A CHARACTERISATION OF OPEN BISIMILARITY USING AN INTUITIONISTIC MODAL LOGIC

KI YUNG AHN, ROSS HORNE, AND ALWEN TIU

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

School of Computer Science and Engineering, Nanyang Technological University, Singapore
e-mail address: rhorne@ntu.edu.sg

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

Abstract. Open bisimilarity is the original notion of bisimilarity to be introduced for the π-calculus that is a congruence. In open bisimilarity, free names in processes are treated as variables that may be instantiated lazily; in contrast to early and late bisimilarity where free names are constants. We build on the established line of work, due to Milner, Parrow, and Walker, on classical modal logics characterising early and late bisimilarity for the π-calculus. The important insight is, 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 (respectful) substitutions, inducing a property known as *intuitionistic hereditary*. Intuitionistic hereditary reflects the lazy instantiation of names in open bisimilarity. The soundness proof for open bisimilarity with respect to the modal logic is mechanised in Abella. The constructive content of the completeness proof provides an algorithm for generating distinguishing formulae, where such formulae are useful as a certificate explaining why two processes are not open bisimilar. We draw attention to the fact that open bisimilarity is not the only notion of bisimilarity that is a congruence: for name-passing calculi there is a classical/intuitionistic spectrum of bisimilarities.

1. Introduction

This work provides insight into the logical nature of open bisimilarity [San96], but firstly we recall why open bisimilarity itself is important. The problem open bisimilarity addressed was that the original notions of bisimilarity proposed for the π-calculus (early and late bisimilarity [MPW92, MPW93]) do not directly define congruence relations. That is, if we show two processes are bisimilar, it is not necessarily the case that they are bisimilar in any context. Having an equivalence relation that is not a congruence is problematic for compositional reasoning, since, having established an algebraic property, it cannot, be applied with confidence, anywhere inside a process. By addressing this problem, by providing a notion of bisimilarity that is a congruence, open bisimilarity allows the π-calculus to 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. Open bisimilarity is the notion of bisimilarity implemented in the Mobility

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

LOGICAL METHODS
IN COMPUTER SCIENCE

DOI:10.2168/LMCS-???

Workbench [VM94] — the first toolkit for the π-calculus. A reason open bisimilarity is efficient to implement is that it takes a lazy approach to instantiating names. When we consider an input action, we are not required to explore all possible inputs, but instead can leave the input name in the action as a variable, symbolically representing all possible inputs. This can avoid unnecessarily exploring 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 extensions of the π-calculus, where infinitely many messages may be received for a single input action [TD10]. Thus open bisimilarity has impact beyond the setting of the π-calculus.

The trick for ensuring 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 $\langle\tau\rangle\mathtt{tt} \vee \neg\langle\tau\rangle\mathtt{tt}$ 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 $P \models \neg\langle\tau\rangle\mathtt{tt}$ now reads, **under any substitution** σ, process $P\sigma$ cannot perform a τ-transition. Under this interpretation we have the following.

$$\overline{a}b \parallel c(x) \not\models \neg\langle\tau\rangle\mathtt{tt}$$

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

$$\overline{a}b \parallel c(x) \not\models \langle\tau\rangle\mathtt{tt}$$

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.

$$\overline{a}b \parallel c(x) \not\models \langle\tau\rangle\mathtt{tt} \vee \neg\langle\tau\rangle\mathtt{tt}$$

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 criteria for any intuitionistic logic.

Intuitively, the absence of the law of excluded middle for the example above can be interpreted as follows. For $\overline{a}b \parallel c(x)$, 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 the two. A distinguishing formula is a formulae 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 in a characteristic logic.

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

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

The above processes are not open bisimilar. Process R satisfies $[\tau](\langle\tau\rangle\mathtt{tt} \vee \neg\langle\tau\rangle\mathtt{tt})$, where the box modality ranges over all τ-transitions. However, process S does not satisfy $[\tau](\langle\tau\rangle\mathtt{tt} \vee \neg\langle\tau\rangle\mathtt{tt})$, since there is a τ-transition to process $\overline{a}b \parallel c(x)$ that we just agreed does not satisfy $\langle\tau\rangle\mathtt{tt} \vee \neg\langle\tau\rangle\mathtt{tt}$. In this example, the absence of the law of excluded middle is necessary for the existence of a formula distinguishing these processes 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 *classical/intuitionistic spectrum*.

