is known about free variables due to how they have been communicated previously to the environment. There are two types of information about variables recorded in a history. Variables x , that were bound by a new name binder and have been *extruded* using output action $\bar{a}(x)$, we call *private names*, and denote them in histories by x^o . Variables z , symbolically representing the possible messages received by an input action $a(z)$, are denoted in histories by z^i . What matters is the alternation in the history between variables representing extruded private names and variables representing symbolic inputs: if an input variable is to be instantiated with a private name, the private name must have been extruded by an earlier output in the history. This is reflected in the following definition of a respectful substitution.

Definition 2.1 (σ respects h). A history is a (dot separated) list of variables annotated with o or i . Substitution σ respects history h whenever, for all h' and h'' such that $h = h' \cdot x^o \cdot h''$, $x\sigma = x$, and for all $y \in \text{fn}(h')$, we have $y\sigma \neq x$. Here $\text{fn}(h')$ is all the variables appearing anywhere in h' .

For example, substitution $\{y_z\}$ respects history $x^i \cdot y^o \cdot z^i$, since input variable z appears after y was output, hence y was known to the environment at the time z was input. In contrast, substitution $\{y_x\}$ does not respect history $x^i \cdot y^o \cdot z^i$, since variable x was input before private name y was output.

Remark 2.2. Note that histories fulfil the role of sets of inequality constraints called *distinctions* in the original work on open bisimilarity [San96]. Although distinctions are more general than histories, it is shown in [TM10] that given a history h and its corresponding distinction D , the corresponding definitions of open bisimilarity coincide.

Histories effectively form a symbolic constraint system restricting the use of variables. It is worth noting that the effect of histories can also be achieved by maintaining a set of fresh name constraints, indicating that a private name output is fresh with respect to the free variables in the process at the moment the private name was output, as proposed for symbolic approaches to the ψ -calculus [JVP12]. In order to capture open bisimilarity using such a symbolic constraint systems, care needs to be taken to ensure that the constraint system is interpreted intuitionistically.

2.1. The semantics of the intuitionistic modal logic \mathcal{OM} . The syntax for modal logic \mathcal{OM} extends intuitionistic logic with equality and modalities, as follows.

$P \models^h \mathbf{tt}$	always holds.
$P \models^h \mathbf{ff}$	never holds.
$P \models^h x = y$	iff x and y are the same variable.
$P \models^h \phi_1 \wedge \phi_2$	iff $P \models^h \phi_1$ and $P \models^h \phi_2$.
$P \models^h \phi_1 \vee \phi_2$	iff $P \models^h \phi_1$ or $P \models^h \phi_2$.
$P \models^h \phi_1 \supset \phi_2$	iff $\forall \sigma$ respecting h , $P\sigma \models^{h\sigma} \phi_1\sigma$ implies $P\sigma \models^{h\sigma} \phi_2\sigma$.
$P \models^h \langle \alpha \rangle \phi$	iff $\exists Q$, $P \xrightarrow{\alpha} Q$ and $Q \models^h \phi$.
$P \models^h \langle \bar{a}(z) \rangle \phi$	iff $\exists Q$, $P \xrightarrow{\bar{a}(z)} Q$ and $Q \models^{h \cdot z^o} \phi$.
$P \models^h \langle a(z) \rangle \phi$	iff $\exists Q$, $P \xrightarrow{a(z)} Q$ and $Q \models^{h \cdot z^i} \phi$.
$P \models^h [\alpha] \phi$	iff $\forall \sigma$ respecting h , $\forall Q$, $P\sigma \xrightarrow{\alpha\sigma} Q$ implies $Q \models^{h\sigma} \phi\sigma$.
$P \models^h [\bar{a}(z)] \phi$	iff $\forall \sigma$ respecting h , $\forall Q$, $P\sigma \xrightarrow{\bar{a}\sigma(z)} Q$ implies $Q \models^{h\sigma \cdot z^o} \phi\sigma$.
$P \models^h [a(z)] \phi$	iff $\forall \sigma$ respecting h , $\forall Q$, $P\sigma \xrightarrow{a\sigma(z)} Q$ implies $Q \models^{h\sigma \cdot z^i} \phi\sigma$.

In each of the above, α is of the form τ or $\bar{a}b$; and z is fresh for h , and σ .

Figure 2: Semantics of the modal logic \mathcal{OM} .

Definition 2.3. The syntax for modal logic \mathcal{OM} is defined by the following grammar.

$\phi ::=$	\mathbf{tt}	top	}	intuitionistic logic
	\mathbf{ff}	bottom		
	$\phi \wedge \phi$	and		
	$\phi \vee \phi$	or		
	$\phi \supset \phi$	implies		
	$x = y$	equality		
	$\langle \pi \rangle \phi$	diamond	}	modalities
	$[\pi] \phi$	box		

Formulae of the form $[\bar{a}(x)]\phi$, $[a(x)]\phi$, $\langle \bar{a}(x) \rangle \phi$, or $\langle a(x) \rangle \phi$, bind variable x in ϕ .

The semantics of intuitionistic modal logic \mathcal{OM} , presented in Fig. 2, is defined in terms of the late labelled transition system in Fig. 1 and history respecting substitutions (Definition 2.1). Satisfaction is defined as follows, by treating all free variables as inputs in the past.

Definition 2.4 (satisfaction). Satisfaction, written $P \models \phi$, holds whenever $P \models^{x_0^i \dots x_n^i} \phi$ according to the inductive definition in Fig. 2, where $\text{fn}(P) \cup \text{fn}(\phi) \subseteq \{x_0, \dots, x_n\}$. Note $\text{fn}(P)$ and $\text{fn}(\phi)$ are simply the free variables in P and ϕ respectively.

Both modalities are closed under all respectful substitutions. However, observe in Fig. 2 there is an asymmetry in the definition of these modalities. In contrast to the box modality, the definition of the diamond modality need not be closed under all respectful substitutions.

To explain this asymmetry between the box and diamond modalities in the definitions, observe, for the diamond modality, a transition must be possible regardless of the substitution. Thus it is sufficient to check the identity substitution. For example, the following is not satisfiable.

$$[x = y]\tau \not\models \langle \tau \rangle \mathbf{tt}$$

To check the above does not hold, it is sufficient to check that $[x = y]\tau$ cannot perform a τ -transition. This is reflected in the semantics of the diamond modalities.

In contrast, for the box modality there may exist substitutions other than the identity substitution enabling a transition, hence we should consider all respectful substitutions. Perhaps the simplest example, requiring closure of box under respectful substitutions, is the following:

$$[x = y]\tau \models [\tau](x = y)$$

The above satisfaction holds since for any substitution σ such that $([x = y]\tau)\sigma \xrightarrow{\tau} 0$ it must be case that $x\sigma = y\sigma$. Thus for all such substitutions we have $0 \models x\sigma = y\sigma$ holds, as required. In contrast, observe the above process does not satisfy $[\tau]\mathbf{ff}$.

Intuitionistic hereditary establishes that all formulae are closed under respectful substitutions. We state this property of \mathcal{OM} as a lemma, since it will be used in the completeness proof later in this paper.

Lemma 2.5 (intuitionistic hereditary). *If $P \models^h \phi$ holds then $P\theta \models^{h\theta} \phi\theta$ holds for any θ respecting h .*

The hereditary lemma suggests a Kripke model for \mathcal{OM} that satisfies the usual frame conditions for intuitionistic logic: consider respectful substitutions as a binary relation between ‘worlds’, where a world is simple a set of equalities between variables. Then it can be proven that this gives rise to an intuitionistic Kripke frame. The satisfaction relation in Figure 2 can be reformulated using this notion of worlds explicitly. The interested reader may consult Appendix A for details of a Kripke semantics for \mathcal{OM} . We note that the Kripke semantics presented there is not needed to prove the main results of this paper; hence can be safely skipped.

2.2. Checking the law of excluded middle is invalidated. Given the semantics in Fig. 2, we can now formally check examples from the introduction. We claimed $\bar{a}b \mid c(x) \not\models \langle \tau \rangle \mathbf{tt} \vee \neg \langle \tau \rangle \mathbf{tt}$, where $\neg\phi$, as standard, is defined as $\phi \supset \mathbf{ff}$. This example demonstrates the law of excluded middle is invalid. Appealing to the rule for disjunction, observe that we have the following.

$$\bar{a}b \mid c(x) \not\models \langle \tau \rangle \mathbf{tt} \quad \text{and} \quad \bar{a}b \mid c(x) \not\models \neg \langle \tau \rangle \mathbf{tt}$$

The former can hold only if $\bar{a}b \mid c(x)$ is guaranteed to make a τ -transition; but such a transition is only possible under a substitution σ such that $a\sigma = c\sigma$, hence $\bar{a}b \mid c(x) \not\models \langle \tau \rangle \mathbf{tt}$. For the latter, we should consider all substitutions which enable a τ -transition; and, since such a substitution $\{c_a\}$ exists, $\bar{a}b \mid c(x) \not\models \neg \langle \tau \rangle \mathbf{tt}$.

Critically for this work, the above example illustrates that a property typically used to establish the completeness of open bisimilarity with respect to a classical modal logic breaks down. In the **classical** setting, we expect $P \not\models \phi$ if and only if $P \models \neg\phi$. However, as the above example demonstrates, there are processes, such as $\bar{a}b \mid c(x)$, that do not satisfy $\langle \tau \rangle \mathbf{tt}$, but also do not satisfy $\neg \langle \tau \rangle \mathbf{tt}$. Hence in the intuitionistic setting we **cannot** appeal to this principle of classical modal logic.

As a further example of this principle, observe the following are both unsatisfiable.

$$\tau \not\models [\tau](x = y) \quad \text{and} \quad \tau \not\models \neg[\tau](x = y)$$

The former is unsatisfiable since, under the identity substitution, $\tau \xrightarrow{\tau} 0$, but $0 \not\models x = y$. The latter is also unsatisfiable since, there is a substitution $\{y_x\}$ such that $\tau\{y_x\} \xrightarrow{\tau} 0$ still holds and $0 \models x\{y_x\} = y\{y_x\}$ holds; but clearly $0 \models \mathbf{ff}\{y_x\}$ can never hold; hence $\tau \not\models \neg[\tau](x = y)$.

As expected for an intuitionistic logic, further classical dualities break, as witnessed by the following examples of unsatisfiable formulae.

$$[x = y]\tau \not\models \neg\neg\langle\tau\rangle\mathbf{tt} \supset \langle\tau\rangle\mathbf{tt} \quad \text{and} \quad 0 \not\models \neg\neg\neg(x = y) \supset \neg(x = y)$$

Also de Morgan dualities cannot be applied. For example, **classically** we have $P \models \neg([\tau]\mathbf{ff} \wedge \langle\tau\rangle\mathbf{tt})$ if and only if $P \models \langle\tau\rangle\mathbf{tt} \vee [\tau]\mathbf{ff}$. However in the **intuitionistic** setting we have the following.

$$\bar{a}b \mid c(x) \models \neg([\tau]\mathbf{ff} \wedge \langle\tau\rangle\mathbf{tt}) \quad \text{but} \quad \bar{a}b \mid c(x) \not\models \langle\tau\rangle\mathbf{tt} \vee [\tau]\mathbf{ff}$$

Note, in this paper, intuitionistic negation is used only to explain such examples illustrating the intuitionistic nature of \mathcal{OM} . Results in subsequent sections do not depend on intuitionistic negation. However, related work [HALT18] highlights that intuitionistic negation has a role when logically characterising open bisimilarity for processes with mismatch (inequality guards, which can model the else branch of an **if-then-else** statement). Thus this formulation of \mathcal{OM} is robust for some useful extensions of the π -calculus.

3. OPEN BISIMILARITY, SOUNDNESS AND COMPLETENESS

We recall the definition of open bisimilarity. Open bisimilarity is the greatest symmetric relation closed under all respectful substitutions and labelled transitions at every step. Notice we use the history to record whenever a (symbolic) input or private output occurs.

Definition 3.1 (open bisimilarity). An open bisimulation \mathcal{R} is a symmetric relation on processes, indexed by a history h , such that: if $P \mathcal{R}^h Q$ then, the following hold:

- For all substitutions σ respecting h , we have $P\sigma \mathcal{R}^{h\sigma} Q\sigma$.
- If $P \xrightarrow{\alpha} P'$, then there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $P' \mathcal{R}^h Q'$, where α is a τ or $\bar{a}b$.
- If $P \xrightarrow{\bar{a}(x)} P'$, for x fresh for h and Q , there exists Q' such that $Q \xrightarrow{\bar{a}(x)} Q'$ and $P' \mathcal{R}^{h \cdot x^o} Q'$.
- If $P \xrightarrow{a(x)} P'$, for x is fresh for h , there exists Q' such that $Q \xrightarrow{a(x)} Q'$ and $P' \mathcal{R}^{h \cdot x^i} Q'$.

Open bisimilarity, written $P \sim Q$, is defined whenever there exists an open bisimulation \mathcal{R} such that $P \mathcal{R}^{x_0^i \dots x_n^i} Q$, where $\text{fn}(P) \cup \text{fn}(Q) \subseteq \{x_0, \dots, x_n\}$.

The main result of this paper is that, for finite π -calculus processes open bisimilarity is characterised by \mathcal{OM} formulae. This result is broken into soundness and completeness of the intuitionistic modal logic characterisation.

Theorem 3.2 (soundness). *If $P \sim Q$ then for all \mathcal{OM} formulae ϕ , $P \models \phi$ iff $Q \models \phi$.*

Theorem 3.3 (completeness). *If we have that for all \mathcal{OM} formulae ϕ , $P \models \phi$ iff $Q \models \phi$, then $P \sim Q$.*

The proof of soundness has been mechanically checked in the proof assistant Abella [BCG⁺14] using the two-level logic approach [GMN12] to reason about the π -calculus semantics specified in λ Prolog [NM88]. The proof of soundness proceeds by induction on the structure of formulae. An explanation of the soundness proof and mechanisation we defer until Section 6.

The proof of completeness is explained in detail in Section 4. Before providing proofs, we provide examples demonstrating the implications of Theorems 3.2 and 3.3. Due to soundness, if two processes are bisimilar, we cannot find a distinguishing formula that holds for one process but does not hold for the other process. Due to completeness, if two process are not

open bisimilar, then we can construct a distinguishing formula that holds for one process but does not hold for the other process. Thus \mathcal{OM} formulae can be used as a certificate that can be presented as efficiently checkable evidence to explain why two processes are not open bisimilar.

3.1. Sketch of algorithm for generating distinguishing formulae. The completeness proof, explained later in Section 4, contains an algorithm for generating distinguishing formulae for processes that are not open bisimilar. Here, we provide a sketch of the algorithm executed on key examples.

3.1.1. Example requiring intuitionistic assumptions. The algorithm proceeds over the structure of a tree of moves that show two processes are not open bisimilar – the *distinguishing strategy*. In the base case, we have a pair of processes where, under a substitution, one process can make a transition, but the other process cannot match the transition. We provide two examples of applying the base case to obtain formulae.

$[x = y]\tau \not\sim \tau$: The distinguishing strategy for these processes is as follows: the process τ leads with transition $\tau \xrightarrow{\tau} 0$, but $[x\theta = y\theta]\tau$ can make a τ -transition only when $x\theta = y\theta$. From this distinguishing strategy we generate two formulae, one biased to each process³. Since process τ leads in the distinguishing strategy, $\langle \tau \rangle \mathbf{tt}$ is a distinguishing formula biased to process τ , as follows.

$$\tau \models \langle \tau \rangle \mathbf{tt} \quad \text{and} \quad [x = y]\tau \not\models \langle \tau \rangle \mathbf{tt}$$

As remarked in the previous section, negating formula $\langle \tau \rangle \mathbf{tt}$ does not provide a formula biased to $[x = y]\tau$. To construct a formula biased towards $[x = y]\tau$, write down a box modality $[\tau]$ followed by the strongest postcondition that holds after a τ -transition is enabled, i.e. $x = y$. This gives rise to the following distinguishing formula, as required.

$$[x = y]\tau \models [\tau](x = y) \quad \text{and} \quad \tau \not\models [\tau](x = y)$$

$[x = y]\tau \not\sim 0$: For these processes the distinguishing strategy is $([x = y]\tau)\{y_x\} \xrightarrow{\tau} 0$, but 0 cannot make a τ -transition, under any substitution. To construct a distinguishing formula biased to $[x = y]\tau$, we write down $x = y$ as the weakest pre-condition under which a τ -transition is enabled, expressed as follows.

$$[x = y]\tau \models (x = y) \supset \langle \tau \rangle \mathbf{tt} \quad \text{and} \quad 0 \not\models (x = y) \supset \langle \tau \rangle \mathbf{tt}$$

To construct a formula biased to 0, write $[\tau]$ followed by \mathbf{ff} , which, vacuously, is the strongest postcondition guaranteed after 0 performs a τ -transition, since no τ -transition is enabled under any substitution. This gives us the following distinguishing formula.

$$0 \models [\tau]\mathbf{ff} \quad \text{and} \quad [x = y]\tau \not\models [\tau]\mathbf{ff}$$

Now consider the *inductive case* of an algorithm for constructing distinguishing formulae. In the inductive cases, two processes cannot be distinguished by an immediate transition. However, under some substitution, one process can make a π transition to a state, say P' , but, under the same substitution the other process can only make a corresponding π transition to reach states Q_i that are not open bisimilar to P' . This allows a distinguishing

³By “biased”, we mean the formula is satisfied by the process indicated but not by the other process considered.

formula to be inductively constructed from the distinguishing formulae for P' paired with each Q_i .

For example, consider how to construct distinguishing formulae for processes P and Q below.

$$\begin{array}{ccc} P & \triangleq \tau.[x = y]\tau + \tau + \tau.\tau & \approx & \tau + \tau.\tau \triangleq Q \\ \downarrow \tau & & & \begin{array}{c} \tau \\ \downarrow \\ 0 \end{array} \begin{array}{c} \tau \\ \searrow \\ \tau \end{array} \\ [x = y]\tau & & & \end{array}$$

Observe from the above transitions, that the process P can perform a τ -transition to a state $[x = y]\tau$ that is not bisimilar to any state reachable by a τ -transition from process Q . Process Q may perform τ -transitions either to τ or 0 . However we have just seen above that $[x = y]\tau \not\sim 0$ and $[x = y]\tau \not\sim \tau$; hence we have a distinguishing strategy.

The distinguishing strategies and distinguishing formulae for the above base cases, enable us to construct distinguishing formulae for this inductive case. The distinguishing formula satisfied by P is a *diamond* modality followed by the *conjunction* of the distinguishing formulae biased to $[x = y]\tau$ in each base case above, as follows.

$$P \models \langle \tau \rangle ([\tau](x = y) \wedge (x = y \supset \langle \tau \rangle \mathbf{tt})) \quad \text{and} \quad Q \not\models \langle \tau \rangle ([\tau](x = y) \wedge (x = y \supset \langle \tau \rangle \mathbf{tt}))$$

The distinguishing formula satisfied by Q is a *box* followed by the *disjunction* of the formulae not satisfied by $[x = y]\tau$ in each of the base cases above, as follows:

$$P \not\models [\tau] (\langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff}) \quad \text{and} \quad Q \models [\tau] (\langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff})$$

To confirm that the above are indeed distinguishing formulae for P and Q , assume for contradiction that $Q \models \langle \tau \rangle ([\tau](x = y) \wedge (x = y \supset \langle \tau \rangle \mathbf{tt}))$ holds. By definition of diamond modalities, this holds iff either $0 \models [\tau](x = y) \wedge (x = y \supset \langle \tau \rangle \mathbf{tt})$ or $\tau \models [\tau](x = y) \wedge (x = y \supset \langle \tau \rangle \mathbf{tt})$ holds. Observe that $0 \models x = y \supset \langle \tau \rangle \mathbf{tt}$ holds iff we make the additional assumption that x and y are persistently distinct, i.e., we have additional assumption $\neg(x = y)$. In addition, observe that $\tau \models [\tau](x = y)$ holds iff we make the additional assumption that $x = y$.

Indeed, by these observations, we know that the following hold:

$$\tau + \tau.\tau \models (x = y \vee \neg(x = y)) \supset \langle \tau \rangle ([\tau](x = y) \wedge (x = y \supset \langle \tau \rangle \mathbf{tt}))$$

$$\tau + \tau.\tau \models \langle \tau \rangle ([\tau](x = y) \wedge (x = y \supset \langle \tau \rangle \mathbf{tt})) \supset (x = y \vee \neg(x = y))$$

Notice that $x = y \vee \neg(x = y)$ is an instance of the law of excluded middle for equality; hence, in the **classical** setting, assuming the law of excluded middle, the formula above biased to Q is also satisfied by P ; and vice versa. Indeed there would be no distinguishing formulae for processes P and Q ; and hence in a classical framework the modal logic would be incomplete for open bisimilarity.

Similarly, in the **intuitionistic** setting, we can mechanically prove the following.

$$\tau + \tau.\tau + \tau.[x = y]\tau \models (x = y \vee \neg(x = y)) \supset [\tau] (\langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff})$$

$$\tau + \tau.\tau + \tau.[x = y]\tau \models [\tau] (\langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff}) \supset (x = y \vee \neg(x = y))$$

Since intuitionistic logics do not assume the law of excluded middle, as long as we evaluate the semantics of \mathcal{OM} in an intuitionistic framework, we have distinguishing formulae. We have formalised in Abella the above four examples of satisfaction involving the law of excluded middle.

3.1.2. *Example involving private names that are distinguishable.* Respectful substitutions ensure that a private name can never be input earlier than it was output. Consider the following processes.

$$P \triangleq \nu x.\bar{a}x.a(y).\tau \quad \approx \quad \nu x.\bar{a}x.a(y).[x = y]\tau \triangleq Q$$

These processes are not open bisimilar because P can make the following three transition steps: $\nu x.\bar{a}x.a(y).\tau \xrightarrow{\bar{a}(x)} a(y).\tau \xrightarrow{a(y)} \tau \xrightarrow{\tau} 0$. However, Q can only match the first two steps. At the third step, a base case of the distinguishing formula algorithm for $\tau \not\sim^{a^i \cdot x^o \cdot y^i} [x = y]\tau$ applies. In this case, any substitution θ respecting $a^i \cdot x^o \cdot y^i$ enabling transition $[x = y]\tau \xrightarrow{\tau} 0$ is such that $y\theta = x$ and $x\theta = x$; hence $x\theta = y\theta$. Hence we have the following formulae biased to each process.

$$[x = y]\tau \models^{a^i \cdot x^o \cdot y^i} [\tau](x = y) \quad \text{and} \quad \tau \models^{a^i \cdot x^o \cdot y^i} \langle \tau \rangle \mathbf{tt}$$

By applying inductive cases of the distinguishing formulae algorithm to the input and output actions, we obtain the following two distinguishing formulae.

$$\nu x.\bar{a}x.a(y).\tau \models \langle \bar{a}(x) \rangle \langle a(y) \rangle \langle \tau \rangle \mathbf{tt} \quad \text{and} \quad \nu x.\bar{a}x.a(y).[x = y]\tau \models [\bar{a}(x)] [a(y)] [\tau](x = y)$$

3.1.3. *Example involving private names that are indistinguishable.* In contrast to the previous example, consider the following processes where the process on the right extrudes a private name and then compares it to a free variable.

$$\nu x.\bar{a}x \quad \sim \quad \nu x.\bar{a}x.[x = a]\tau$$

These processes are open bisimilar, hence by Theorem 3.2 there is no distinguishing formula. The existence of a distinguishing formula of the form $\langle \bar{a}(x) \rangle (x = a \supset \langle \tau \rangle \mathbf{tt})$ is *prevented* by the history. For example, both $\nu x.\bar{a}x.[x = a]\tau \models \langle \bar{a}(x) \rangle (x = a \supset \langle \tau \rangle \mathbf{tt})$ and $\nu x.\bar{a}x \models \langle \bar{a}(x) \rangle (x = a \supset \langle \tau \rangle \mathbf{tt})$ hold.

To see why, observe $\nu x.\bar{a}x \models^{a^i} \langle \bar{a}(x) \rangle (x = a \supset \langle \tau \rangle \mathbf{tt})$ holds if and only if $\nu x.\bar{a}x \xrightarrow{\bar{a}(x)} 0$ and $0 \models^{a^i \cdot x^o} x = a \supset \langle \tau \rangle \mathbf{tt}$. By definition of implication, this holds if only if, for all θ respecting $a^i \cdot x^o$ and such that $x\theta = a\theta$, we have $0 \models^{a^i \cdot x^o} \langle \tau \rangle \mathbf{tt}$. However, there is no substitution θ respecting $a^i \cdot x^o$ such that $x\theta = a\theta$. By the definition of a respectful substitution, θ must satisfy $x\theta = x$ and $x \neq a\theta$, contradicting constraint $x\theta = a\theta$. Thereby $0 \models^{a^i \cdot x^o} x = a \supset \langle \tau \rangle \mathbf{tt}$ holds vacuously; hence we have that $\nu x.\bar{a}x \models^{a^i} \langle \bar{a}(x) \rangle (x = a \supset \langle \tau \rangle \mathbf{tt})$ holds as required.

4. COMPLETENESS OF OPEN BISIMILARITY WITH RESPECT TO \mathcal{QM}

In order to prove completeness we first provide a direct definition of what it means for two processes to be not open bisimilar, which we refer to as *distinguishability*. Since open bisimilarity is defined in terms of a greatest fixed point of relations satisfying a certain closure property, distinguishability is defined in terms of a least fixed point satisfying the dual property. This leads to the direct definition of distinguishability in this section.

Since distinguishability is defined in terms of a least fixed point, there is a distinguishing strategy, consisting of a finite tree of moves. We inductively define distinguishability in terms of a family of relations on processes indexed by a history \mathcal{H}_n , for $n \in \mathbb{N}$. The base case is when, for some respectful substitution one player can make a move, that move cannot be matched by the other player without applying an additional substitution. We then define

inductively, the family of relations $P \not\sim_n^h Q$ containing all processes that can be distinguished by a strategy with depth at most n , i.e., at most n moves are required to reach a pair of processes distinguished according to relation $\not\sim_0$, at which point, as just explained above, there is a process reachable by a respectful substitution that can make a move that the other process cannot match under the same substitution.

Definition 4.1 (distinguishability). The relation $\not\sim_0$ is the least relation, indexed by a history, such that $P \not\sim_0^h Q$ holds whenever there exist action π and substitution σ respecting h such that one of the following holds:

- there exists process P' such that $P\sigma \xrightarrow{\pi\sigma} P'$ and there is no Q' such that $Q\sigma \xrightarrow{\pi\sigma} Q'$, or
- there exist process Q' such that $Q\sigma \xrightarrow{\pi\sigma} Q'$ and there is no P' such that $P\sigma \xrightarrow{\pi\sigma} P'$.

In both cases, we require that if $x \in \text{bn}(\pi)$, then x is fresh for $P\sigma$, $Q\sigma$ and $h\sigma$.

Inductively, $\not\sim_{n+1}$ is the least relation extending $\not\sim_n$ such that $P \not\sim_{n+1}^h Q$ whenever for some substitution σ respecting h , one of the following holds, where, in the following, α is τ or $\bar{a}b$ and x is fresh for $P\sigma$, $Q\sigma$ and $h\sigma$:

- $\exists P'. P\sigma \xrightarrow{\alpha\sigma} P'$ and $\forall Q_i$ such that $Q\sigma \xrightarrow{\alpha\sigma} Q_i$, $P' \not\sim_n^{h\sigma} Q_i$, or
- $\exists P'. P\sigma \xrightarrow{\bar{a}\sigma(x)} P'$, and, $\forall Q_i$ such that $Q\sigma \xrightarrow{\bar{a}\sigma(x)} Q_i$, $P' \not\sim_n^{h\sigma \cdot x^o} Q_i$, or
- $\exists P'. P\sigma \xrightarrow{a\sigma(x)} P'$, and, $\forall Q_i$ such that $Q\sigma \xrightarrow{a\sigma(x)} Q_i$, $P' \not\sim_n^{h\sigma \cdot x^i} Q_i$, or
- $\exists Q'. Q\sigma \xrightarrow{\alpha\sigma} Q'$ and $\forall P_i$ such that $P\sigma \xrightarrow{\alpha\sigma} P_i$, $P_i \not\sim_n^{h\sigma} Q'$, or
- $\exists Q'. Q\sigma \xrightarrow{\bar{a}\sigma(x)} Q'$, and, $\forall P_i$ such that $P\sigma \xrightarrow{\bar{a}\sigma(x)} P_i$, $Q' \not\sim_n^{h\sigma \cdot x^o} P_i$, or
- $\exists Q'. Q\sigma \xrightarrow{a\sigma(x)} Q'$, and, $\forall P_i$ such that $P\sigma \xrightarrow{a\sigma(x)} P_i$, $Q' \not\sim_n^{h\sigma \cdot x^i} P_i$.

The relation $\not\sim$, pronounced distinguishability, is defined to be the least relation containing $\not\sim_n$ for all $n \in \mathbb{N}$, i.e. $\bigcup_{n \in \mathbb{N}} \not\sim_n$. Define $P \not\sim Q$ whenever $P \not\sim^{x_1^i \dots x_m^i} Q$ where $\text{fn}(P) \cup \text{fn}(Q) \subseteq \{x_1, \dots, x_m\}$.

It is immediate from the definition that distinguishability is symmetric.

Lemma 4.2. *The relations $\not\sim$ and $\not\sim_n$, for all $n \geq 0$, are symmetric.*

It is an established result that, for the version of the π -calculus with replication that we employ, image finiteness holds.

Lemma 4.3 (image finiteness [San95]). *For process P and action π there are finitely many P_i , up to α -conversion, such that $P \xrightarrow{\pi} P_i$.*

Distinguishability in Definition 4.1 coincides with the negation of open bisimilarity in Definition 3.1, which relies on the fact that we consider an image-finite process calculus in this work.

Lemma 4.4. *$P \not\sim Q$ (Def. 4.1) does not hold, if and only if $P \sim Q$ (Def. 3.1) holds.*

Proof. In order to establish the forward implication, we construct a relation such that $P \mathcal{R}^h Q$ whenever $\text{fn}(P) \cup \text{fn}(Q) \subseteq \text{fn}(h)$ and there does not exist n such that $P \not\sim_n^h Q$. We then show that \mathcal{R} is an open bisimulation. Symmetry of \mathcal{R} is immediate from Lemma 4.2. Below we consider the remaining cases required to show that \mathcal{R} is an open bisimulation.

Case of a respectful substitution: Assume that $P \mathcal{R}^h Q$ holds and θ respects h , and suppose for contradiction that $P\theta \mathcal{R}^{h\theta} Q\theta$ does not hold. Thus there exists n such that $P\theta \not\sim_n^{h\theta} Q\theta$. In the case $n = 0$, there exists σ respecting $h\theta$ and action $\pi\theta$ such that either: $P\theta\sigma \xrightarrow{\pi\theta\sigma} P'$ but there is no Q' such that $Q\theta\sigma \xrightarrow{\pi\theta\sigma} Q'$; or $Q\theta\sigma \xrightarrow{\pi\theta\sigma} Q'$ but there is no

P' such that $P\theta\sigma \xrightarrow{\pi\theta\sigma} P'$. Without loss of generality, consider the former case, where $P\theta\sigma \xrightarrow{\pi\theta\sigma} P'$ but there is no Q' such that $Q\theta\sigma \xrightarrow{\pi\theta\sigma} Q'$, and observe that, since $\theta \cdot \sigma$ respects h , by definition, we have $P \not\sim_0^h Q$ holds, contradicting the assumption that $P \mathcal{R}^h Q$ holds. A similar argument yields a contradiction in the case that $n > 0$. Therefore $P\theta \mathcal{R}^{h\theta} Q\theta$ holds.

Case of a free transition: Assume that $P \mathcal{R}^h Q$ and $P \xrightarrow{\alpha} P'$ hold, where $\alpha = \tau$ or $\alpha = \bar{x}z$. For contradiction, suppose there is no Q' such that $Q \xrightarrow{\alpha} Q'$. Thereby $P \not\sim_0^h Q$, contradicting the assumption that $P \mathcal{R}^h Q$; hence there is at least one Q_i such that $Q \xrightarrow{\alpha} Q_i$. By image finiteness, there are finitely many such Q_i (quotienting by α -conversion). Now, for contradiction, assume that $P' \mathcal{R}^h Q_i$ does not hold for all i . Hence for all i there exists n_i such that $P' \not\sim_{n_i}^h Q_i$. Now let $n = \max_i \{n_i\}$, and observe that for all i , $P' \not\sim_n^h Q_i$, thereby $P \not\sim_{n+1}^h Q_i$, contradicting the assumption that $P \mathcal{R}^h Q$. Thus for some i , we have $P' \mathcal{R}^h Q_i$, as required.

Case of a bound output transition: Assume that $P \mathcal{R}^h Q$ and $P \xrightarrow{\bar{x}(z)} P'$ hold, where z is fresh for P , Q and h . For contradiction, suppose there is no Q' such that $Q \xrightarrow{\bar{x}(z)} Q'$. Thereby $P \not\sim_0^h Q$, contradicting the assumption that $P \mathcal{R}^h Q$; hence for some Q_i we have $Q \xrightarrow{\bar{x}(z)} Q_i$. By image finiteness, there are finitely many such Q_i (quotienting by α -conversion). Now assume that for all i , we have $P' \mathcal{R}^{h \cdot x^o} Q_i$ does not hold. Hence for all i , there exists n_i such that $P' \not\sim_{n_i}^{h \cdot x^o} Q_i$. Now let $n = \max_i \{n_i\}$, and observe that for all i , $P' \not\sim_n^{h \cdot x^o} Q_i$, hence $P \not\sim_{n+1}^h Q_i$, contradicting the assumption that $P \mathcal{R}^h Q$. Thus for some i , we have $P' \mathcal{R}^{h \cdot x^o} Q_i$, as required.

Case of an input transition: This is almost identical to the case for bound output transitions.

Thus \mathcal{R} is an open bisimulation. Now assume $P \not\sim Q$ does not hold, hence we have $P \mathcal{R}^{x_1^i \dots x_m^i} Q$ holds, where $\text{fn}(P) \cup \text{fn}(Q) \subseteq \{x_1, \dots, x_m\}$; thereby, by definition of open bisimilarity, $P \sim Q$ holds, as required. The converse direction is immediate from the definitions. \square

Notice that distinguishability in Def. 4.1 requires that a distinguishing strategy is finite, and finite distinguishing strategies are sufficient to distinguish processes whose transition systems are image finite. In a more general setting where we do not have image finiteness (such as for weak variants of open bisimilarity or where infinitely branching processes are permitted in an extended process language) such a finite notion of distinguishability would not suffice. In such a setting without image finiteness, in order for the negation of bisimilarity and distinguishability to coincide, distinguishability must be defined in terms of transfinite induction. Hence we would be required to extend \mathcal{OM} with features offering additional distinguishing power, for example with least and greatest fixed points as in the μ -calculus [Koz83], or with finitely supported conjunctions as in related work on nominal transition systems [PBE⁺20]. Here we stick to the setting where we have image finiteness, hence, by Lemma 4.4, we can rely on finite distinguishing strategies. An investigation of the more general setting where we cannot rely on image finiteness is proposed as future work.

4.1. Preliminaries. For the completeness proof that follows, we require the following terminology for substitutions, and abbreviations for formulae. These are mainly standard.