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 principals, 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 describes how the proof assistant Abella [BCG+14] was used to mechanically prove soundness of $\mathcal{OM}$ with respect to open bisimulation. Section 6 situates $\mathcal{OM}$ with respect to other modal logics in the classical/intuitionistic spectrum. The soundness theorem (Section 5) and selected examples (Section 2 and Section 4) have been mechanised in the Abella theorem prover, and are available online.[1]

2. Introducing the intuitionistic modal logic $\mathcal{OM}$

We recall the syntax and labelled transition semantics for the finite π-calculus (Fig. 1). All features are standard: the deadlocked process that can do nothing, the ν quantifier that binds private names, the output prefix that outputs a name on a channel, the input prefix that binds the name received on a channel, the silent progress action τ, the name match guard, parallel composition, and non-deterministic choice. There are four types of action ranged over by π, where a *free output* sends a free name, whereas a *bound output* extrudes a ν-bounded private name. We employ the late labelled transition system for the π-calculus, where the name on the input channel is a symbolic place holder for a name that is not chosen until after an input transition. We use the notations $\mathrm{bn}(E)$ and $\mathrm{fn}(E)$ to represent the bound names and, respectively, free names in a given expression (processes, actions, formulae, etc) E.

Histories are used in the definitions of both the intuitionistic modal logic and open bisimilarity. Histories are lists (separated using a dot) representing what is known about free variables due to how

[1] Via https://github.com/alwentiu/abella/tree/master/pic

$$\begin{array}{rcll}
\pi & ::= & \tau & \text{(progress)} \\
 & & \overline{x}z & \text{(free out)} \\
 & & \overline{x}(z) & \text{(bound out)} \\
 & & x(z) & \text{(input)} \\
 & & & \\
P & ::= & 0 & \text{(deadlock)} \\
 & & \nu x.P & \text{(nu)} \\
 & & \pi.P & \text{(action)} \\
 & & [x = y]P & \text{(match)} \\
 & & P \parallel P & \text{(par)} \\
 & & P + P & \text{(choice)}
\end{array}$$

$$\frac{}{\pi.P \xrightarrow{\pi} P} \qquad \frac{P \xrightarrow{\pi} Q}{\nu x.P \xrightarrow{\pi} \nu x.Q}\ x \notin \mathrm{n}(\pi)$$

$$\frac{P \xrightarrow{\overline{x}z} Q}{\nu z.P \xrightarrow{\overline{x}(z)} Q}\ x \neq z \qquad \frac{P \xrightarrow{\pi} Q}{P \parallel R \xrightarrow{\pi} Q \parallel R}\ \begin{array}{c}\text{if } x \in \mathrm{bn}(\pi) \\ \text{then } x \text{ fresh for } R\end{array}$$

$$\frac{P \xrightarrow{\pi} R}{P + Q \xrightarrow{\pi} R} \qquad \frac{P \xrightarrow{\overline{x}(z)} P' \quad Q \xrightarrow{x(z)} Q'}{P \parallel Q \xrightarrow{\tau} \nu z.(P' \parallel Q')}$$

$$\frac{P \xrightarrow{\pi} R}{[x = x]P \xrightarrow{\pi} R} \qquad \frac{P \xrightarrow{\overline{x}y} P' \quad Q \xrightarrow{x(z)} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'\{^{y}\!/\!_{z}\}}$$

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

they have been communicated previously to the environment. There are two types of information about names recorded in a history: fresh private names that have been output, using action $\overline{a}(x)$, denoted by x^o; and variables (symbolically) representing inputs, using action $a(z)$, denoted by z^i. What matters is the alternation between the bound outputs and symbolic inputs, since an input variable can only be instantiated with private names that were output earlier 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 $y \in \mathrm{fn}(h')$, we have $y\sigma \neq x$.*

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.

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.

2.1. **The semantics of the intuitionistic modal logic $\mathcal{OM}$.** The syntax for modal logic $\mathcal{OM}$ is defined by the following grammar, extending intuitionistic logic with equality and modalities.

$$\left.\begin{array}{rll}
\phi ::= & \mathtt{tt} & \text{top} \\
\mid & \mathtt{ff} & \text{bottom} \\
\mid & \phi \wedge \phi & \text{and} \\
\mid & \phi \vee \phi & \text{or} \\
\mid & \phi \supset \phi & \text{implies} \\
\mid & x = y & \text{equality}
\end{array}\right\}\ \text{intuitionistic logic}$$

$$\left.\begin{array}{rll}
\mid & \langle \pi \rangle \phi & \text{diamond} \\
\mid & [\pi]\phi & \text{box}
\end{array}\right\}\ \text{modalities}$$

$$
\begin{array}{lll}
P \models^h \mathtt{tt} & \text{and} & P \models^h x = x \quad \text{always hold.} \\
P \models^h \phi_1 \wedge \phi_2 & \text{iff} & P \models^h \phi_1 \text{ and } P \models^h \phi_2. \\
P \models^h \phi_1 \vee \phi_1 & \text{iff} & P \models^h \phi_1 \text{ or } P \models^h \phi_2. \\
P \models^h \phi_1 \supset \phi_1 & \text{iff} & \forall \sigma \text{ respecting } h, P\sigma \models^{h\sigma} \phi_1\sigma \text{ implies } P\sigma \models^{h\sigma} \phi_2\sigma. \\
P \models^h \langle \alpha \rangle \phi & \text{iff} & \exists Q,\ P \xrightarrow{\alpha} Q \text{ and } Q \models^h \phi. \\
P \models^h \langle \overline{a}(z) \rangle \phi & \text{iff} & \exists Q,\ P \xrightarrow{\overline{a}(z)} Q \text{ and } Q \models^{h \cdot z^o} \phi. \\
P \models^h \langle a(z) \rangle \phi & \text{iff} & \exists Q,\ P \xrightarrow{a(z)} Q \text{ and } Q \models^{h \cdot z^i} \phi. \\
P \models^h [\alpha]\phi & \text{iff} & \forall \sigma \text{ respecting } h, \forall Q,\ P\sigma \xrightarrow{\alpha\sigma} Q \text{ implies } Q \models^{h\sigma} \phi\sigma. \\
P \models^h [\overline{a}(z)]\phi & \text{iff} & \forall \sigma \text{ respecting } h,\ \forall Q, P\sigma \xrightarrow{\overline{a\sigma}(z)} Q \text{ implies } Q \models^{h\sigma \cdot z^o} \phi\sigma. \\
P \models^h [a(z)]\phi & \text{iff} & \forall \sigma \text{ respecting } h, \forall Q, P\sigma \xrightarrow{a\sigma(z)} Q \text{ implies } Q \models^{h\sigma \cdot z^i} \phi\sigma.
\end{array}
$$

Figure 2: Semantics of the modal logic $\mathcal{OM}$, where α is τ or $\overline{a}b$; and z is fresh for P, h, and σ.

The semantics of intuitionistic modal logic $\mathcal{OM}$, presented in Fig. 2, is defined in terms of the late labelled transitions 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 2.2 (satisfaction)**.** *Satisfaction, written $P \models \phi$, holds whenever $P \models^{x_0^i \cdot \ldots \cdot x_n^i} \phi$ according to the inductive definition in Fig. 2, where* $\text{fn}(P) \cup \text{fn}(\phi) \subseteq \{x_0, \ldots, x_n\}$.

Observe the semantics for connectives of intuitionistic logic can be obtained from a suitable Kripke semantics. Take *worlds* to be pairs of processes and histories. Now define a relation over worlds $\leq$ as follows.

$P, h \ \leq_\sigma Q, h'$ whenever σ respects h, $P\sigma = Q$ and $h\sigma = h'$.

As standard for such an intuitionistic semantics, implication is closed under all worlds reachable according to such a relation. Thus satisfaction $P \models^h \phi \supset \psi$, involving an implication, is defined such that under all Q, h' and σ, whenever $P, h \leq_\sigma Q, h'$ we have $Q \models^{h'} \phi\sigma \supset Q \models^{h'} \psi\sigma$. This Kripke semantics gives rise to the definition of implication provided in Fig. 2.

Modalities must also be closed under all respectful substitutions. There is an asymmetry in the definition of these modalities. Unlike the box modality, for the diamond modality we need not close the definition under all reachable worlds.

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.

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

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 interesting 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 $[\tau]\mathtt{ff}$.

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

$$\overline{a}b \parallel c(x) \not\models \langle\tau\rangle\mathtt{tt} \qquad \text{and} \qquad \overline{a}b \parallel c(x) \not\models \neg\langle\tau\rangle\mathtt{tt}$$

The former can hold only if $\overline{a}b \parallel c(x)$ 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 $\overline{a}b \parallel c(x) \not\models \langle\tau\rangle\mathtt{tt}$. For the latter, we should consider all substitutions which enable a τ-transition; and, since such a substitution $\{^c/_a\}$ exists, $\overline{a}b \parallel c(x) \not\models \neg\langle\tau\rangle\mathtt{tt}$.