Definition 4.5. Composition of substitutions σ and θ is defined such that $x(\sigma \cdot \theta) = (x\sigma)\theta$, for all x . For substitutions σ and θ , $\sigma \leq \theta$ holds whenever there exists σ' such that $\sigma \cdot \sigma' = \theta$. For a finite substitution $\sigma = \{z_1/x_1\} \cdots \{z_n/x_n\}$ formula $[\sigma]\phi$ abbreviates formula $(x_n = z_n) \supset \dots (x_1 = z_1) \supset \phi$. For finite set of formulae ϕ_i , formula $\bigvee_i \phi_i$ abbreviates $\phi_1 \vee \dots \vee \phi_n$, where the empty disjunction is \mathbf{ff} . Similarly $\bigwedge_i \phi_i$ abbreviates $\phi_1 \wedge \dots \wedge \phi_n$, where the empty conjunction is \mathbf{tt} .

We require the following technical lemmas. The first unfolds the definition of $[\sigma]\phi$ in Def. 4.5 above. The second is required in inductive cases involving bound output and input. The third is a monotonicity property for transitions, along with side conditions for the bookkeeping of bound names that may appear in labels.

Lemma 4.6. *If for all θ respecting h and $\sigma \leq \theta$, it holds that $P\theta \models^{h\theta} \phi\theta$, then $P \models^h [\sigma]\phi$ holds.*

Lemma 4.7. *If $\sigma \cdot \theta$ respects h , then θ respects $h\sigma$.*

Lemma 4.8 (monotonicity). *If $P \xrightarrow{\pi} Q$ then $P\theta \xrightarrow{\pi\theta} Q\theta$, for all θ such that if $x \in \text{bn}(\pi)$ and $y\theta = x$ then $x = y$.*

Proof. The proof follows directly by induction on the derivation of rules. We consider two cases only. Consider the base case for input transitions.

$$\frac{}{a(x).P \xrightarrow{a(x)} P}$$

Clearly, if $x\theta = x$, then we have $a\theta(x).P\theta = (a(x).P)\theta$ and the following labelled transition is enabled, as required.

$$\frac{}{a\theta(x).P\theta \xrightarrow{a\theta(x)} P\theta}$$

Consider the inductive case for the following rule.

$$\frac{P \xrightarrow{\bar{a}x} Q}{\nu x.P \xrightarrow{\bar{a}(x)} Q}$$

By the induction hypothesis, for all θ , we have $P\theta \xrightarrow{\bar{a}\theta x\theta} Q\theta$. Hence if, in addition, $x\theta = x$, we have $P\theta \xrightarrow{\bar{a}\theta x} Q\theta$ and $\nu x.P\theta = (\nu x.P)\theta$ and so the following labelled transition is enabled, as required.

$$\frac{P\theta \xrightarrow{\bar{a}\theta x} Q\theta}{\nu x.P\theta \xrightarrow{\bar{a}\theta(x)} Q\theta} \quad \square$$

We comment on the generality of the results in this work. Remarkably, monotonicity of the labelled transitions (Lemma 4.8) is the only property we require of the process model in order to prove completeness, other than image finiteness (which, as discussed previously, could even be lifted in an extended logic). Thus it would be possible to make the results of this paper more abstract, by ranging over any process model where the labelled transition system satisfies image finiteness (Lemma 4.3), monotonicity of the labelled transition system (Lemma 4.8), and, furthermore, has the same labels as for the late transition system of the π -calculus (i.e., of the form τ , $\bar{x}(z)$, $\bar{x}z$ or $x(z)$, where x and z are variables). When viewed in terms of Kripke semantics in Appendix A, monotonicity is essentially the first compatibility condition of Plotkin and Sterling [PS86]; which is the “zig-zag” between the modal accessibility relation (here defined by the labelled transition along with its history)

and the intuitionistic information partial ordering (here defined by substitutions respecting a history), that is sufficient to guarantee intuitionistic hereditary.

We have chosen to stick a more concrete formulation of the π -calculus rather than being more abstract, for two reasons. Firstly, we can clearly provide concrete examples in a single process language familiar to a large audience. Secondly, results obtained using such a more abstract approach should be treated with caution, since it does not immediately cover many richer process calculi, for which the definition of open bisimilarity must be modified. For example, when extending our results to the π -calculus with mismatch [HALT18], we require that the definition of open bisimilarity is extended to allow for the retrospective creation of fresh private names in the past, when our supply of private names runs out, otherwise open bisimilarity is not a congruence. Going further, for the applied π -calculus [ABF18] or ψ -calculi [BJPV11], to define open bisimilarity the labels employed are of a more general form, so both the labels and the definition of open bisimilarity change in order to conservatively extend open bisimilarity to these settings while retaining the property that open bisimilarity is a congruence [HM21]. Thus it is a deliberate choice that, in this paper, we do not provide results at the maximum level of abstraction or generality that we know how to provide; instead, we seek to clearly map out the key novel ideas in a widely understood process language.

4.2. Algorithm for distinguishing formulae. The direct definition of distinguishability (Definition 4.1) provides us with a tree of substitutions and actions forming a strategy showing that two processes are not open bisimilar. The following proposition shows that \mathcal{OM} formulae are sufficient to describe such strategies. For any strategy that distinguishes two processes, we can construct *distinguishing* formula in \mathcal{OM} . A distinguishing formula holds for one process but not for the other process. In the proof of the following proposition, at each step we construct two distinguishing formulae, one biased to the process on the left and another biased to the process on the right, since we cannot simply construct a formula biased to one process and negate it to obtain a formula biased to the other process, which is the standard trick used since the early days of classical Hennessy-Milner logics [HM85]. We discussed in Section 2, why the left biased formula cannot be simply obtained by negating the right biased formula and vice versa; both must be constructed simultaneously and may be unrelated by negation.

Proposition 4.9. *If $P \not\sim Q$ then there exists ϕ_L such that $P \models \phi_L$ and $Q \not\models \phi_L$.*

Proof. Since $\not\sim$ is defined by a least fixed point over a family of relations $\not\sim_n$, if $P \not\sim^h Q$, there exists n such that $P \not\sim_n^h Q$, so we can proceed by induction on the depth of a distinguishing strategy.

In the base case, assume $P \not\sim_0^h Q$, hence by definition, for substitution σ respecting h , $P\sigma \xrightarrow{\pi\sigma} P'$, for $x \in \text{bn}(\pi)$, x is fresh for $P\sigma$, $Q\sigma$ and $h\sigma$, such that there is no Q' such that $Q\sigma \xrightarrow{\pi\sigma} Q'$. It is sufficient to consider only this base case without loss of generality, since the other case is symmetric (Q leads and P cannot follow).

We require the following property concerning substitutions enabling $\pi\theta$ -transitions from $Q\theta$, exploiting the observation that necessarily each such θ must induce an additional equality that was not yet enabled by σ . There exist finitely many pairs of variables x_j and y_j in $\text{fn}(P) \cup \text{fn}(Q) \cup \text{fn}(\pi)$ such that $x_j\sigma$ and $y_j\sigma$ are distinct, and, for any R and substitution θ respecting h , if $Q\theta \xrightarrow{\pi\theta} R$ there exists j such that $x_j\theta = y_j\theta$. To see why, assume for

contradiction that there is some θ respecting h such that $Q\theta \xrightarrow{\pi\theta} R$ but there is no x and y in $\text{fn}(P) \cup \text{fn}(Q) \cup \text{fn}(\pi)$ such that $x\sigma$ and $y\sigma$ are distinct, and $x\theta = y\theta$. Stated otherwise, for all x and y in $\text{fn}(P) \cup \text{fn}(Q) \cup \text{fn}(\pi)$ if $x\theta = y\theta$ then $x\sigma = y\sigma$, which is precisely the definition of a function, i.e., there is a substitution, say θ' , defined on the range of θ such that θ' maps $z\theta$ to $z\sigma$. In that case, $\theta \cdot \theta' = \sigma$ on $\text{fn}(P) \cup \text{fn}(Q) \cup \text{fn}(\pi)$; and hence, by Lemma 4.8, $Q\theta\theta' \xrightarrow{\pi\theta\theta'} R\theta'$ contradicting the initial assumption for the base case that no transition $Q\sigma \xrightarrow{\pi\sigma} Q'$ exists for any Q' .

In this case, there are two distinguishing formulae $[\sigma]\langle\pi\rangle\mathbf{tt}$ and $[\pi]\bigvee_j(x_j = y_j)$ biased to P and Q respectively. There are four cases to check to confirm that these are distinguishing formulae.

Case $P \models^h [\sigma]\langle\pi\rangle\mathbf{tt}$: Consider all θ respecting h such that $\sigma \leq \theta$. By definition there exists θ' such that $\sigma \cdot \theta' = \theta$, so since $P\sigma \xrightarrow{\pi\sigma} P'$, by Lemma 4.8, $P\theta \xrightarrow{\pi\theta} P'\theta'$. Thereby, since $P'\theta' \models^{h'} \mathbf{tt}$ holds, $P\theta \models^{h\theta} \langle\pi\theta\rangle\mathbf{tt}$. Hence, by Lemma 4.6, $P \models^h [\sigma]\langle\pi\rangle\mathbf{tt}$.

Case $Q \not\models^h [\sigma]\langle\pi\rangle\mathbf{tt}$: For contradiction, assume $Q \models^h [\sigma]\langle\pi\rangle\mathbf{tt}$. Since σ respects h and $\sigma \leq \sigma$, by Lemma 4.6, $Q \models^h [\sigma]\langle\pi\rangle\mathbf{tt}$ holds only if $Q\sigma \models^{h\sigma} \langle\pi\sigma\rangle\mathbf{tt}$ holds; which holds only if there exists Q' such that $Q\sigma \xrightarrow{\pi\sigma} Q'$, contradicting the assumption no such Q' exists. Thereby $Q \not\models^h [\sigma]\langle\pi\rangle\mathbf{tt}$.

Case $Q \models^h [\pi]\bigvee_j(x_j = y_j)$: Consider substitutions θ respecting h and Q' such that $Q\theta \xrightarrow{\pi\theta} Q'$. It must be the case that there exists j such that $x_j\theta = y_j\theta$, thereby $Q' \models^{h\theta} x_j\theta = y_j\theta$ holds; hence clearly $Q' \models^{h\theta} \bigvee_j(x_j = y_j)\theta$ holds. Hence $Q \models^h [\pi]\bigvee_j(x_j = y_j)$.

Case $P \not\models^h [\pi]\bigvee_j(x_j = y_j)$: Assume for contradiction $P \models^h [\pi]\bigvee_j(x_j = y_j)$. This holds iff for all processes S and substitutions θ respecting h , $P\theta \xrightarrow{\pi\theta} S$ implies $S \models^{h'} \bigvee_j(x_j = y_j)\theta$. Since we know that σ respects h and $P\sigma \xrightarrow{\pi\sigma} P'$, for some h'' , we have $P' \models^{h''} \bigvee_j(x_j = y_j)\sigma$. This holds only if for some j , $P' \models^{h''} x_j\sigma = y_j\sigma$; hence, $x_j\sigma = y_j\sigma$ for some j , which contradicts the assumption that $x_j\sigma$ and $y_j\sigma$ are distinct. Thereby $P \not\models^h [\pi]\bigvee_j(x_j = y_j)$.

Now consider the inductive cases. Given P, Q , if $P \not\sim_{n+1}^h Q$, up to symmetry of $\not\sim_{n+1}^h$, there are three cases to consider, for some substitution σ respecting h , where α is either τ or $\bar{a}b$, where x is fresh for $P\sigma, Q\sigma$ and $h\sigma$:

- $P\sigma \xrightarrow{\alpha\sigma} P'$ and for all Q_i such that $Q\sigma \xrightarrow{\alpha\sigma} Q_i, P' \not\sim_n^{h\sigma} Q_i$.
- $P\sigma \xrightarrow{\bar{a}\sigma(x)} P'$, and, for all Q_i such that $Q\sigma \xrightarrow{\bar{a}\sigma(x)} Q_i, P' \not\sim_n^{h\sigma \cdot x^\circ} Q_i$.
- $P\sigma \xrightarrow{a\sigma(x)} P'$, and, for all Q_i such that $Q\sigma \xrightarrow{a\sigma(x)} Q_i, P' \not\sim_n^{h\sigma \cdot x^i} Q_i$.

We consider the second case above involving bound output only, the other two cases are similar – differing only in the accounting for respectful substitutions according to Def. 2.1.

For $P\sigma \xrightarrow{\bar{a}\sigma(x)} P'$, by Lemma 4.3, there exist finitely many Q_i such that $Q\sigma \xrightarrow{\bar{a}\sigma(x)} Q_i$. For each i , since $P' \not\sim_n^{h\sigma \cdot x^\circ} Q_i$, by the induction hypothesis, there exist ϕ_i^L and ϕ_i^R such that $P' \models^{h\sigma \cdot x^\circ} \phi_i^L\sigma$ and $Q_i \not\models^{h\sigma \cdot x^\circ} \phi_i^L\sigma$ and $P' \not\models^{h\sigma \cdot x^\circ} \phi_i^R\sigma$ and $Q_i \models^{h\sigma \cdot x^\circ} \phi_i^R\sigma$.

We require the following property, referred to later using \dagger . There are finitely many pairs of variables x_j and y_j selected from $\text{fn}(P) \cup \text{fn}(Q) \cup \{a\}$ such that $x_j\sigma$ and $y_j\sigma$ are distinct, and, for any substitution θ respecting h (note we can apply α -conversion to ensure that θ also respects $h \cdot x^\circ$), such that $\sigma \leq \theta$, and for any S such that, $Q\theta \xrightarrow{\bar{a}\theta(x)} S$ then either: for some i , we have $S \models^{h\theta \cdot x^\circ} \phi_i^R\theta$, or there exists some j such that $x_j\theta = y_j\theta$.

To see why such pairs of variables x_j and y_j can be constructed, suppose, for contradiction, that they cannot be constructed in general. Hence, there would exist substitution ρ respecting

$h \cdot x^o$, where $\sigma \leq \rho$, and process S such that: $Q\rho \xrightarrow{\overline{a\rho}(x)} S$, there is no i such that $S \models^{h\rho \cdot x^o} \phi_i^R \rho$, and also there is no pair of variables u and v in $\text{fn}(P) \cup \text{fn}(Q) \cup \{a\}$ such that $u\sigma$ and $v\sigma$ are distinct and $u\rho = v\rho$. Hence $\rho \leq \sigma$; therefore, there exists ρ' respecting $h\rho \cdot x^o$ such that $\rho \cdot \rho' = \sigma$ and hence, by Lemma 4.8, $Q\sigma \xrightarrow{\overline{a\sigma}(x)} S\rho'$, where $S\rho' = Q_i$ for some i . Since, $\rho \leq \sigma$ and $\sigma \leq \rho$, we know ρ' has an inverse, say σ' . Now since, by Lemma 4.8, $Q\sigma\sigma' \xrightarrow{\overline{a\sigma\sigma'}(x)} Q_i\sigma'$, we have $Q\rho \xrightarrow{\overline{a\rho}(x)} Q_i\sigma'$; and, since $Q_i \models^{h\sigma \cdot x^o} \phi_i^R \sigma$, by Lemma 2.5, we have $Q_i\sigma' \models^{h\rho \cdot x^o} \phi_i^R \rho$, i.e., $S \models^{h\rho \cdot x^o} \phi_i^R \rho$, contradicting the assumption no such i exists.

From the above, it is possible to construct distinguishing formulae $[\sigma] \langle \overline{a}(x) \rangle \bigwedge_i \phi_i^L$ and $[\sigma] [\overline{a}(x)] \left(\bigvee_i \phi_i^R \vee \bigvee_j (x_j = y_j) \right)$. There are four cases to consider to verify these are distinguishing formulae.

Case $P \models^h [\sigma] \langle \overline{a}(x) \rangle \bigwedge_i \phi_i^L$: Consider all θ such that $\sigma \leq \theta$, θ respects h , and without loss of generality x is fresh for θ , i.e., for $y \in \text{dom}(\theta)$ and $x \notin y\theta$. By definition, there exists θ' such that $\sigma \cdot \theta' = \theta$. Now since $\sigma \cdot \theta'$ respects h , by Lemma 4.7, θ' respects $h\sigma$ hence since $x \notin \text{dom}(\theta')$ and $x \notin \text{fn}(h\sigma\theta')$, θ' respects $h\sigma \cdot x^o$. Thereby since θ' respects $h\sigma \cdot x^o$ and also $P' \models^{h\sigma \cdot x^o} \phi_i^L \sigma$ holds, by Lemma 2.5, it holds that $P'\theta' \models^{h\theta \cdot x^o} \phi_i^L \theta$. The above holds for all i , hence it holds that $P'\theta' \models^{h\theta \cdot x^o} \bigwedge_i \phi_i^L \theta$. Now, since $P\sigma \xrightarrow{\overline{a\sigma}(x)} P'$, by Lemma 4.8, since x is fresh, $P\theta \xrightarrow{\overline{a\theta}(x)} P'\theta'$ holds; and hence $P\theta \models^{h\theta} (\langle \overline{a}(x) \rangle \bigwedge_i \phi_i^L) \theta$ holds. Thereby, by Lemma 4.6, $P \models^h [\sigma] \langle \overline{a}(x) \rangle \bigwedge_i \phi_i^L$ holds.

Case $Q \not\models^h [\sigma] \langle \overline{a}(x) \rangle \bigwedge_i \phi_i^L$: Assume for contradiction that $Q \models^h [\sigma] \langle \overline{a}(x) \rangle \bigwedge_i \phi_i^L$ holds. Since σ respects h and $\sigma \leq \sigma$, by Lemma 4.6, the above assumption holds only if $Q\sigma \models^{h\sigma} (\langle \overline{a}(x) \rangle \bigwedge_i \phi_i^L) \sigma$ holds. Now $Q\sigma \models^{h\sigma} \langle \overline{a\sigma}(x) \rangle \bigwedge_i \phi_i^L \sigma$ holds only if there exists Q' such that $Q\sigma \xrightarrow{\overline{a\sigma}(x)} Q'$ and $Q' \models^{h\sigma \cdot x^o} \bigwedge_i \phi_i^L \sigma$, which holds only if $Q' \models^{h\sigma \cdot x^o} \phi_i^L \sigma$ for all i . Notice that $Q' = Q_k$ for some k , and therefore $Q_k \models^{h\sigma \cdot x^o} \phi_k^L \sigma$; but it was assumed that $Q_k \not\models^{h\sigma \cdot x^o} \phi_k^L \sigma$ leading to a contradiction. Therefore $Q \not\models^h [\sigma] \langle \overline{a}(x) \rangle \bigwedge_i \phi_i^L$.