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 $\overline{a}b \parallel c(x)$, that do not satisfy $\langle\tau\rangle\mathtt{tt}$, but also do not satisfy $\neg\langle\tau\rangle\mathtt{tt}$. Hence in the intuitionistic setting we **cannot** appeal to this principal of classical modal logic.

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

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

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 $0 \models \mathtt{ff}\{^y/_x\}$ 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.

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

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

$$\overline{a}b \parallel c(x) \models \neg([\tau]\mathtt{ff} \wedge \langle\tau\rangle\mathtt{tt}) \qquad \text{but} \qquad \overline{a}b \parallel c(x) \not\models \langle\tau\rangle\mathtt{tt} \vee [\tau]\mathtt{ff}$$

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 actions 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 $\overline{a}b$.*
- *If $P \xrightarrow{\overline{a}(x)} P'$, for x fresh for P, Q and h, there exists Q' such that $Q \xrightarrow{\overline{a}(x)} Q'$ and $P'\ \mathcal{R}^{h\cdot x^o}\ Q'$.*
- *If $P \xrightarrow{a(x)} P'$, for x is fresh for P, Q and 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 $P\ \mathcal{R}^{x_0^i\cdot\ldots\cdot x_n^i}\ Q$, where $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \{x_0, \ldots, x_n\}$.

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)**.** *For π-calculus processes P and Q (including replication), If $P \sim Q$ then for all $\mathcal{OM}$ formulae ϕ, $P \models \phi$ iff $Q \models \phi$.*

Theorem 3.3 (completeness)**.** *For finite π-calculus processes P and Q, 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 λProlog [NM88]. The proof of soundness proceeds by induction on the structure of the logical formulae in the definition of logical equivalence. An explanation of the soundness proof and mechanisation we defer until Section 5.

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 it is impossible to prove that two process are 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 to certify when 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 non-bisimilar processes. 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.

$[x = y]\tau \nsim \tau$**:** The distinguishing strategy for these processes is 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 process. Since process τ leads in the distinguishing strategy, $\langle\tau\rangle\mathtt{tt}$ is a distinguishing formula biased to process τ, as follows.

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

As remarked in the previous section, negating formula $\langle\tau\rangle\mathtt{tt}$ 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 post-condition that holds after a τ-transition is enabled, i.e. $x = y$. This gives rise to the following distinguishing formula, as required.

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

$[x = y]\tau \nsim 0$**:** 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.

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

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

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

Now consider the *inductive case* of an algorithm for constructing distinguishing formulae. In an inductive case, 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 non-bisimilar to P'. This allows a distinguishing 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.

$$P \triangleq \tau.[x = y]\tau + \tau + \tau.\tau \quad \not\sim \quad \tau + \tau.\tau \triangleq Q$$
$$P \xrightarrow{\tau} [x = y]\tau \qquad Q \xrightarrow{\tau} 0 \qquad Q \xrightarrow{\tau} \tau$$

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.

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

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:

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

To confirm that the above are indeed distinguishing formulae for P and Q, assume for contradiction that $Q \models \langle\tau\rangle([\tau](x = y) \wedge (x = y \supset \langle\tau\rangle\mathtt{tt}))$ holds. By definition of diamond modalities, this holds iff either $0 \models [\tau](x = y) \wedge (x = y \supset \langle\tau\rangle\mathtt{tt})$ or $\tau \models [\tau](x = y) \wedge (x = y \supset \langle\tau\rangle\mathtt{tt})$ holds. Observe that $0 \models x = y \supset \langle\tau\rangle\mathtt{tt}$ 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 know that the following hold:

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

Notice that $x = y \vee \neg(x = y)$ is an instance of the law of excluded middle for name 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.

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

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 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.

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

These processes are not open bisimilar because P can make the following three transition steps: $\nu x.\overline{a}x.a(y).\tau \xrightarrow{\overline{a}(x)} a(y).\tau \xrightarrow{a(y)} \tau \xrightarrow{\tau} 0$. 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 $[x = y]\tau\theta \xrightarrow{\tau} 0$ 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.

$$[x = y]\tau \vDash^{a^i \cdot x^o \cdot y^i} [\tau](x = y) \qquad \text{and} \qquad \tau \vDash^{a^i \cdot x^o \cdot y^i} \langle\tau\rangle\mathtt{tt}$$

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

$$\nu x.\overline{a}x.a(y).\tau \vDash \langle\overline{a}(x)\rangle\langle a(y)\rangle\langle\tau\rangle\mathtt{tt} \qquad \text{and} \qquad \nu x.\overline{a}x.a(y).[x = y]\tau \vDash [\overline{a}(x)][a(y)][\tau](x = y)$$

3.1.3. *Example involving private names that are indistinguishable.* In contrast to the previous example, consider the following processes where a fresh name is output and compared to a name already known.

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

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

To see why, observe $\nu x.\overline{a}x \vDash^{a^i} \langle\overline{a}(x)\rangle(x = a \supset \langle\tau\rangle\mathtt{tt})$ holds if and only if $\nu x.\overline{a}x \xrightarrow{\overline{a}(x)} 0$ and $0 \vDash^{a^i \cdot x^o} x = a \supset \langle\tau\rangle\mathtt{tt}$. 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 $0 \vDash^{a^i \cdot x^o} \langle\tau\rangle\mathtt{tt}$. 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 $0 \vDash^{a^i \cdot x^o} x = a \supset \langle\tau\rangle\mathtt{tt}$ holds vacuously; hence we have that $\nu x.\overline{a}x \vDash^{a^i} \langle\overline{a}(x)\rangle(x = a \supset \langle\tau\rangle\mathtt{tt})$ holds as required.

4. Completeness of open bisimilarity with respect to $\mathcal{OM}$

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

Since non-bisimilarity is defined in terms of a least fixed point, there is a winning strategy, consisting of a finite tree of moves. We inductively define non-bisimilarity in terms of a family of relations on processes indexed by a history $\not\sim_n$, for $n \in \mathbb{N}$. The base case is when, for some respectful substitution one player can make a move, that cannot be matched by the other player without assuming a stronger 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 in $\nsim_0$, at which point there is an accessible world in which a process can make a move that the other process cannot match.

Definition 4.1 (non-bisimilarity). *Relation $\nsim_0$ is the least relation such that $P \nsim_0^h Q$ 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 case, we require that if $x \in \mathrm{bn}(\pi)$, then x is fresh for $P\sigma$, $Q\sigma$ and $h\sigma$.

Inductively, $\nsim_{n+1}$ is the least relation extending $\nsim_n$ such that $P \nsim_{n+1}^h Q$ whenever for some substitution σ respecting h, one of the following holds, where, in the following, α is τ or $\overline{a}b$ 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* $Q\sigma \xrightarrow{\alpha\sigma} Q_i$, $P' \nsim_n^{h\sigma} Q_i$.
- $\exists P'.\ P\sigma \xrightarrow{\overline{a\sigma}(x)} P'$, *and,* $\forall Q_i$ *such that* $Q\sigma \xrightarrow{\overline{a\sigma}(x)} Q_i$, $P' \nsim_n^{h\sigma\cdot x^o} Q_i$.
- $\exists P'.\ P\sigma \xrightarrow{a\sigma(x)} P'$, *and,* $\forall Q_i$ *such that* $Q\sigma \xrightarrow{a\sigma(x)} Q_i$, $P' \nsim_n^{h\sigma\cdot x^i} Q_i$.
- $\exists Q'.\ Q\sigma \xrightarrow{\alpha\sigma} Q'$ *and* $\forall P_i$ *such that* $P\sigma \xrightarrow{\alpha\sigma} P_i$, $P_i \nsim_n^{h\sigma} Q'$.
- $\exists Q'.\ Q\sigma \xrightarrow{\overline{a\sigma}(x)} Q'$, *and,* $\forall P_i$ *such that* $P\sigma \xrightarrow{\overline{a\sigma}(x)} P_i$, $Q' \nsim_n^{h\sigma\cdot x^o} P_i$.
- $\exists Q'.\ Q\sigma \xrightarrow{a\sigma(x)} Q'$, *and,* $\forall P_i$ *such that* $P\sigma \xrightarrow{a\sigma(x)} P_i$, $Q' \nsim_n^{h\sigma\cdot x^i} P_i$.

The relation $\nsim$, pronounced non-bisimilarity, is defined to be the least relation containing $\nsim_n$ for all $n \in \mathbb{N}$, i.e. $\bigcup_{n\in\mathbb{N}} \nsim_n$. Define $P \nsim Q$ whenever $P \nsim^{x_1^i \cdots x_m^i} Q$ where $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \{x_1, \dots x_m\}$.