Case $Q \models^h [\sigma] [\overline{a}(x)] \left(\bigvee_i \phi_i^R \vee \bigvee_j (x_j = y_j) \right)$: Fix Q' and θ respecting h , such that $\sigma \leq \theta$, and $Q\theta \xrightarrow{\overline{a\theta}(x)} Q'$. Above, in †, we established that, in this scenario, either: for some ℓ , we have $Q' \models^{h\theta \cdot x^o} \phi_\ell^R \theta$, or there exists some k such that $x_k\theta = y_k\theta$. In the case where, for some k , $x_k\theta = y_k\theta$, we have $Q' \models^{h\theta \cdot x^o} x_k\theta = y_k\theta$ holds. Hence in either case we have $Q' \models^{h\theta \cdot x^o} \left(\bigvee_i \phi_i^R \vee \bigvee_j (x_j = y_j) \right) \theta$, by definition of disjunction. Thereby, by definition, $Q \models^h [\sigma] [\overline{a}(x)] \left(\bigvee_i \phi_i^R \vee \bigvee_j (x_j = y_j) \right)$ holds.

Case $P \not\models^h [\sigma] [\overline{a}(x)] \left(\bigvee_i \phi_i^R \vee \bigvee_j (x_j = y_j) \right)$: Let us assume for contradiction that $P \models^h [\sigma] [\overline{a}(x)] \left(\bigvee_i \phi_i^R \vee \bigvee_j (x_j = y_j) \right)$. Since σ respects h , $\sigma \leq \sigma$, and $P\sigma \xrightarrow{\overline{a\sigma}(x)} P'$, the previous assumption can hold only if $P' \models^{h\sigma \cdot x^o} \left(\bigvee_i \phi_i^R \vee \bigvee_j (x_j = y_j) \right) \sigma$. This holds only if, for some i , $P' \models^{h\sigma \cdot x^o} \phi_i^R \sigma$, or, for some j , $P' \models^{h\sigma \cdot x^o} (x_j\sigma = y_j\sigma)$. However, for all i , $P' \not\models^{h\sigma \cdot x^o} \phi_i^R \sigma$; and also, for all j , we have $x_j\sigma$ and $y_j\sigma$ are distinct and $P' \not\models^{h\sigma \cdot x^o} (x_j\sigma = y_j\sigma)$, leading to a contradiction in either case. Thereby $P \not\models^h [\sigma] [\overline{a}(x)] \left(\bigvee_i \phi_i^R \vee \bigvee_j (x_j = y_j) \right)$.

By induction we have established that, for any history h , processes P and Q , and any n , if $P \not\sim_n^h Q$ then we can construct ϕ_L such that $P \models^h \phi_L$ and $Q \not\models^h \phi_L$; and also we can construct ϕ_R such that $Q \models^h \phi_R$ and $P \not\models^h \phi_R$. The result then follows by observing that,

since $\not\sim$ is the least relation containing all $\not\sim_n$ whenever $P \not\sim Q$; there exists n such that $P \not\sim_n^{x_1^i \dots x_n^i} Q$ and, where $\text{fn}(P) \cup \text{fn}(Q) \subseteq \{x_1^i, \dots, x_n^i\}$; for which, there is ϕ^L such that $P \models^{x_1^i \dots x_n^i} \phi^L$ and $Q \not\models^{x_1^i \dots x_n^i} \phi^L$; and also ϕ^R such that $Q \models^{x_1^i \dots x_n^i} \phi^R$ and $P \not\models^{x_1^i \dots x_n^i} \phi^R$. Hence, by Definition 2.4, indeed $P \models \phi_L$, $Q \not\models \phi_L$, $Q \models \phi_R$ and $P \not\models \phi_R$ as required. \square

4.3. The proof of completeness. Combining Proposition 4.9 with Lemma 4.4 yields immediately the completeness of \mathcal{OM} with respect to open bisimilarity. Completeness (Theorem 3.3) establishes that the set of all pairs of processes that have the same set of distinguishing formulae is an open bisimulation. The proof can now be stated as follows.

Proof of Theorem 3.3: Assume that for finite processes P and Q , for all formulae ϕ , $P \models \phi$ iff $Q \models \phi$. Now for contradiction suppose that $P \sim Q$ does not hold. By Lemma 4.4, $P \not\sim Q$ must hold. Hence by Proposition 4.9 there exists ϕ_L such that $P \models \phi_L$ but $Q \not\models \phi_L$, but we assumed at the beginning that $P \models \phi_L$ holds iff $Q \models \phi_L$ holds, leading to a contradiction. Thereby $P \sim Q$. \square

4.4. Example runs of distinguishing formula algorithm. We provide further examples of processes that are not open bisimilar that illustrate subtle aspects of the algorithm. In particular, these examples illustrate various scenarios where postconditions are required.

4.4.1. Multiple postconditions and postconditions in an inductive step. The following example leads to multiple postconditions. Consider the following distinguishable processes.

$$[x = y]\tau + [w = z]\tau \quad \not\sim \quad \tau$$

Observe that clearly $\tau \xrightarrow{\tau} 0$ but $([x = y]\tau + [w = z]\tau)\theta \xrightarrow{\tau}$ only if $x\theta = y\theta$ or $w\theta = z\theta$. Thus, $[x = y]\tau + [w = z]\tau \models [\tau]((x = y) \vee (w = z))$ is a distinguishing formula biased to the left process, while $\tau \models \langle \tau \rangle \mathbf{tt}$ is biased to the right.

We consider now an example where postconditions are required in the inductive case of the distinguishing formulae algorithm. However, firstly observe that $\bar{a}a + \bar{b}b \not\sim \bar{a}a$ are distinguished since $\bar{a}a + \bar{b}b \xrightarrow{\bar{b}b} 0$, but process $\bar{a}a$ can only make a $\bar{b}b$ transition under a substitution such that $a = b$. Hence we have the distinguishing formulae $\bar{a}a + \bar{b}b \models \langle \bar{b}b \rangle \mathbf{tt}$ and $\bar{a}a \models [\bar{b}b](a = b)$. Now consider the following.

$$P \triangleq \tau.(\bar{a}a + \bar{b}b) + [x = y]\tau.\bar{a}a \quad \not\sim \quad \tau.(\bar{a}a + \bar{b}b) + \tau.\bar{a}a \triangleq Q$$

To distinguish these processes, $Q \xrightarrow{\tau} \bar{a}a$ leads, a move which can only be matched by $P \xrightarrow{\tau} \bar{a}a + \bar{b}b$.

To construct formulae distinguishing P from Q we use the following ingredients: the distinguishing formulae constructed for the sub-problem $\bar{a}a + \bar{b}b \not\sim \bar{a}a$; and the observation that, for substitutions θ such that $x\theta = y\theta$, there is an additional τ -transitions enabled: $P\theta \xrightarrow{\tau} \bar{a}a$. These observations lead us to the following distinguishing formula biased to the process P in the left above, consisting of a box τ followed by a disjunction comprised of the distinguishing formula for $\bar{a}a + \bar{b}b \not\sim \bar{a}a$ biased to the process $\bar{a}a + \bar{b}b$ on the left, and the postcondition $x = y$, which must hold after the additional τ transition is enabled.

$$\tau.(\bar{a}a + \bar{b}b) + [x = y]\tau.\bar{a}a \models [\tau](\langle \bar{b}b \rangle \mathbf{tt} \vee x = y)$$

The distinguishing formula biased to the process Q on the right above is “diamond τ ” followed by the distinguishing formula for $\bar{a}a + \bar{b}b \not\sim \bar{a}a$ biased to the process $\bar{a}a$, as follows.

$$\tau.(\bar{a}a + \bar{b}b) + \tau.\bar{a}a \models \langle \tau \rangle [\bar{b}b](a = b)$$

4.4.2. *Formulae generated by substitutions applied to labels.* In some cases, substitutions applied to labels play a role when generating distinguishing formulae. For a minimal example consider the following distinguishable processes: $\bar{a}a \not\sim \bar{a}b$. A distinguishing strategy is where process $\bar{a}b$ makes a $\bar{a}b$ transition, which cannot be matched by $\bar{a}a$. However, we do have transition $(\bar{a}a)\sigma \xrightarrow{(\bar{a}b)\sigma} 0$ for any substitution such that $a\sigma = b\sigma$, leading to distinguishing formula $[\bar{a}b](a = b)$ biased to $\bar{a}a$. Notice substitution σ is applied to both the process and the label.

For a trickier example consider the following processes.

$$\nu b.\bar{a}b.a(x).[x = b]\bar{x}x \not\sim \nu b.\bar{a}b.a(x).\bar{x}x$$

After two actions, the above problem reduces to base case $[x = b]\bar{x}x \not\sim^{a^i \cdot b^o \cdot x^i} \bar{x}x$, where $\bar{x}x$ can perform a $\bar{x}x$ action, but $[x = b]\bar{x}x$ cannot. However, $([x = b]\bar{x}x)\{b_x\} \xrightarrow{\bar{x}x\{b_x\}} 0$ does hold, and furthermore $\{b_x\}$ respects $a^i \cdot b^o \cdot x^i$. From these observations we can construct a distinguishing formula biased to the left as follows.

$$\nu b.\bar{a}b.a(x).[x = b]\bar{x}x \models [a(b)][a(x)][\bar{x}x](x = b)$$

4.4.3. *Alternative forms for distinguishing formulae.* Note our algorithm copes with sub-optimal distinguishing strategies. To understand this, consider the distinguishing strategy for the following processes that are clearly not open bisimilar.

$$[x = y]\tau \quad \not\sim \quad \tau.[x = y]\tau$$

There is an obvious optimal distinguishing strategy: $\tau.[x = y]\tau \xrightarrow{\tau} [x = y]\tau$, which cannot be matched by $[x = y]\tau$. By appealing to the base case of the distinguishing formulae algorithm, we obtain two distinguishing formulae $[\tau](x = y)$ and $\langle \tau \rangle \mathbf{tt}$ biased each respective process.

There are however, sub-optimal, distinguishing strategies. Under substitution $\{y_x\}$, the process on the left has transition $([x = y]\tau)\{y_x\} \xrightarrow{\tau} 0$, which can be matched, under the same substitution, by $(\tau.[x = y]\tau)\{y_x\} \xrightarrow{\tau} [y = y]\tau$. Now 0 and $[y = y]\tau$ are distinguished, since $[y = y]\tau \xrightarrow{\tau} 0$ whereas 0 is deadlocked. By applying the algorithm in Proposition 4.9, we obtain the formula $x = y \supset [\tau]\langle \tau \rangle \mathbf{tt}$ biased to the process on the right, which is indeed distinguishing.

As a further example of alternative distinguishing formulae, consider the following processes.

$$[x = y]\tau.\tau + \tau \quad \not\sim \quad \tau.\tau + \tau$$

The following is a distinguishing formula biased to the left process: $[\tau][\tau](x = y)$. However, this is different from the left-biased formula $[\tau]([\tau]\mathbf{ff} \vee (x = y))$ generated by the algorithm. Thus, there exist alternative distinguishing formulae ... and alternative algorithms. In particular, the above two examples highlight the open question of whether restricting ourselves to minimal substitutions in the distinguishing strategy allows us to simplify slightly the formulae in the inductive case of the distinguishing formula algorithm, thereby avoiding generating formulae such as $x = y \supset [\tau]\langle \tau \rangle \mathbf{tt}$ featuring a prefix $x = y \supset$ before a box modality.

4.4.4. *A more elaborate example.* This example forces the use of postconditions regardless of whether we construct a distinguishing formula biased to the process on the left or on the right. Consider the following processes.

$$P \triangleq \tau + \tau.(\tau.\tau + \tau) + \tau.[x = y](\tau.[u = v]\tau + \tau.\tau + \tau) \quad Q \triangleq P + \tau.[x = y](\tau.\tau + \tau)$$

The processes above are distinguished by the following strategy. Firstly, the process Q moves, as follows; for which there are three moves P can perform.

$$\begin{array}{ccc} Q & & P \\ \downarrow \tau & \swarrow \tau & \downarrow \tau \quad \searrow \tau \\ [x = y](\tau.\tau + \tau) & 0 & \tau.\tau + \tau \quad [x = y](\tau.[u = v]\tau + \tau.\tau + \tau) \end{array}$$

This leads to three sub-problems, for which we know already the distinguishing strategies and formulae. Note, to distinguish $[x = y](\tau.\tau + \tau)$ from $[x = y](\tau.[u = v]\tau + \tau.\tau + \tau)$, there is a switch in the process that leads.

From the above strategy, we can construct the following distinguishing formulae.

$$Q \models \langle \tau \rangle ((x = y \supset \langle \tau \rangle \mathbf{tt}) \wedge [\tau](x = y) \wedge [\tau](\langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff}))$$

$$P \models [\tau](\langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff} \vee (x = y \supset \langle \tau \rangle ((u = v \supset \langle \tau \rangle \mathbf{tt}) \wedge [\tau](u = v))))$$

Notice this example nests a classic example, explained previously, inside itself. The absence of the law of excluded middle is essential for the existence of distinguishing formulae in this example.

5. SITUATING \mathcal{OM} WITH RESPECT TO OTHER MODAL LOGICS CHARACTERISING BISIMILARITIES

Open bisimilarity is not the only bisimilarity congruence. We consider here the relationship between the intuitionistic modal logic for open bisimilarity presented in this work and other modal logics. In doing so, we clarify why we introduce \mathcal{OM} rather than taking an intuitionistic variant of an established modal logic. We check that \mathcal{OM} has a classical counterpart characterising late bisimilarity. Also, we note open bisimilarity is not the only notion of bisimilarity that is a congruence relation. We provide a sharp picture explaining where open bisimilarity sits in relation to other notions of bisimilarity; notably the bisimilarity congruences *open barbed bisimilarity* [SW01] and a newly introduced late variant of open barbed bisimilarity which we call *intermediate bisimilarity*, both of which can also be characterised by intuitionistic modal logics.

5.1. **Why a new modal logic \mathcal{OM} , rather than an intuitionistic variant of \mathcal{LM} ?** A classical logic characterising *late bisimilarity*, called \mathcal{LM} for “(\mathcal{L}) late modality with (\mathcal{M}) match,” was provided by Milner, Parrow, and Walker [MPW93]. \mathcal{LM} differs from \mathcal{OM} in two significant ways. Firstly, \mathcal{LM} is classical: a classical semantics is induced due to the fact that all grounded inputs are considered immediately after an input action, where variables appearing free represent distinct ground names, hence an input is either equal to another ground message or it is not. Secondly, the late input box modality is defined differently, involving an existential quantification over substitutions. Moving to an intuitionistic variant of \mathcal{LM} , this gives rise to the following variant of the box input modality.

$$P \models [a(x)]^L \phi \text{ iff } \forall \sigma \text{ respecting } h, \forall Q, P\sigma \xrightarrow{a\sigma(x)} Q \implies \exists x, \text{ such that } Q \models^{h\sigma} \phi.$$

In the semantics of \mathcal{OM} , we deliberately use a universally quantified box input modality, recalled bellow; rather than existentially quantified box input modality used in \mathcal{LM} above.

$$P \models [a(x)]\phi \text{ iff } \forall \sigma \text{ respecting } h, \forall Q, P\sigma \xrightarrow{a\sigma(x)} Q \implies Q \models^{h\sigma \cdot x^i} \phi.$$

Recall from Sec. 6, in the box input modality of \mathcal{OM} immediately above, the x^i appended to the history has the effect of $\forall x$ appearing immediately after the implication (made explicit in Fig. 5).

Hence, due to the differences in quantification for the box input modality, \mathcal{OM} is not quite an intuitionistic variant of \mathcal{LM} . The carefully selected box input modality in \mathcal{OM} is necessary for our construction of distinguishing formulae in the completeness proof for the characterisation of open bisimilarity using \mathcal{OM} . To understand why, consider the following processes that are not open bisimilar.

$$a(x).\tau + a(x) + a(x).[x = a]\tau \not\sim a(x).\tau + a(x)$$

For the above processes, our algorithm for distinguishing formulae, Proposition 4.9, correctly generates the following \mathcal{OM} formula biased to the right:

$$a(x).\tau + a(x) + a(x).[x = a]\tau \not\models [a(x)](\langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff}) \text{ and } a(x).\tau + a(x) \models [a(x)](\langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff})$$

If we were to use an intuitionistic variant of the input box modality of \mathcal{LM} , as suggested in related work [TM10], both processes satisfy the above formulae modified with a late box input modality $[a(x)]^L(\langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff})$. To see why $a(x).\tau + a(x) + a(x).[x = a]\tau \models [a(x)]^L(\langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff})$ holds, observe that for the transition $a(x).\tau + a(x) + a(x).[x = a]\tau \xrightarrow{a(x)} [x = a]\tau$ there exists $\{a_x\}$ such that $([x = a]\tau)\{a_x\} \models \langle \tau \rangle \mathbf{tt} \vee [\tau] \mathbf{ff}$.