Lemma 4.2. *The relations $\nsim$ and $\nsim_n$, for all $n \geq 0$, are symmetric.*

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 4.3. *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, $\sigma \leq \theta$ whenever there exists σ' such that $\sigma \cdot \sigma' = \theta$. For a finite substitution $\sigma = \{^{z_1}\!/\!_{x_1}\} \cdots \{^{z_n}\!/\!_{x_n}\}$ the formula $[\sigma]\phi$ abbreviates the 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 $\mathtt{ff}$. Similarly $\bigwedge_i \phi_i$ abbreviates $\phi_1 \wedge \dots \wedge \phi_n$, where the empty conjunction is $\mathtt{tt}$.*

We require the following technical lemmas. The first, image finiteness, ensures that there are finitely many reachable states in one step, up to renaming. The second unfolds the definition of box substitution abbreviation, defined above. The third is required in inductive cases involving bound output and input. The fourth is a monotonicity property for satisfaction. The fifth is a monotonicity property for transitions, assuming names bound by labels are not changed by a substitution.

Lemma 4.4. *For process P and action π there are finitely many P_i such that $P \xrightarrow{\pi} P_i$.*

Lemma 4.5. *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 4.6. *If $\sigma \cdot \theta$ respects h, then θ respects $h\sigma$.*

Lemma 4.7. *If $P \models^h \phi$ holds then $P\theta \models^{h\theta} \phi\theta$ holds for any θ respecting h.*

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

4.2. **Algorithm for distinguishing formulae.** The direct definition of non-bisimilarity gives 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 capture such strategies. For any strategy that distinguishes two processes, we can construct *distinguishing* $\mathcal{OM}$ formulae. A distinguishing formula holds for one process but not for the other process. Furthermore, there are always at least two distinguishing formulae, one biased to the left and another biased to the right, as in the construction of the proof for the following proposition. As discussed in Section 2, 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 $P \nsim Q$ then there exists ϕ_L such that $P \models \phi_L$ and $Q \not\models \phi_L$, and also there exists ϕ_R such that $Q \models \phi_R$ and $P \not\models \phi_R$.*

Proof. Since $\nsim$ is defined by a least fixed point over a family of relations $\nsim_n$, if $P \nsim^h Q$, there exists n such that $P \nsim^h_n Q$, so we can proceed by induction on the depth of a winning strategy.

In the base case, assume $P \nsim^h_0 Q$, hence by definition, for substitution σ respecting h, $P\sigma \xrightarrow{\pi\sigma} P'$, for $x \in \mathrm{bn}(\pi)$, x is fresh for $P\sigma$, $Q\sigma$ and $h\sigma$, such that there is no Q' such that $Q\sigma \xrightarrow{\pi\sigma} Q'$, up to symmetry of $\nsim^h_n$.

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 $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \cup \mathrm{fn}(\pi)$ 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 $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \cup \mathrm{fn}(\pi)$ such that $x\sigma$ and $y\sigma$ are distinct, and $x\theta = y\theta$. Stated otherwise, for all x and y in $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \cup \mathrm{fn}(\pi)$ if $x\theta = y\theta$ then $x\sigma = y\sigma$, which is precisely the definition of a function, i.e., 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 $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \cup \mathrm{fn}(\pi)$; 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 $[\sigma]\langle\pi\rangle\mathtt{tt}$ and $[\pi]\bigvee_j \left(x_j = y_j\right)$ biased to P and Q respectively. There are four cases to check to confirm that these are distinguishing formulae.

Case $P \models^h [\sigma]\langle\pi\rangle\mathtt{tt}$: 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 $P'\theta' \models^{h'} \mathtt{tt}$ holds, $P\theta \models^{h\theta} \langle\pi\theta\rangle\mathtt{tt}$. Hence, by Lemma 4.5, $P \models^h [\sigma]\langle\pi\rangle\mathtt{tt}$.

Case $Q \not\models^h [\sigma]\langle\pi\rangle\mathtt{tt}$: For contradiction, assume $Q \models^h [\sigma]\langle\pi\rangle\mathtt{tt}$. Since σ respects h and $\sigma \leq \sigma$, by Lemma 4.5, $Q \models^h [\sigma]\langle\pi\rangle\mathtt{tt}$ holds only if $Q\sigma \models^{h\sigma} \langle\pi\sigma\rangle\mathtt{tt}$ holds; which holds only if there exists Q' such that $Q\sigma \xrightarrow{\pi\sigma} Q'$, contradicting the assumption no such Q' exists. Thereby $Q \not\models^h [\sigma]\langle\pi\rangle\mathtt{tt}$.