Also, note the formula $[a(x)]^L((x = a \wedge [\tau] \mathbf{ff}) \vee \langle \tau \rangle \neg(x = a))$ fails in intuitionistic \mathcal{LM} for both processes, despite being distinguishing for these processes in classical \mathcal{LM} . Although, if we use triple negation the formula becomes distinguishing in intuitionistic \mathcal{LM} , as follows:

$$a(x).\tau + a(x) \models [a(x)]^L(((x = a \wedge [\tau] \mathbf{ff}) \vee \langle \tau \rangle \neg\neg\neg(x = a))) \quad (5.1)$$

5.1.1. *Example where box input is necessary.* Notice, for the above example processes, formula $\langle a(x) \rangle (x = a \supset \langle \tau \rangle \mathbf{tt} \wedge [\tau] (x = a))$ is a distinguishing formula biased to the left, in both intuitionistic \mathcal{LM} and \mathcal{OM} , since the diamond input modalities are the same in both intuitionistic modal logics. For a more sophisticated example where box input modalities are necessary, regardless of the bias, consider the follow processes that are not open bisimilar.

$$P \triangleq a(x) + a(x).(a(y) + a(y).\tau) + a(x).[x = v](a(y) + a(y).\tau + a(y).[y = w]\tau)$$

$$Q \triangleq P + a(x).[x = v](a(y) + a(y).\tau)$$

Consider the distinguishing strategy. The first move must be made by Q , which can be matched by P in three ways.

$$\begin{array}{c} Q \\ \downarrow a(x) \\ [x = v](a(y) + a(y).\tau) \end{array} \quad \begin{array}{c} a(x) \quad P \quad a(x) \\ \swarrow \quad \downarrow a(x) \quad \searrow \\ a(y) + a(y).\tau \quad [x = v](a(y) + a(y).\tau + a(y).[y = w]\tau) \end{array}$$

Each of the processes reachable by a $a(x)$ -transition from P are not open bisimilar to the process reachable from Q indicated above. The interesting case is the third process reached from P above. After applying substitution $\{v_x\}$, the process on the right leads the distinguishing strategy.

$$\begin{array}{ccc}
\begin{array}{c} [v = v](a(y) + a(y).\tau) \\ \swarrow \quad \searrow \\ 0 \quad \quad \tau \\ \text{\scriptsize } a(y) \quad \text{\scriptsize } a(y) \end{array} & & \begin{array}{c} [v = v](a(y) + a(y).\tau + a(y).[y = w]\tau) \\ \quad \quad \quad \downarrow \text{\scriptsize } a(y) \\ [y = w]\tau \end{array}
\end{array}$$

The necessity of box input modalities is due to the switch from Q leading initially to the other process leading for the second input in the distinguishing strategy. From the above distinguishing strategy the following formula biased to P can be constructed.

$$P \models [a(x)]([a(y)]\mathbf{ff} \vee \langle a(y) \rangle \mathbf{tt} \vee (x = v \supset \langle a(y) \rangle (y = w \supset \langle \tau \rangle \mathbf{tt} \wedge [\tau](y = w))))$$

For a formula biased to Q we obtain the following.

$$Q \models \langle a(x) \rangle (x = v \supset \langle a(y) \rangle \mathbf{tt} \wedge [a(y)](x = v) \wedge [a(y)](\langle \tau \rangle \mathbf{tt} \vee [\tau]\mathbf{ff}))$$

Neither of the above formulae would be distinguishing if, instead of the open box modalities of \mathcal{OM} , the late box modalities of \mathcal{LM} were employed.

5.1.2. Discussion on intuitionistic \mathcal{LM} . We have formalised the intuitionistic variant of \mathcal{LM} in Abella. The language of formulae for \mathcal{LM} replaces the “basic” box input modality of \mathcal{OM} with the following “late” box input modality:

$$\text{Type } \Box_L^\downarrow \quad \mathbf{n} \rightarrow (\mathbf{n} \rightarrow \mathbf{o}') \rightarrow \mathbf{o}'.$$

The clauses for the satisfaction relation (encoded as the predicate `satLM`) are those for \mathcal{OM} (Figure 5) without the “basic” box operator, but with the following clause for `satLM` :

$$\text{satLM } P (\Box_L^\downarrow X A) := \forall Q, \{\text{oneb } P (\text{dn } X) Q\} \rightarrow \exists z, \text{sat } (Q z) (A z).$$

The example involving triple negation above (5.1) has been verified using this formalisation of intuitionistic \mathcal{LM} .

Related work [TM10] suggested that intuitionistic \mathcal{LM} characterises open bisimilarity. Unfortunately, the completeness proof in that work is flawed since they appeal to classical principles that are not valid in the intuitionistic setting. This oversight is rectified in the current paper, by a more direct construction in the completeness proof and by the careful choice of input modalities in \mathcal{OM} , explained in this section. Note however the example above involving triple negation, suggests the problem of whether intuitionistic \mathcal{LM} characterises open bisimilarity remains an open problem. To offer an intuition for triple negation: it can be regarded as an explicit test that variables are “not equal yet,” in contrast to single negation indicating that variables are never going to be equal.

5.2. What about the classical counterpart to \mathcal{OM} ? A criteria an intuitionistic modal logic is expected to satisfy is that, when the law of excluded middle is induced, we obtain a meaningful classical logic [Sim94]. Fortunately, this criteria holds for \mathcal{OM} – the classical counterpart to \mathcal{OM} characterises late bisimilarity. For convenience we, recall a definition of late bisimilarity.

Definition 5.1 (late bisimilarity). A late bisimulation \mathcal{R} is a symmetric relation, such that, whenever $P \mathcal{R} Q$:

- If $P \xrightarrow{\alpha} P'$ then there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $P' \mathcal{R} Q'$.
- If $P \xrightarrow{\bar{a}(x)} P'$ then there exists Q' such that $Q \xrightarrow{\bar{a}(x)} Q'$ and $P' \mathcal{R} Q'$.
- If $P \xrightarrow{a(x)} P'$ then there exists Q' such that $Q \xrightarrow{a(x)} Q'$ and, for all x , $P' \mathcal{R} Q'$.

Late bisimilarity \sim_L is the greatest late bisimulation.

$$\begin{array}{ll}
P \Vdash_L \mathbf{tt} & \text{and} \quad P \Vdash_L x = x \quad \text{always hold.} \\
P \Vdash_L \phi_1 \supset \phi_2 & \text{iff} \quad P \Vdash_L \phi_1 \implies P \Vdash_L \phi_2. \\
P \Vdash_L \langle \alpha \rangle \phi & \text{iff} \quad \exists Q, P \xrightarrow{\alpha} Q \text{ and } Q \Vdash_L \phi. \\
P \Vdash_L \langle a(z) \rangle \phi & \text{iff} \quad \exists Q, P \xrightarrow{a(z)} Q \text{ and, } \forall y, Q\{y/z\} \Vdash_L \phi\{y/z\}. \\
P \Vdash_L [a(z)] \phi & \text{iff} \quad \forall Q, P \xrightarrow{a(z)} Q \implies \forall z, Q \Vdash_L \phi.
\end{array}$$

Figure 3: Semantics of “classical \mathcal{OM} ”, where α is τ , $\bar{a}b$ or $\bar{a}(z)$.

A direct semantics of classical \mathcal{OM} , in the style of Milner, Parrow and Walker [MPW93], is presented in Fig. 3. Observe histories are not employed in the classical semantics since inputs are instantiated eagerly, immediately after performing an input transition (see the clauses for the input labelled transitions). Also, missing operators (conjunction, disjunction, and $[a]\phi$) are derivable using classical negation; whereas in an intuitionistic modal logic they have independent interpretations. Classical \mathcal{OM} characterises late bisimilarity.

Corollary 5.2 (characterisation). *$P \sim_L Q$ if and only if, for all \mathcal{OM} formulae ϕ , we have $P \Vdash_L \phi$ iff $Q \Vdash_L \phi$, according to the classical semantics for \mathcal{OM} in Fig. 3.*

Proof. Observe that the definition of $\langle a(x) \rangle \phi$ in Figure 3 coincides with the late modality $\langle a(x) \rangle^L \phi$ in \mathcal{LM} . Also observe that, classically, $\neg[a(x)]\neg\phi$ is the “basic” diamond modality of Milner, Parrow and Walker [MPW93]; hence classical \mathcal{OM} is classical \mathcal{LM} extended with “basic” modalities. That original paper on modal logics for the π -calculus establishes that, classical \mathcal{LM} characterises late bisimilarity, and also \mathcal{LM} extend with basic modalities has the same expressive power at \mathcal{LM} . \square

Historically, Milner, Parrow and Walker emphasised late equivalence (the greatest congruence contained in late bisimilarity) rather than late bisimilarity in the original paper on the π -calculus [MPW92]. This is because late equivalence is closed under input prefixes. Late equivalence can be defined by restricting to late bisimulations closed under substitutions; and its characteristic modal logic can be defined in a similar way, as follows.

Definition 5.3. P is late equivalent to Q , written $P \sim_L Q$, whenever there exists a late bisimulation \mathcal{R} such that for all σ , $P\sigma \mathcal{R} Q\sigma$. Define $P \models_L \phi$ whenever for all σ , $P\sigma \Vdash_L \phi\sigma$.

Quantifying over all substitutions, combined with the distinct name assumption, means that we check late bisimilarity with respect to all combinations of equalities and inequalities between free variables. As such, late equivalence is not a bisimilarity; but is a late bisimulation. Using the above, we obtain a characteristic logic for late equivalence, using \mathcal{OM} formulae.

Corollary 5.4. *$P \sim_L Q$ if and only if, for all ϕ , $P \models_L \phi$ iff $Q \models_L \phi$.*

Proof. This follows immediately from Corollary 5.2 and the following facts. For a fresh for P , Q and ϕ , $a(x_1) \dots a(x_n).P \sim_L a(x_1) \dots a(x_n).Q$ if and only if $P \sim_L Q$, and $P \models_L \phi$ if and only if $a(x_1) \dots a(x_n).P \Vdash_L \langle a(x_1) \rangle \dots \langle a(x_n) \rangle \phi$ where $\text{fn}(P) \cup \text{fn}(Q) \cup \text{fn}(\phi) \subseteq \{x_1, \dots, x_n\}$. \square

As for open bisimilarity, $[x = y]\tau$ and 0 are not late equivalent. This is because $([x = y]\tau)\{x/y\}$ and $0\{x/y\}$ are clearly not late bisimilar. Two distinguishing formulae in this logic are defined as follows: $P \models_L x = y \supset \langle \tau \rangle \mathbf{tt}$ and $Q \models_L [\tau] \mathbf{ff}$.

The point is, if we take \mathcal{OM} and induce the law of excluded middle, we obtain a logic, defined by \models_L , characterising late equivalence. If we then, in addition, enforce the distinct name assumption, we obtain a logic, defined by $\models_{=L}$, characterising late bisimilarity.

5.3. A sharpened picture of the spectrum of bisimilarity congruences. We emphasise here that open bisimilarity is not the only bisimilarity congruence. A notable, strictly coarser, bisimilarity congruence for the π -calculus is *open barbed bisimilarity* [SW01]. Notions of open barbed bisimilarity are, by definition, the greatest bisimilarity congruences. We give the strong formulation of open barbed bisimilarity here, consistent with the rest of the paper (of course, the weak formulation of open barbed bisimilarity is coarser).

Definition 5.5. Process P has a barb x , written $P \downarrow x$, whenever $P \xrightarrow{x(z)} P'$ or $P \xrightarrow{\bar{x}y} P'$ or $P \xrightarrow{\bar{x}(z)} P'$. An *open barbed bisimulation* \mathcal{R} is a symmetric relation such that, whenever $P \mathcal{R} Q$ we have:

- If $P \xrightarrow{\tau} P'$ then there exists Q' such that $Q \xrightarrow{\tau} Q'$ and $P' \mathcal{R} Q'$.
- If $P \downarrow x$ then $Q \downarrow x$.
- For contexts $\mathcal{C}\{ \cdot \}$, we have $\mathcal{C}\{ P \} \mathcal{R} \mathcal{C}\{ Q \}$.

Open barbed bisimilarity \simeq is the greatest open barbed bisimulation.

Unlike open bisimilarity, open barbed bisimilarity is incomparable with late bisimilarity. A key example that holds for open barbed bisimilarity, but not for late bisimilarity is the following.

$$\nu k. \bar{a}k.(a(x).P + a(x)) \quad \simeq \quad \nu k. \bar{a}k.(a(x).[x = k]P + a(x).P + a(x))$$

There is however a (minimal) refinement of open barbed bisimilarity forbidding the above property, defined as follows.

Definition 5.6. An intermediate bisimulation \mathcal{R} is a symmetric relation indexed by a set of variables, such that, whenever $P \mathcal{R}^\mathcal{E} Q$ the following hold:

- If $\{x, y\} \cap \mathcal{E} = \emptyset$ then $P\{y_x\} \mathcal{R}^\mathcal{E} Q\{y_x\}$.
- If $P \xrightarrow{\alpha} P'$ then there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $P' \mathcal{R}^\mathcal{E} Q'$.
- If $P \xrightarrow{\bar{a}(x)} P'$ then there exists Q' such that $Q \xrightarrow{\bar{a}(x)} Q'$ and $P' \mathcal{R}^{\mathcal{E}, x} Q'$.
- If $P \xrightarrow{a(x)} P'$ then there exists Q' such that $Q \xrightarrow{a(x)} Q'$ and, for all x , $P' \mathcal{R}^\mathcal{E} Q'$.

Intermediate bisimilarity \sim_I is the greatest intermediate bisimulation.

Intermediate bisimilarity, a secondary contribution of this paper, defined above, sits between open bisimilarity, late equivalence and open barbed bisimilarity. Intermediate bisimilarity is a congruence, hence is sound with respect to open barbed bisimilarity. Strictness of this inclusion follows since $\nu k. \bar{a}k.(a(x).\tau + a(x))$ and $\nu k. \bar{a}k.(a(x).[x = k]\tau + a(x).\tau + a(x))$ are distinguished by intermediate bisimilarity, as witnessed by the following strategy.

$$\begin{array}{ccc}
 \nu k. \bar{a}k.(a(x).\tau + a(x)) & \not\sim_I & \nu k. \bar{a}k.(a(x).\tau + a(x)) \\
 \downarrow \bar{a}(k) & & \downarrow \bar{a}(k) \\
 a(x).\tau + a(x) & \not\sim_I^k & a(x).[x = k]\tau + a(x).\tau + a(x) \\
 \swarrow a(x) \quad \searrow a(x) & & \downarrow a(x) \\
 \tau & & [x = k]\tau \\
 & & \downarrow a(x) \\
 & & 0
 \end{array}$$

Observe that, there are two cases to check at this point: either we apply substitution $\{k_x\}$, in which case we have $0 \not\sim_I^k [k = k]\tau\{k_x\}$; or we apply any other substitution for

x , say σ , in which case $[x\sigma = k]\tau$ is deadlocked, hence $\tau \not\sim_I^k [x\sigma = k]\tau$. In contrast, as remarked previously, these processes are open barbed bisimilar.

Intermediate bisimilarity is strictly coarser than open bisimilarity. To see why, observe the following processes are equivalent according to intermediate bisimilarity.

$$\nu k.\bar{a}k.a(x).(\tau + \tau.\tau + \tau.[x = k]\tau) \sim_I \nu k.\bar{a}k.a(x).(\tau + \tau.\tau)$$

In contrast, for open bisimilarity, there is a distinguishing strategy for the same pair of processes, as witnessed by the following formula in \mathcal{OM} .

$$\nu k.\bar{a}k.a(x).(\tau + \tau.\tau + \tau.[x = k]\tau) \models \langle a(k) \rangle \langle a(x) \rangle \langle \tau \rangle (x = k \supset \langle \tau \rangle \mathbf{tt} \wedge [\tau](x = k))$$

The difference is, when constructing an open bisimulation, we can proceed with the first τ transition without deciding whether $x = k$ or $x \neq k$. In contrast, intermediate bisimilarity forces this decision immediately after x is input.

It is important to note that we are not advocating that intermediate bisimilarity should be used in preference to open bisimilarity. What we are emphasising here is that open bisimilarity does not hold a canonical status as a bisimilarity congruence sound with respect to late bisimilarity. Indeed, there is a spectrum bisimilarities between open bisimilarity and open barbed bisimilarity.

A picture of part of the spectrum surrounding open bisimilarity is provided in Fig. 4. To complete the picture in Fig. 4, note that related work [HALT18] introduced a modal logic characterising open barbed bisimilarity called intuitionistic \mathcal{FM} – the intuitionistic counterpart to a classical modal logic characterising early bisimilarity. That paper emphasises the merits of open barbed bisimilarity due to its more objective definition, and, more importantly still, its coarser granularity suitable for verifying privacy properties. Open barbed bisimilarity can be used to verify properties of protocols that make use of else branches to maintain the privacy of honest participants; whereas open bisimilarity fails to verify such scenarios, instead discovering spurious attacks. This is due to the intuitionistic

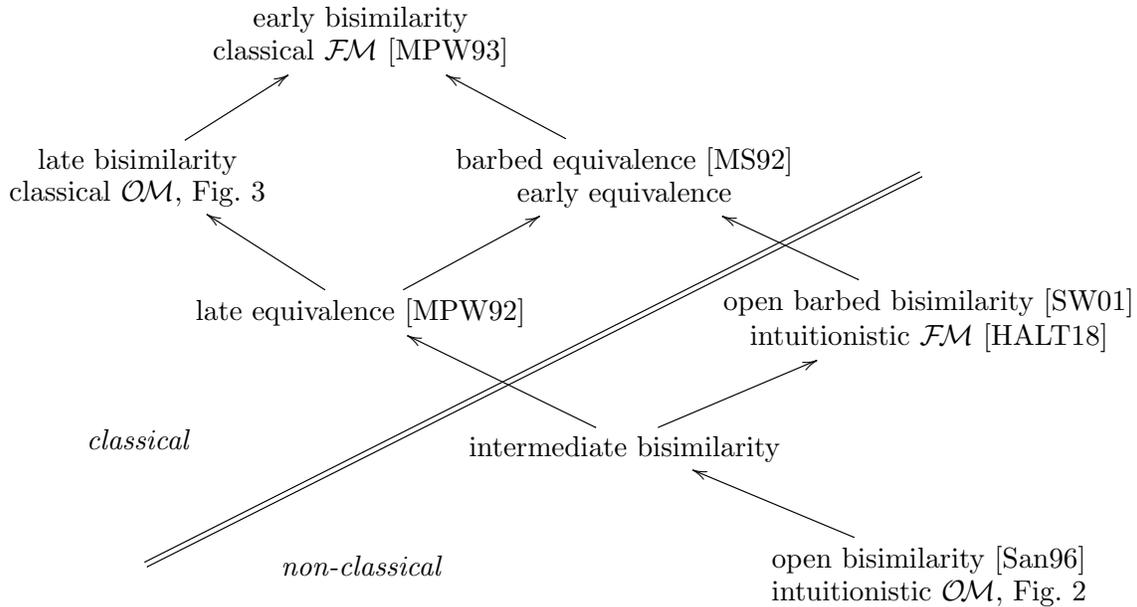


Figure 4: The line between classical and non-classical notions of bisimilarity.

nature of open bisimilarity, which, as we have seen, assumes the absence of the law of excluded middle everywhere; whereas open barbed bisimilarity verifies more properties since it induces the law of excluded middle for private names only. Related work [Hor18], elaborates on this perspective, making a case for why variants of open barbed bisimilarity and \mathcal{FM} are, respectively, the notions of bisimilarity congruence and intuitionistic modal logic suited to the applied π -calculus; rather than open bisimilarity and \mathcal{OM} .

5.4. Related work: an alternative logic formalised in Nominal Isabelle. Parrow et al. [PBE⁺15] provided a general proof of the soundness and completeness of logical equivalence for various modal logics with respect to corresponding bisimulations. The proof is parametric on properties of substitutions, which can be instantiated for a range of bisimulations. Moreover, their proof is mechanised using Nominal Isabelle. The conference version [PBE⁺15] sketches how to instantiate their abstract framework for open bisimilarity for a fragment of the π -calculus without input prefixes. A forthcoming journal version [PBE⁺20] generalises their methodology such that open bisimilarity with input prefixes may also be handled.

Stylistically, our intuitionistic modal logic is quite different from an instantiation of the abstract framework of Parrow et al. for open bisimilarity. Their framework, is classical and works by syntactically restricting “effect” modalities in formulae, depending on the type of bisimulation. Their effects represent substitutions that reach worlds permitted by the type of bisimulation. In contrast, the modalities of the intuitionistic modal logic \mathcal{OM} in this paper are syntactically closer to long established modalities for the π -calculus [MPW93]; differing instead in their intuitionistic interpretation. An explanation for the stylistic differences is that for every intuitionistic logic, such as the intuitionistic modal logic in this work, there should be a corresponding classical modal logic based on an underlying Kripke semantics. Such a Kripke semantics would reflect the accessible worlds, as achieved by the syntactically restricted effect modalities in the abstract classical framework instantiated for open bisimilarity.

6. MECHANISING THE SOUNDNESS PROOF IN ABELLA

Abella [BCG⁺14] is a proof assistant based on intuitionistic logic that supports both inductive and coinductive reasoning over logical specifications of operational semantics for languages that contain binding structures, such as the π -calculus. In particular, Abella is well-suited for reasoning involving operational semantics specified in the higher-order logic programming language λ Prolog [MN12]. The formalisation of the modal logic \mathcal{OM} in this section is built on top of existing work on the formalisation of the π -calculus and bisimulation based on the higher-order abstract syntax (HOAS) approach [BGM⁺07, TM10, BCG⁺14]. We present the coinductive definition of open bisimilarity (Section 6.1) and the semantics of the modal logic \mathcal{OM} (Section 6.2) formalised in Abella, leading up to our mechanised proof of the soundness theorem (Theorem 3.2).

Interestingly, the proof of soundness (Theorem 3.2) is quite abstract since it can be proven without defining a specific language of process terms and their labelled transition system rules, since the proof only looks at the labels and makes the implicit assumption that transitions satisfy monotonicity. Thus although it is not required for the main theorems of this paper, we none-the-less, for a self-contained presentation also recall an established λ Prolog specification of the π -calculus at the end of this section (Section 6.3), which can be used as the basis of tooling.

```

1 Specification "finite-pic". % load the finite pi-calc. spec. in Fig.6
2
3 CoDefine bisim : p → p → prop % open bisimulation
4 by bisim P Q
5   := (∀ A P1, {one P A P1} → ∃ Q1, {one Q A Q1} ∧ bisim P1 Q1)
6   ∧ (∀ X M, {oneb P (dn X) M} → ∃ N, {oneb Q (dn X) N} ∧ ∀ z, bisim (M z) (N z))
7   ∧ (∀ X M, {oneb P (up X) M} → ∃ N, {oneb Q (up X) N} ∧ ∀ z, bisim (M z) (N z))
8   ∧ (∀ A Q1, {one Q A Q1} → ∃ P1, {one P A P1} ∧ bisim Q1 P1)
9   ∧ (∀ X N, {oneb Q (dn X) N} → ∃ M, {oneb P (dn X) M} ∧ ∀ z, bisim (N z) (M z))
10  ∧ (∀ X N, {oneb Q (up X) N} → ∃ M, {oneb P (up X) M} ∧ ∀ z, bisim (N z) (M z)).
11
12 Kind o' type. % syntax of the modal logic
13 Type tt, ff o'.
14 Type Υ, ∧, ⊃ o' → o' → o'.
15 Type = n → n → o'.
16 Type □, ◇ a → o' → o'.
17 Type □↑, ◇↑, □↓, ◇↓ n → (n → o') → o'.
18
19 Define sat : p → o' → prop % semantics of the modal logic
20 by sat P tt
21 ; sat P (Υ A B) := sat P A ∧ sat P B
22 ; sat P (∧ A B) := sat P A ∧ sat P B
23 ; sat P (⊃ A B) := sat P A → sat P B
24 ; sat P (≡ X Y) := X = Y
25 ; sat P (□ X A) := ∀ P1, {one P X P1} → sat P1 A
26 ; sat P (◇ X A) := ∃ P1, {one P X P1} ∧ sat P1 A
27 ; sat P (□↑X A) := ∀ Q, {oneb P (up X) Q} → ∀ z, sat (Q z) (A z)
28 ; sat P (◇↑X A) := ∃ Q, {oneb P (up X) Q} ∧ ∀ z, sat (Q z) (A z)
29 % basic input modality (see Section 5.1 for related discussion)
30 ; sat P (□↓X A) := ∀ Q, {oneb P (dn X) Q} → ∀ z, sat (Q z) (A z)
31 % late input modality (see Section 5.1 for related discussion)
32 ; sat P (◇↓X A) := ∃ Q, {oneb P (dn X) Q} ∧ ∀ z, sat (Q z) (A z).

```

Figure 5: A coinductive definition of open bisimulation and an inductive definition of the modal logic \mathcal{OM} in Abella.

6.1. Coinductive definition of open bisimulation in Abella's reasoning logic. Open bisimulation relation `bisim` is coinductively defined in Fig. 5. The relation `bisim` is an Abella encoding of the open bisimulation relation \mathcal{R} in Definition 3.1 from Section 3. Lines 5, 6, and 7 correspond to the latter three of the four bullet items in Definition 3.1, which state the closure property under every pairwise bisimulation step where P leads and Q follows. Lines 8, 9, and 10 are symmetric cases where Q leads and P follows. Curly braces (e.g., $\{\text{one } P \ A \ P_1\}$) are used for referring to the object logic proposition (i.e., λ Prolog proposition) from the reasoning logic of Abella. The λ Prolog relation $\text{one} : p \rightarrow a \rightarrow p \rightarrow o$ (see Section 6.3 for further details), when applied to three arguments, becomes an object logic proposition $\text{one } P \ A \ P_1 : o$. In order refer to such λ Prolog propositions from Abella's reasoning logic,

we use curly braces to convert a λ Prolog proposition (\circ) into a reasoning logic proposition (**prop**). For instance, $\{\text{one } P \text{ A } P_i\} : \text{prop}$. Abella's reasoning logic is richer than the object logic. It supports coinductive definitions, which we used to define **bisim**. It also supports nominal quantification (∇), which will be discussed shortly, in addition to universal (\forall) and existential (\exists) quantifications.

The first bullet item in Definition 3.1 states that open bisimulation must be closed under all substitutions that respect the history. In the definition of **bisim**, Abella guarantees this closure property under respectful substitutions for free. Let us first demonstrate how histories are being handled in the relation **bisim**, in order to explain how the closure property under respectful substitutions is ensured in Abella. Consider a trivial bisimulation over identical processes, illustrated using both Abella and mathematical notations as follows:⁴

$$\frac{\frac{\forall \mathbf{x}, \nabla \mathbf{z}, \forall \mathbf{y}, \text{bisim } 0 \ 0 \quad \vdots}{\forall \mathbf{x}, \nabla \mathbf{z}, \text{bisim } (x(\mathbf{y}).0) \ (x(\mathbf{y}).0)} \quad \vdots}{\forall \mathbf{x}, \text{bisim } (\nu z.\bar{x}z.x(\mathbf{y}).0) \ (\nu z.\bar{x}z.x(\mathbf{y}).0)} \quad \frac{\frac{0 \sim^{x^i.z^o.y^i} 0 \quad \vdots}{x(\mathbf{y}).0 \sim^{x^i.z^o} x(\mathbf{y}).0} \quad \vdots}{\nu z.\bar{x}z.x(\mathbf{y}).0 \sim^{x^i} \nu z.\bar{x}z.x(\mathbf{y}).0}$$

Even for identical processes without nondeterministic constructs, the bisimulation tree has at least two branches for each node because either one of the two sides may take a leading step to be followed by the other side. Here, let us focus on the leftmost branches where the left process leads. The environment of quantified variables for the Abella relation **bisim** grows after each bisimulation step. Growing the environment exactly corresponds to growing the history. The bound output step extends the environment with $\nabla \mathbf{z}$ in Abella, which corresponds to extending the history with z^o . The input step extends the environment with $\forall \mathbf{y}$ in Abella, which corresponds to extending the history with y^i . These quantified variables come from the definition of **bisim** in Fig. 5, more specifically, from lines 6 and 7.

Recall the definition of respectful substitution (Definition 2.1) from Section 2. An input variable in the history adds no restriction to the respectfulness of a substitution. An extruded private name in the history adds a restriction such that respectful substitutions should not unify the output variable with any variable that precedes the output variable. The nabla quantifier (∇) in Abella coincides with such a notion of restriction. Nabla quantified variables are guaranteed to be fresh names with respect to all the previously introduced names. For instance, consider the environment $\forall \mathbf{x}, \nabla \mathbf{z}, \forall \mathbf{y}, \dots$. Abella ensures that \mathbf{z} cannot occur free in \mathbf{x} , hence, \mathbf{x} cannot be unified with \mathbf{z} ; however, \mathbf{y} can be unified with \mathbf{z} because \mathbf{y} is introduced after \mathbf{z} .

Intuitively, universal quantification represents all possible substitutions over universally quantified variables. For example, consider $\forall \mathbf{x}, \text{pred } \mathbf{x}$. Proving this in Abella means that the predicate **pred** holds for all possible substitutions over \mathbf{x} . Together with nabla quantification, the notion of all possible respectful substitutions can be represented by the environment of quantified variables in Abella. In summary, the list of universal and nabla quantified variables before the **bisim** relation in Abella not only transcribes the history but also represents all possible respectful substitutions.

⁴Here, mathematical notations of process terms are used on the left side, instead of the actual λ Prolog embeddings, as well as on the right side. The embeddings of process terms are provide in Section 6.3.

6.2. Embedding of \mathcal{OM} in Abella and the soundness proof. The latter part of Fig. 5 is an embedding of the syntax and semantics of \mathcal{OM} introduced earlier in Section 2.1. Recall the stylistic difference between Abella and mathematical notations for the process syntax – prefixes of free actions, bound output actions, and input actions are defined as three different syntactic constructs in the Abella definitions (see Fig. 5). There are similar stylistic difference regarding \mathcal{OM} formulae in the Abella embedding (Fig. 5) and the notation in Section 2.1. There are three formulae constructs for each kind of modality. For instance, \Box , \Box^\uparrow and \Box^\downarrow are the three different syntactic constructs of the box modality for free actions, bound output actions, and input actions, respectively. Similarly, there are three constructs for the diamond modality.⁵

Recall that histories on the bisimulation relation are transcribed as universal and nabla quantified variables in Abella and that closure under respectful substitutions holds for free in Abella. Similarly, histories in the semantics of \mathcal{OM} are handled in exactly the same manner in Abella and enjoy the closure properties regarding respectful substitutions. For example, $\forall x, \nabla z, \forall y, \text{sat } (P \ x \ y \ z) \ \phi$ corresponds to $\forall \sigma$ respecting $x^i \cdot z^o \cdot y^i$, $P(x, y, z) \models^{x^i \cdot z^o \cdot y^i} \phi$. The relation sat in Fig. 5 is an embedding of \mathcal{OM} (\models) in Abella. There is no explicit handing of substitutions in the semantics of the definition of the sat relation because they are handled by Abella automatically.

We mechanised the proof of soundness of open bisimilarity with respect to \mathcal{OM} by proving the following theorem.

```
Theorem bisim_sat :  $\forall P \ Q \ F, \text{form } F \rightarrow$ 
                     $\text{bisim } P \ Q \rightarrow ((\text{sat } P \ F \rightarrow \text{sat } Q \ F) \wedge (\text{sat } Q \ F \rightarrow \text{sat } P \ F)).$ 
```

In the above, $\text{form} : \mathcal{o}' \rightarrow \text{prop}$ is an inductive predicate for well-formed \mathcal{OM} formulae, defined as follows:

```
Define form :  $\mathcal{o}' \rightarrow \text{prop}$ 
by form tt
; form ff
; form ( $\forall A \ B$ ) := form A  $\wedge$  form B
; form ( $\nabla A \ B$ ) := form A  $\wedge$  form B
; form ( $\supset A \ B$ ) := form A  $\wedge$  form B
; form ( $= X \ Y$ ) := form A
; form ( $\Box X \ A$ ) := form A
; form ( $\Diamond X \ A$ ) := form A
; form ( $\Box^\uparrow X \ A$ ) :=  $\forall w, \text{form } (A \ w)$ 
; form ( $\Diamond^\uparrow X \ A$ ) :=  $\forall w, \text{form } (A \ w)$ 
; form ( $\Box^\downarrow X \ A$ ) :=  $\forall w, \text{form } (A \ w)$ 
; form ( $\Diamond^\downarrow X \ A$ ) :=  $\forall w, \text{form } (A \ w).$ 
```

The predicate form is a trick used to guide the induction in Abella by the structure of formulae (instead of by the structure of sat , which is not stratified [MT04], due to the presence of implication). Thereby the proof of theorem `bisim_sat` below is established by induction on the structure of the modal formulae and by case analyses on the definition of the satisfiability relation sat .

⁵In the Abella proof scripts, \Box , \Box^\uparrow and \Box^\downarrow are represented using keywords `boxAct`, `boxOut` and `boxIn`, respectively.

Theorem soundness: $\forall P Q,$
 $\text{bisim } P Q \rightarrow \forall F, \text{form } F \rightarrow (\text{sat } P F \rightarrow \text{sat } Q F) \wedge (\text{sat } Q F \rightarrow \text{sat } P F).$

The soundness theorem is a simple corollary of `bisim_sat`, since `form F` is a tautology. Due to the adequacy of this embedding, this concludes the mechanisation of the soundness of open bisimilarity with respect to \mathcal{OM} (Theorem 3.2).

We have not yet discussed the λ Prolog embedding of the labelled transition system because specific details of the transition system do not affect the soundness property. During the proof of `bisim_sat`, the transition relations (`one` and `oneb`) are never inspected, nor are processes inspected. The only specific details of the labelled transition system, which are evident in the definitions of `bisim` and `sat`, are the transition labels `up` and `dn`. Thus, the minimal definitions that are required in the specification `finite-pic` of the soundness proof are:

```

kind n type. % names
kind p type. % processes
kind a type. % actions (transition labels)
type up, dn n → n → a.

type one p → a → o. % one step free transition
type oneb p → (n → a) → (n → p) → o. % one step bound transition
one P A Q :- ... .. % arbitrary transition definition suffice
oneb P A Q :- ... .. % arbitrary transition definition suffice

```

Additional details of the processes syntax and the transition relations are irrelevant to the soundness property. Recall that both the bisimilarity (`bisim`) and modal logic satisfaction (`sat`) is defined in terms of single step transition relations `one` and `oneb`. The soundness property could be understood in term of the question of whether the use of transition relations in `bisim` match well with the use of those relations in `sat`. Regardless of how we specify the process syntax and the transition relations, `bisim_sat` continues to hold in Abella without any modification to our proof script. This indicates that our soundness result (and also completeness, for similar reasons) is independent of the terms of the processes calculus, relying only on the form of the transition labels and monotonicity of the labelled transition system. Note, the style of embedding for the labelled transition system, which relies on Abella's logical quantifiers for implicitly embedding histories, already assumes monotonicity of labelled transitions (Proposition A.3) due to the intuitionistic nature of Abella.

6.3. Specification of the syntax and labelled transition systems of the π -calculus in λ Prolog. Despite the abstract nature of our soundness result, discussed above, in order to check specific examples of bisimilarity and satisfaction in Abella, an embedding of the syntax and labelled transition system of the π -calculus in Fig. 1 is required. Fig. 6 is a transcription of the syntactic constructs and transition rules into λ Prolog. This embedding of the π -calculus labelled transition system and bisimulation originates from the work of Tiu and Miller [TM10], where they provide rigorous adequacy result for their logical embedding, which has an obvious direct transcription into λ Prolog. By adequacy we mean that standard definitions and the logical embedding prove the same theorems.