Case $Q \models^h [\pi]\bigvee_j \left(x_j = y_j\right)$: 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 $Q' \models^{h\theta} \bigvee_j \left(x_j = y_j\right)\theta$ holds. Hence $Q \models^h [\pi]\bigvee_j \left(x_j = y_j\right)$.

Case $P \not\models^h [\pi]\bigvee_j \left(x_j = y_j\right)$: Assume for contradiction $P \models^h [\pi]\bigvee_j \left(x_j = y_j\right)$. This holds iff for all processes S and substitutions θ respecting h, $P\theta \xrightarrow{\pi\theta} S$ implies $S \models^{h'} \bigvee_j \left(x_j = y_j\right)\theta$. Since we know that σ respects h and $P\sigma \xrightarrow{\pi\sigma} P'$, for some h'', we have $P' \models^{h''} \bigvee_j \left(x_j = y_j\right)\sigma$. 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 $P \not\models^h [\pi]\bigvee_j \left(x_j = y_j\right)$.

Now consider the inductive cases. Given P, Q, if $P \nsim^{h}_{n+1} Q$, up to symmetry of $\nsim^{h}_{n+1}$, there are three cases to consider, for some substitution σ respecting h, where α is either τ or $\overline{a}b$, 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 $Q\sigma \xrightarrow{\alpha\sigma} Q_i$, $P' \nsim^{h\sigma}_{n} Q_i$.
- $P\sigma \xrightarrow{\overline{a\sigma}(x)} P'$, and, for all Q_i such that $Q\sigma \xrightarrow{\overline{a\sigma}(x)} Q_i$, $P' \nsim^{h\sigma\cdot x^o}_{n} Q_i$.
- $P\sigma \xrightarrow{a\sigma(x)} P'$, and, for all Q_i such that $Q\sigma \xrightarrow{a\sigma(x)} Q_i$, $P' \nsim^{h\sigma\cdot x^i}_{n} Q_i$.

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 $P\sigma \xrightarrow{\overline{a\sigma}(x)} P'$, by Lemma 4.4, there exist finitely many Q_i such that $Q\sigma \xrightarrow{\overline{a\sigma}(x)} Q_i$. For each i, since $P' \nsim^{h\sigma\cdot x^o}_{n} Q_i$, by the induction hypothesis, there exist ϕ^L_i and ϕ^R_i such that $P' \models^{h\sigma\cdot x^o} \phi^L_i\sigma$ and $Q_i \not\models^{h\sigma\cdot x^o} \phi^L_i\sigma$ and $P' \not\models^{h\sigma\cdot x^o} \phi^R_i\sigma$ and $Q_i \models^{h\sigma\cdot x^o} \phi^R_i\sigma$.

We require the following property, referred to later using †. There are finitely many pairs of variables x_j and y_j selected from $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \cup \{a\}$ 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 $h \cdot x^o$), such that $\sigma \leq \theta$, and for any S such that, $Q\theta \xrightarrow{\overline{a\theta}(x)} S$ then either: for some i, we have $S \models^{h\theta\cdot x^o} \phi^R_i\theta$, 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 $S \models^{h\rho\cdot x^o} \phi^R_i\rho$, and also there is no pair of variables u and v in $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \cup \{a\}$ 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 $Q_i \models^{h\sigma\cdot x^o} \phi^R_i\sigma$, by Lemma 4.7, we have $Q_i\sigma' \models^{h\rho\cdot x^o} \phi^R_i\rho$, i.e., $S \models^{h\rho\cdot x^o} \phi^R_i\rho$, contradicting the assumption no such i exists.

From the above, distinguishing formulae $[\sigma]\langle\overline{a}(x)\rangle\bigwedge_i \phi^L_i$ and $[\sigma][\overline{a}(x)]\left(\bigvee_i \phi^R_i \vee \bigvee_j \left(x_j = y_j\right)\right)$ can be constructed. There are four cases to consider to verify these are distinguishing formulae.