Each syntactic category is declared as a type (e.g., `n`, `p`, and `a`) using the `kind` keyword. Syntactic constructs are defined as constants in λ Prolog where some of which may require multiple arguments to construct the desired syntactic category. For instance, `plus` needs two

```

1  sig finite-pic. % file: finite-pic.sig
2
3  kind n type. % names
4
5  kind p type. % processes
6  type null      p.          % deadlock
7  type taup     p → p.      % progress action
8  type plus, par p → p → p. % choice, par
9  type match, out n → n → p → p. % match, output action
10 type in       n → (n → p) → p. % input action
11 type nu      (n → p) → p.    % nu
12
13 kind a type. % actions (transition labels)
14 type tau     a.
15 type up, dn  n → n → a.
16
17 type one     p → a → p → o. % one step free transition
18 type oneb   p → (n → a) → (n → p) → o. % one step bound transition
19
20
21 module finite-pic. % file: finite-pic.mod
22
23 oneb (in X M) (dn X) M. % bound input
24 one  (out X Y P) (up X Y) P. % free output
25 one  (taup P) tau P. % tau
26 % match prefix
27 one  (match X X P) A Q :- one P A Q.    oneb (match X X P) A M :- oneb P A M.
28 % sum
29 one  (plus P Q) A R :- one P A R.    oneb (plus P Q) A M :- oneb P A M.
30 one  (plus P Q) A R :- one Q A R.    oneb (plus P Q) A M :- oneb Q A M.
31 % par
32 one  (par P Q) A (par P1 Q) :- one P A P1.    oneb (par P Q) A (z\par (M z) Q) :- oneb P A M.
33 one  (par P Q) A (par P Q1) :- one Q A Q1.    oneb (par P Q) A (z\par P (N z)) :- oneb Q A N.
34 % restriction
35 one  (nu z\p z) A (nu z\q z) :- pi z\ one (P z) A (Q z).
36 oneb (nu z\p z) A (y\nu z\q z y) :- pi z\ oneb (P z) A (y\q x z).
37 % open (bound output)
38 oneb (nu x\p x) (up X) Q :- pi y\ one (P y) (up X y) (Q y).
39 % close
40 one  (par P Q) tau (nu z\par (M z) (N z)) :- oneb P (dn X) M, oneb Q (up X) N.
41 one  (par P Q) tau (nu z\par (M z) (N z)) :- oneb P (up X) M, oneb Q (dn X) N.
42 % comm (interaction)
43 one  (par P Q) tau (par (M Y) T) :- oneb P (dn X) M, one Q (up X Y) T.
44 one  (par P Q) tau (par R (M Y)) :- oneb Q (dn X) M, one P (up X Y) R.

```

Figure 6: λ Prolog specification of the syntax and transitions of the π -calculus. (Adopted from one of the examples distributed with Abella. The adequacy for this λ Prolog embedding has been justified by Tiu and Miller [TM10].)

process arguments to construct a process, as its type ($\mathbf{p} \rightarrow \mathbf{p} \rightarrow \mathbf{p}$) suggests. The table below summarises the process syntax in λ Prolog and the notations used in the previous sections.

λ Prolog syntax	mathematical notation
<code>null</code>	0
<code>nu x \ P x</code> (or <code>nu P</code>)	$\nu z.P$ ($(P \ z)$ corresponds to P)
<code>taup P</code>	$\tau.P$
<code>out x z P</code>	$\bar{x}z.P$
<code>in x z \ P z</code> (or <code>in x P</code>)	$x(z).P$ ($(P \ z)$ corresponds to P)
<code>match x y P</code>	$[x = y]P$
<code>par P Q</code>	$P \mid Q$
<code>plus P Q</code>	$P + Q$

Fig. 6 has few stylistic differences from Fig. 1. Firstly, distinct constants are used for actions and their related processes (e.g., a `tau` action for `taup` prefixed process) because the constants cannot be overloaded as in mathematical notation. Secondly, the action prefix $\pi.P$ in Fig. 1, where π ranges over several different types of action (progress, free out, and input), is transcribed as three distinct process syntactic constructs (`taup`, `out`, and `in`). Thirdly, free and bound actions are distinguished by their types instead of using different notations ($\bar{x}z$ and $\bar{x}(z)$) as in Fig. 1. That is, bound actions (e.g., `up x : n → a`) are partially applied free actions (e.g., `up x z : a`) using the same constants. Fourthly, two different sets of transition relations are defined: `one` relating a process (\mathbf{p}) to another process (\mathbf{p}) via a free action (\mathbf{a}) and `oneb` relating a process (\mathbf{p}) with a bound process ($\mathbf{n} \rightarrow \mathbf{p}$) via a bound action ($\mathbf{n} \rightarrow \mathbf{a}$).

One advantage of using λ Prolog [MN12] is that we can rely on its native support for a variant of HOAS, known as *λ -tree syntax* [MP99], for handling bound variables and $\alpha\beta\eta$ -equivalence automatically. For instance, consider the rule for name extrusion from both Fig. 6 and Fig. 1:

$$\text{oneb } (\text{nu } z \backslash P \ z) \ (\text{up } X) \ Q \ :- \quad \frac{P \xrightarrow{\bar{x}z} Q}{\nu z.P \xrightarrow{\bar{x}(z)} Q} \quad x \neq z$$

There is no need to explicitly state and keep track of the side conditions such as $x \neq z$ and $x \notin n(\pi)$ in λ Prolog definitions. For example, consider `pi z \ one (P z) (up X z) (Q z)` from above. Here, it is guaranteed that z does not occur free in the logic variables P , X , and Q . This guarantee comes from the scoping of variables: the scopes of P , X , and Q go beyond the scope of z , which is limited only to `one (P z) (up X z) (Q z)`. Had z freely occurred in any of P , X , or Q , the scope of z would have been violated. Hence, X cannot be unified with z .

7. DISTINGUISHING FORMULAE GENERATION ALGORITHM IMPLEMENTATION

Our completeness proof in Section 4 is constructive in the sense that it follows the structure of an algorithm (Section 4.2) to build a pair of formulae for a pair of distinguishable processes. That is, one can find distinguishing formulae for any given pair of processes when they are actually distinguishable, guided by the steps described in our completeness proof (Section 4.3). These steps can be automated by writing a program that implements this algorithm. We first implemented the distinguishing formulae generation algorithm using

```

In [25]: 1 -- runBisimExperiment history leftProces rightProcess
2 runBisimExperiment h prL prR =
3   sequence . map (display . html) $
4     map (texParen . show)
5       [ text "h=" <> texHistory h
6         , text "\\mathsf{L}\\!:" <> pPrintTeX prL
7         , text "\\mathsf{R}\\!:" <> pPrintTeX prR
8         , text $ "\\mathsf{L}"
9           ++ (if bisimResult then "\\sim" else "\\not\\sim")
10          ++ "\\mathsf{R}"
11          ++ "\\quad()\\text{" ++ (if bisimResult then "" else "non-") ++ "bisimilar}"
12         ]
13 ++ [ texParen $ "\\phi_L=" ++ ppTeXString fl ++ "\\;, ~\\phi_R=" ++ ppTeXString fr
14      | (fl, fr) <- forest2df bisimForest ]
15 ++ [ texForest bisimForest ]
16 where
17   bisimResult = bisim h prL prR
18   bisimForest = bisim' h prL prR

```

```

In [26]: 1 -- runBisimExperiment history leftProces rightProcess
2 x = s2n "x"
3 y = s2n "y"
4 runBisimExperiment [All x, All y] (tau .+ tau p tau) (taup $ (x= y) tau)

h = y^j . x^i
L: τ0 + τ.τ0
R: τ[x = y]τ0
L ≈ R    non-bisimilar
φL = ⟨τ⟩τ⊥ , φR = [τ][x = y]⟨τ⟩T
φL = ⟨τ⟩⟨τ⟩T , φR = [τ][τ](x = y)
φL = [τ](τ⊥ ∨ ⟨τ⟩T) , φR = ⟨τ⟩([x = y]⟨τ⟩T ∧ [τ](x = y))

• L: h = yj . xi , σ = [] -τ→ 0
  • R: h = yj . xi , σ = [] -τ→ [x = y]τ0
    • R: h = yj . xi , σ = [(x, y)] -τ→ 0

• L: h = yj . xi , σ = [] -τ→ τ0
  • R: h = yj . xi , σ = [] -τ→ [x = y]τ0
    • L: h = yj . xi , σ = [] -τ→ 0
    • R: h = yj . xi , σ = [(x, y)] -τ→ 0
      • L: h = yj . xi , σ = [(x, y)] -τ→ 0

• R: h = yj . xi , σ = [] -τ→ [x = y]τ0
  • L: h = yj . xi , σ = [] -τ→ 0
    • R: h = yj . xi , σ = [(x, y)] -τ→ 0
  • L: h = yj . xi , σ = [] -τ→ τ0
    • L: h = yj . xi , σ = [] -τ→ 0
    • R: h = yj . xi , σ = [(x, y)] -τ→ 0
      • L: h = yj . xi , σ = [(x, y)] -τ→ 0

```

Figure 7: Screenshot of our distinguishing formulae generation algorithm running in Jupyter.

Haskell [AHT17b], accompanying the conference publication [AHT17a] of our work. Here, we provide pointers to our current implementation and describe continuing work to make it more accessible to those who are not accustomed to Haskell development tools including the GHC compiler.

Our current implementation is available online from a public GitHub repository⁶. This repository contains an example notebook and some utility shell scripts, which utilize a Docker container published on DockerHub to run the example notebook. All the necessary software dependencies, including Jupyter and Haskell, to run our implementation is contained in a Docker image so that it does not interfere with your own system. Anyone with access to an Internet-connected machine with a properly working Docker system can easily run our implementation with a single shell script command, although it would initially require some download time and disk space for a sizable (<10GB) Docker image to start running.

Figure 7 is a screenshot of a web browser connected to a Jupyter server running on the same machine (i.e., localhost). The Haskell source code documented in our technical report [AHT17b] is executed via IHaskell, a Haskell language kernel for the Jupyter notebook environment. Some additional features are the enhanced output exploiting Jupyter notebook’s ability to render HTML and also some LaTeX via MathJax. In addition to the plain text output, which you can still use inside the Jupyter notebook, we provide LaTeX output that looks the same as the notation used in this article.

In Figure 7, we define the function `runBisimExperiment` and demonstrate it used on a pair of non-bisimilar processes, discussed previously in Section 3.1.1. We are providing the implementation as a Haskell library so that one can build programs that automate tasks related to bisimulation. The function `runBisimExperiment` is defined in terms of more primitive definitions provided in our implementation. This function is applied to three arguments: the initial history, the left processes, and the right process. The free variables of the two processes must be closed by the initial history, either as input (using `A11`) as output (using `Nab`). In the example run, we closed the two free variables `x` and `y` as inputs in the initial history `[A11 x, A11 y]`. To provide processes, one should build them as Haskell values representing abstract syntax trees, whose data type is defined as follows:

```
import GHC.Generics (Generic)
import Unbound.Generics.LocallyNameless -- using unbound-generics library

type Nm = Name Tm
newtype Tm = Var Nm deriving (Eq, Ord, Show, Generic)

data Pr = Null | TauP Pr | Out Tm Tm Pr | In Tm PrB | Match Tm Tm Pr
        | Plus Pr Pr | Par Pr Pr | Nu PrB deriving (Eq, Ord, Show, Generic)
type PrB = Bind Nm Pr

instance Eq PrB where (==) = aeq; instance Ord PrB where compare = acompare
instance Alpha Tm; instance Alpha Pr; instance Subst Tm Pr
```

For instance, the processes $\tau.0 + \tau.\tau.0$ is written as `(Plus (TauP Null) (TauP(TauP Null)))` and the process $\tau.[x = y]\tau.0$ is written as `(TauP(Match (Var x) (Var y) (TauP Null)))` using the data type definition above. We additionally provide shorthand definitions below to reduce keyboard strokes and to make it look closer to the notations used in this article.

```
(.\) = bind; infixr 1 .\ ; (.)+ = Plus; infixl 6 .+ ; (.)| = Par; infixl 5 .|
```

⁶<https://github.com/kyagr/ihaskell-picalc>

```
o = Null ;   taup = TauP ;   out x y = Out(Var x)(Var y) ;   inp = In . Var ;   nu = Nu
x .= y = Match (Var x) (Var y) ;   tau = TauP Null ;   tautau = TauP (TauP Null)
```

Using these shorthand definitions, we can write $\tau.0+\tau.\tau.0$ and $\tau.[x=y]\tau.0$ as $(\text{tau} \ .+ \ \text{taup} \ \text{tau})$ and $(\text{taup} \ \$ \ (x.=y) \ \text{tau})$.

As a result of running the function `runBisimExperiment`, it displays the following:

- the initial history and the two processes,
- the result of bisimilarity test,
- a pair of distinguishing formulae (for non-bisimilar processes), and
- a set of trees consisting of all transitions in a distinguishing strategy.

The first two items are self explanatory from its rendered output. Let us explain few additional details on the last two items above.

Our implementation handles a subset of \mathcal{OM} -formulae (Defintion 2.3) that is sufficient to generate distinguishing formulae for any pair of non-bisimilar processes. The distinguishing formulae constructed during our completeness proof contain a limited form of implication $(x = y) \supset \phi$, where the left side of the implication connective is always an equality. For more compact output, our implementation uses a notation that abbreviates this form of implication. For instance, $[x = y]\phi$ abbreviates $(x = y) \supset \phi$. More generally, $[x_1 = y_1, \dots, x_n = y_n]\phi$ abbreviates $(x_1 = y_1) \supset \dots \supset (x_n = y_n) \supset \phi$. This abbreviation is also used in the screenshot of Figure 7.

The tool can also displaying multiple pairs of distinguishing formulae and displaying the entire bisimulation tree, which can be useful since the first distinguishing formula generated is not necessarily the most insightful. A single pair of distinguishing formulae is enough to witness distinguishability (or, non-bisimilarity) and the entire bisimulation tree is rarely required to generate a pair of distinguishing formulae. Our Haskell source code conceptually computes over the structure of the entire bisimulation tree in order to generate distinguishing formulae. However, only part of the tree that is needed for the formulae construction would actually be computed, thanks to Haskell's lazy evaluation, unless the entire tree is needed elsewhere. Further details of our algorithm implementation can be found in the technical report on our initial implementation [AHT17b].

8. CONCLUSION

The main result of this paper is a sound and complete logical characterisation of open bisimilarity for the π -calculus. To achieve this result, we introduce modal logic \mathcal{OM} , defined in Fig. 2. The soundness of \mathcal{OM} with respect to open bisimilarity, Theorem 3.2, is mechanically proven in Abella as explained in Section 6. The details of the completeness, Theorem 3.3, are provided in Section 4.

Intuitionistic modal logic \mathcal{OM} satisfies the following established criteria for an intuitionistic modal logic [Sim94]:

- Intuitionistic \mathcal{OM} is a conservative extension of intuitionistic logic. Removing modalities, we obtain a standard semantics of intuitionistic modal logic without any new theorems.
- Intuitionistic \mathcal{OM} satisfies intuitionistic hereditary. Every operator is closed under an accessible world relation, as given by our Kripke semantics in the Appendix, and also as captured by the notion of respectful substitution in the body of the paper.

- The law of excluded middle is invalidated. As demonstrated in Examples 3.1.1 and 4.4.4, the absence of the law of excluded middle is essential for the existence of distinguishing formulae in \mathcal{OM} for processes that are not open bisimilar but are late equivalent.
- As explored in Corollary 5.4, if we induce the law of excluded middle, we obtain a classical modal logic (characterising late equivalence).
- In contrast to classical modal logics, diamond and box modalities have independent interpretations, not de Morgan dual to each other.

A more direct proof theory for \mathcal{OM} is left as an open problem. A proof system can be used to confirm criteria such as: if $\phi \vee \psi$ has a proof, then either ϕ has a proof or ψ has a proof. A sound and complete proof system would be a step towards addressing the following, more philosophical, criterion for an intuitionistic modal logic [Sim94]:

There is an intuitionistically comprehensible explanation of the meaning of the modalities, relative to which IML is sound and complete.

Previous work on intuitionistic modal logic for program analysis [PS86, SI94] was motivated by topological interpretations of liveness properties. The intuitionistic information partial ordering in that work is quite different from in \mathcal{OM} , where the intuitionistic information partial ordering is given by the instantiation of inputs. We expect creative use of intuitionistic information partial orderings will lead to further useful intuitionistic modal logics.

The main novelty of this paper is the completeness proof, Proposition 4.9, involving an algorithm constructing distinguishing formulae for processes that are not open bisimilar. To use this algorithm, firstly attempt to prove that two processes are open bisimilar. If they are not open bisimilar, after a finite number of steps, a distinguishing strategy, according to Def. 4.1, will be discovered. The strategy can then be used to inductively construct two distinguishing formulae, one biased to each process. A key feature of the construction is the use of preconditions and diamond for the leading process, e.g., $x = y \supset \langle \pi \rangle \mathbf{tt}$, and box and postconditions for the following process, e.g., $[\pi](x = y)$. Interesting examples involving postconditions are provided in Section 4.4.

The logic \mathcal{OM} is suitable for formal and automated reasoning. It has natural encodings in Abella for mechanised reasoning, used to establish Theorem 3.2. In addition, our distinguishing formulae generation algorithm is implemented in Haskell, as explained in Section 7 and a companion report [AHT17b]. We envision that \mathcal{OM} and related intuitionistic modal logics characterising bisimilarity congruences have a role in symbolic model checking.

ACKNOWLEDGMENTS

We are grateful to Sam Staton for providing an example that helped us discover the completeness proof. We also appreciate the comments of the anonymous reviewers. The authors receive support from MOE Tier 2 grant MOE2014-T2-2-076. The first author receives support from NRF grant 2018R1C1B5046826 funded by Korea government (MSIT). The second author receives support from Singapore NRF grant NRF2014NCR-NCR001-30.

REFERENCES

- [ABF18] Martín Abadi, Bruno Blanchet, and Cédric Fournet. The applied pi calculus: Mobile values, new names, and secure communication. *J. ACM*, 65(1):1:1–1:41, 2018. doi:10.1145/3127586.
- [AHT17a] Ki Yung Ahn, Ross Horne, and Alwen Tiu. A Characterisation of Open Bisimilarity using an Intuitionistic Modal Logic. In Roland Meyer and Uwe Nestmann, editors, *28th International Conference on Concurrency Theory (CONCUR 2017)*, volume 85 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:17, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CONCUR.2017.7.
- [AHT17b] Ki Yung Ahn, Ross Horne, and Alwen Tiu. Generating witness of non-bisimilarity for the pi-calculus. *CoRR*, abs/1705.10908, 2017. URL <http://arxiv.org/abs/1705.10908>.
- [BCG⁺14] David Baelde, Kaustuv Chaudhuri, Andrew Gacek, Dale Miller, Gopalan Nadathur, Alwen Tiu, and Yuting Wang. Abella: A system for reasoning about relational specifications. *Journal of Formalized Reasoning*, 7(2):1–89, 2014. doi:10.6092/issn.1972-5787/4650.
- [BGM⁺07] David Baelde, Andrew Gacek, Dale Miller, Gopalan Nadathur, and Alwen Tiu. *The Bedwyr System for Model Checking over Syntactic Expressions*, pages 391–397. 2007. doi:10.1007/978-3-540-73595-3_28.
- [BJPV11] Jesper Bengtson, Magnus Johansson, Joachim Parrow, and Björn Victor. Psi-calculi: a framework for mobile processes with nominal data and logic. *Logical Methods in Computer Science*, 7(1), 2011. doi:10.2168/LMCS-7(1:11)2011.
- [BN06] Sébastien Briaïs and Uwe Nestmann. Open bisimulation, revisited. *Electr. Notes Theor. Comput. Sci.*, 154(3):109–123, 2006. doi:10.1016/j.tcs.2007.07.010.
- [GMN12] Andrew Gacek, Dale Miller, and Gopalan Nadathur. A two-level logic approach to reasoning about computations. *Journal of Automated Reasoning*, 49(2):241–273, 2012. doi:10.1007/s10817-011-9218-1.
- [HALT18] Ross Horne, Ki Yung Ahn, Shang-Wei Lin, and Alwen Tiu. Quasi-open bisimilarity with mismatch is intuitionistic. In *Proceedings of LICS '18: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, Oxford, United Kingdom, July 9-12, 2018 (LICS '18)*, page 10, 2018. doi:10.1145/3209108.3209125.
- [HM85] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985. doi:10.1145/2455.2460.
- [HM21] Ross Horne and Sjouke Mauw. Discovering epassport vulnerabilities using bisimilarity. *Logical Methods in Computer Science*, 17(2):24:1–24:52, 2021. doi:10.23638/LMCS-17(2:24)2021.
- [Hor18] Ross Horne. A bisimilarity congruence for the applied pi-calculus sufficiently coarse to verify privacy properties. *arXiv*, (arXiv:1811.02536), 2018. URL <https://arxiv.org/abs/1811.02536>.
- [JVP12] Magnus Johansson, Björn Victor, and Joachim Parrow. Computing strong and weak bisimulations for psi-calculi. *The Journal of Logic and Algebraic Programming*, 81(3):162–180, 2012. doi:10.1016/j.jlap.2012.01.001.
- [Koz83] Dexter Kozen. Results on the propositional mu-calculus. *Theoretical Comput. Sci.*, 27(3):333–354, 1983. doi:10.1016/0304-3975(82)90125-6.
- [Kri65] Saul A Kripke. Semantical analysis of intuitionistic logic I. In *Studies in Logic and the Foundations of Mathematics*, volume 40, pages 92–130. Elsevier, 1965. doi:10.1016/S0049-237X(08)71685-9.
- [MN12] Dale Miller and Gopalan Nadathur. *Programming with Higher-Order Logic*. Cambridge University Press, 2012. doi:10.1017/CBO9781139021326.
- [MP99] Dale Miller and Catuscia Palamidessi. Foundational aspects of syntax. *ACM Computing Surveys*, 31(3), 1999. doi:10.1145/333580.333590.
- [MPW92] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, Parts I and II. *Information and Computation*, 100(1):1–77, 1992. doi:10.1016/0890-5401(92)90008-4.
- [MPW93] Robin Milner, Joachim Parrow, and David Walker. Modal logics for mobile processes. *Theoretical Computer Science*, 114(1):149–171, 1993. doi:10.1016/0304-3975(93)90156-N.
- [MS92] Robin Milner and Davide Sangiorgi. Barbed bisimulation. In W. Kuich, editor, *Automata, Languages and Programming*, pages 685–695, 1992. doi:10.1016/S0049-237X(08)71685-9.
- [MT04] Alberto Momigliano and Alwen Tiu. Induction and co-induction in sequent calculus. In Stefano Berardi, Mario Coppo, and Ferruccio Damiani, editors, *Types for Proofs and Programs*, pages 293–308, Berlin, Heidelberg, 2004.

- [NM88] Gopalan Nadathur and Dale Miller. An Overview of λ Prolog. In *Appears in the Fifth International Conference Symposium on Logic Programming, 15-19 August 1988, Seattle, Washington*, pages 810–827. MIT Press, 1988. URL <http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/iclp88.pdf>.
- [PBE⁺15] Joachim Parrow, Johannes Borgström, Lars-Henrik Eriksson, Ramunas Gutkovas, and Tjark Weber. Modal logics for nominal transition systems. In *CONCUR 2015*, volume 42 of *LIPICs*, pages 198–211, 2015. doi:10.4230/LIPICs.CONCUR.2015.198.
- [PBE⁺20] Joachim Parrow, Johannes Borgström, Lars-Henrik Eriksson, Ramunas Gutkovas, and Tjark Weber. Modal logics for nominal transition systems. *arXiv*, 2020. URL <https://arxiv.org/abs/1904.02564>.
- [PS86] Gordon Plotkin and Colin Stirling. A framework for intuitionistic modal logics. In *Proceedings of the 1986 Conference on Theoretical aspects of reasoning about knowledge*, pages 399–406, 1986. doi:10.1017/S0022481200028826.
- [San95] Davide Sangiorgi. On the proof method for bisimulation. In Jiří Wiedermann and Petr Hájek, editors, *Mathematical Foundations of Computer Science*, pages 479–488, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg. doi:10.1007/3-540-60246-1_153.
- [San96] Davide Sangiorgi. A theory of bisimulation for the π -calculus. *Acta Informatica*, 33(1):69–97, 1996. doi:10.1007/s002360050036.
- [SI94] Bernhard Steffen and Anna Ingólfssdóttir. Characteristic formulas for processes with divergence. *Information and Computation*, 110(1):149 – 163, 1994. doi:10.1006/inco.1994.1028.
- [Sim94] Alex K. Simpson. *The proof theory and semantics of intuitionistic modal logic*. PhD thesis, University of Edinburgh, UK, 1994.
- [SW01] Davide Sangiorgi and David Walker. On barbed equivalences in pi-calculus. In Kim Guldstrand Larsen and Mogens Nielsen, editors, *CONCUR 2001 - Concurrency Theory, 12th International Conference, Aalborg, Denmark, August 20-25, 2001, Proceedings*, volume 2154 of *Lecture Notes in Computer Science*, pages 292–304. Springer, 2001. doi:10.1007/3-540-44685-0_20.
- [TD10] Alwen Tiu and Jeremy Dawson. Automating open bisimulation checking for the spi calculus. In *Computer Security Foundations Symposium (CSF), 2010 23rd IEEE*, pages 307–321. IEEE, 2010. doi:10.1109/CSF.2010.28.
- [TM10] Alwen Tiu and Dale Miller. Proof search specifications of bisimulation and modal logics for the π -calculus. *ACM Transactions on Computational Logic*, 11(2):13:1–13:35, 2010. doi:10.1145/1656242.1656248.
- [TNH16] Alwen Tiu, Nam Nguyen, and Ross Horne. SPEC: An equivalence checker for security protocols. In Atsushi Igarashi, editor, *Programming Languages and Systems*, pages 87–95, Cham, 2016. Springer. doi:10.1007/978-3-319-47958-3_5.
- [VM94] Björn Victor and Faron Moller. The mobility workbench - A tool for the pi-calculus. In David L. Dill, editor, *Computer Aided Verification, 6th International Conference, CAV '94, Stanford, California, USA, June 21-23, 1994, Proceedings*, volume 818 of *Lecture Notes in Computer Science*, pages 428–440. Springer, 1994. doi:10.1007/3-540-58179-0_73.

APPENDIX A. A KRIPKE SEMANTICS FOR \mathcal{OM}

To provide another explanation for why \mathcal{OM} is intuitionistic, we provide here a reformulation of the semantics of \mathcal{OM} in Figure 2 in terms of an intuitionistic Kripke semantics.

In the following, we denote with \mathcal{E} (possibly with subscripts) a finite binary relation between variables. We write $\text{fn}(\mathcal{E})$ to denote the set of variables occurring in \mathcal{E} , and we write \mathcal{E}^* to denote the reflexive-symmetric-transitive closure of \mathcal{E} . That is, \mathcal{E}^* is an equivalence relation on variables. Given a substitution σ and a relation \mathcal{E} , we write $\sigma \Vdash \mathcal{E}$ iff $x\sigma = y\sigma$ for every $(x, y) \in \mathcal{E}$.

A *world* is a triple (P, h, \mathcal{E}) of process P , history h and a binary relation \mathcal{E} on variables such that

- $\text{fn}(P) \subseteq \text{fn}(h)$, and $\text{fn}(\mathcal{E}) = \text{fn}(h)$;
- there exists a substitution σ respecting h such that $\sigma \Vdash \mathcal{E}$.

Let us denote with W the set of worlds. Define a relation \preceq on worlds as follows:

$$(P_1, h_1, \mathcal{E}_1) \preceq (P_2, h_2, \mathcal{E}_2) \quad \text{iff} \quad P_1 = P_2, \quad h_1 = h_2, \quad \text{and} \quad \mathcal{E}_1^* \subseteq \mathcal{E}_2^*.$$

It is easy to see that \preceq is a partial order on W (reflexive, anti-symmetric and transitive), so the pair (W, \preceq) forms an intuitionistic Kripke frame. A Kripke model for \mathcal{OM} is then a triple (W, \preceq, w) where $w \in W$. In the following, we fix the Kripke frame to (W, \preceq) so it will be implicitly assumed in the definition of the satisfiability relation. Given a world (h, \mathcal{E}) and a substitution σ , we say that σ respects (h, \mathcal{E}) iff σ respects h and $\sigma \Vdash \mathcal{E}$.

We write $P \models^{h, \mathcal{E}} \phi$ when ϕ is true in the world (P, h, \mathcal{E}) . The complete definition of this satisfaction relation is given in Figure 8. In the figure, the relation $[P \xrightarrow{\pi} Q]^{h, \mathcal{E}}$ is defined as

$$[P \xrightarrow{\pi} Q]^{h, \mathcal{E}} \quad \text{iff} \quad \forall \sigma. \text{ s.t. } \sigma \text{ respects } (h, \mathcal{E}) \text{ and } \text{bn}(\pi) \notin \text{fn}(\sigma), \text{ we have } P\sigma \xrightarrow{\pi\sigma} Q\sigma.$$

Lemma A.1. *If $[P_1 \xrightarrow{\pi} Q]^{h_1, \mathcal{E}_1}$ and $(P_1, h_1, \mathcal{E}_1) \preceq (P_2, h_2, \mathcal{E}_2)$ then $[P_2 \xrightarrow{\pi} Q]^{h_2, \mathcal{E}_2}$.*

Proof. By the definition of \preceq , we have $P_1 = P_2$, $h_1 = h_2$ and $\mathcal{E}_1 \subseteq \mathcal{E}_2$. The lemma then follows from the fact that if σ respects (h, \mathcal{E}_2) then it also respects (h, \mathcal{E}_1) . \square

Lemma A.2. *For every world (P, h, \mathcal{E}) , σ , π and Q such that σ respects (h, \mathcal{E}) , $\text{dom}(\sigma) \subseteq \text{fn}(h)$, and $P\sigma \xrightarrow{\pi} Q$, where $\text{bn}(\pi) \notin \text{fn}(h) \cup \text{fn}(\sigma)$, there exists Q' and π' such that $\pi = \pi'\sigma$ and $Q = Q'\sigma$ and $[P \xrightarrow{\pi'} Q']^{h, \mathcal{E}}$.*

Proof. By structural induction on the derivation of $P\sigma \xrightarrow{\pi} Q$. We show here an interesting case where match is involved, e.g., when $P = [x = y]R$, and $x\sigma = y\sigma = u$, so $P\sigma = [u = u]R\sigma$.

$$\frac{R\sigma \xrightarrow{\pi} Q}{[u = u]R\sigma \xrightarrow{\pi} Q}$$

By the induction hypothesis, we have Q' and π' such that $[R \xrightarrow{\pi'} Q']^{h, \mathcal{E}}$. To show $[x = y]R \xrightarrow{\pi'} Q']^{h, \mathcal{E}}$ we need to show that for every θ that respects (h, \mathcal{E}) , we have $[x\theta = y\theta]R\theta \xrightarrow{\pi'\theta} Q'\theta$. This is constructed as follows:

$$\frac{R\theta \xrightarrow{\pi'\theta} Q'\theta}{[x\theta = y\theta]R\theta \xrightarrow{\pi'\theta} Q'\theta}$$

The premise follows from $[R \xrightarrow{\pi'} Q']^{h, \mathcal{E}}$. It remains to show that this application of the match rule is valid, i.e., that $x\theta = y\theta$. Since $x\sigma = y\sigma$, and since (P, h, \mathcal{E}) is a world (which means $\text{fn}(P) \subseteq \text{fn}(h) = \text{fn}(\mathcal{E})$), and σ respects (h, \mathcal{E}) , it follows that $(x, y) \in \mathcal{E}$. Therefore any θ respecting (h, \mathcal{E}) would satisfy $x\theta = y\theta$. \square

$$\begin{array}{ll}
P \models^{h,\mathcal{E}} \mathbf{tt} & \text{always holds.} \\
P \models^{h,\mathcal{E}} x = y & \text{iff } (x, y) \in \mathcal{E} \\
P \models^{h,\mathcal{E}} \phi_1 \wedge \phi_2 & \text{iff } P \models^{h,\mathcal{E}} \phi_1 \text{ and } P \models^{h,\mathcal{E}} \phi_2. \\
P \models^{h,\mathcal{E}} \phi_1 \vee \phi_2 & \text{iff } P \models^{h,\mathcal{E}} \phi_1 \text{ or } P \models^{h,\mathcal{E}} \phi_2. \\
P \models^{h,\mathcal{E}} \phi_1 \supset \phi_2 & \text{iff } \forall (P', h', \mathcal{E}') \geq (P, h, \mathcal{E}). P \models^{h',\mathcal{E}'} \phi_1 \text{ implies } P \models^{h',\mathcal{E}'} \phi_2. \\
P \models^{h,\mathcal{E}} \langle \alpha \rangle \phi & \text{iff } \exists Q, [P \xrightarrow{\alpha} Q]^{h,\mathcal{E}} \text{ and } Q \models^{h,\mathcal{E}} \phi. \\
P \models^{h,\mathcal{E}} \langle \bar{a}(z) \rangle \phi & \text{iff } \exists Q, [P \xrightarrow{\bar{a}(z)} Q]^{h,\mathcal{E}} \text{ and } Q \models^{h \cdot z^o, \mathcal{E}_z} \phi. \\
P \models^{h,\mathcal{E}} \langle a(z) \rangle \phi & \text{iff } \exists Q, [P \xrightarrow{a(z)} Q]^{h,\mathcal{E}} \text{ and } Q \models^{h \cdot z^i, \mathcal{E}_z} \phi. \\
P \models^{h,\mathcal{E}} [\alpha] \phi & \text{iff } \forall (P', h', \mathcal{E}') \geq (P, h, \mathcal{E}), \forall Q, [P' \xrightarrow{\alpha} Q]^{h',\mathcal{E}'} \text{ implies } Q \models^{h',\mathcal{E}'} \phi. \\
P \models^{h,\mathcal{E}} [\bar{a}(z)] \phi & \text{iff } \forall (P', h', \mathcal{E}') \geq (P, h, \mathcal{E}), \forall Q, [P' \xrightarrow{\bar{a}(z)} Q]^{h',\mathcal{E}'} \text{ implies } Q \models^{h' \cdot z^o, \mathcal{E}'_z} \phi. \\
P \models^{h,\mathcal{E}} [a(z)] \phi & \text{iff } \forall (P', h', \mathcal{E}') \geq (P, h, \mathcal{E}), \forall Q, [P' \xrightarrow{a(z)} Q]^{h',\mathcal{E}'} \text{ implies } Q \models^{h' \cdot z^i, \mathcal{E}'_z} \phi.
\end{array}$$

Figure 8: An alternative semantics of \mathcal{OM} . In the figure, \mathcal{E}_z denotes $\mathcal{E} \cup \{(z, z)\}$.

The following results, easy to establish by induction, show that we can translate between the Kripke semantics in Figure 8 and the semantics in Figure 2.

Proposition A.3 (Monotonicity). *If $(P_1, h_1, \mathcal{E}_1) \preceq (P_2, h_2, \mathcal{E}_2)$ and $P_1 \models^{h_1, \mathcal{E}_1} \phi$ then $P_2 \models^{h_2, \mathcal{E}_2} \phi$.*

Proof. By induction on the definition of $P_1 \models^{h_1, \mathcal{E}_1} \phi$ and Lemma A.1. □

Proposition A.4. *$P \models^{h, \mathcal{E}} \phi$ if and only if $P\sigma \models^{h\sigma} \phi\sigma$ for every σ that respects (h, \mathcal{E}) .*

Proof. This follows from the definition of \models^h and $\models^{h, \mathcal{E}}$, and Lemma A.2. □

As a consequence of the result above, we obtain the following precise relationship between our two equivalence presentations of the semantics of \mathcal{OM} .

Corollary A.5. *$P \models^{h, id} \phi$ if and only if $P \models^h \phi$, where id is the identity relation on $\text{fn}(h)$.*

Proof. This is a consequence of Proposition A.4 and Lemma 2.5. □