Case $P \models^h [\sigma]\langle\overline{a}(x)\rangle\bigwedge_i \phi^L_i$: Consider all θ such that $\sigma \leq \theta$, θ respects h, and without loss of generality x is fresh for θ, i.e., for $y \in \mathrm{dom}(\theta)$ 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 4.6, θ' respects $h\sigma$ hence since $x \notin \mathrm{dom}(\theta')$ and $x \notin \mathrm{fn}(h\sigma\theta')$, θ' respects $h\sigma \cdot x^o$. Thereby since θ' respects $h\sigma \cdot x^o$ and also $P' \models^{h\sigma\cdot x^o} \phi^L_i\sigma$ holds, by Lemma 4.7, it holds that $P'\theta' \models^{h\theta\cdot x^o} \phi^L_i\theta$. The above holds for all i, hence it holds that $P'\theta' \models^{h\theta\cdot x^o} \bigwedge_i \phi^L_i\theta$. 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 $P\theta \models^{h\theta} \left(\langle\overline{a}(x)\rangle\bigwedge_i \phi^L_i\right)\theta$ holds. Thereby, by Lemma 4.5, $P \models^h [\sigma]\langle\overline{a}(x)\rangle\bigwedge_i \phi^L_i$ holds.

Case $Q \not\models^h [\sigma]\langle\overline{a}(x)\rangle\bigwedge_i \phi^L_i$: Assume for contradiction that $Q \models^h [\sigma]\langle\overline{a}(x)\rangle\bigwedge_i \phi^L_i$ holds. Since σ respects h and $\sigma \leq \sigma$, by Lemma 4.5, the above assumption holds only if $Q\sigma \models^{h\sigma} \left(\langle\overline{a}(x)\rangle\bigwedge_i \phi^L_i\right)\sigma$ holds. Now $Q\sigma \models^{h\sigma} \langle\overline{a\sigma}(x)\rangle\bigwedge_i \phi^L_i\sigma$ holds only if there exists Q' such that $Q\sigma \xrightarrow{\overline{a\sigma}(x)} Q'$ and $Q' \models^{h\sigma\cdot x^o} \bigwedge_i \phi^L_i\sigma$, which holds only if $Q' \models^{h\sigma\cdot x^o} \phi^L_i\sigma$ for all i. Notice that $Q' = Q_k$ for some k, and therefore $Q_k \models^{h\sigma\cdot x^o} \phi^L_k\sigma$; but it was assumed that $Q_k \not\models^{h\sigma\cdot x^o} \phi^L_k\sigma$ leading to a contradiction. Therefore $Q \not\models^h [\sigma]\langle\overline{a}(x)\rangle\bigwedge_i \phi^L_i$.

Case $Q \models^h [\sigma][\overline{a}(x)]\left(\bigvee_i \phi^R_i \vee \bigvee_j \left(x_j = y_j\right)\right)$: 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

$Q' \vDash^{h\theta \cdot x^o} \phi^R_\ell \theta$, 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 $Q' \vDash^{h\theta \cdot x^o} x_k\theta = y_k\theta$ holds. Hence in either case we have $Q' \vDash^{h\theta \cdot x^o} \left(\bigvee_i \phi^R_i \vee \bigvee_j \left(x_j = y_j\right)\right)\theta$, by definition of disjunction. Thereby, by definition, $Q \vDash^h [\sigma][\overline{a}(x)]\left(\bigvee_i \phi^R_i \vee \bigvee_j \left(x_j = y_j\right)\right)$ holds.

Case $P \nvDash^h [\sigma][\overline{a}(x)]\left(\bigvee_i \phi^R_i \vee \bigvee_j \left(x_j = y_j\right)\right)$**:** Assume $P \vDash^h [\sigma][\overline{a}(x)]\left(\bigvee_i \phi^R_i \vee \bigvee_j \left(x_j = y_j\right)\right)$ for contradiction. Since σ respects h, $\sigma \leq \sigma$, and $P\sigma \xrightarrow{\overline{a\sigma}(x)} P'$, the previous assumption can hold only if $P' \vDash^{h\sigma \cdot x^o} \left(\bigvee_i \phi^R_i \vee \bigvee_j \left(x_j = y_j\right)\right)\sigma$. This holds only if, for some i, $P' \vDash^{h\sigma \cdot x^o} \phi^R_i\sigma$, or, for some j, $P' \vDash^{h\sigma \cdot x^o} \left(x_j\sigma = y_j\sigma\right)$. However, for all i, $P' \nvDash^{h\sigma \cdot x^o} \phi^R_i\sigma$; and also, for all j, we have $x_j\sigma$ and $y_j\sigma$ are distinct and $P' \nvDash^{h\sigma \cdot x^o} \left(x_j\sigma = y_j\sigma\right)$, leading to a contradiction in either case. Thereby $P \nvDash^h [\sigma][\overline{a}(x)]\left(\bigvee_i \phi^R_i \vee \bigvee_j \left(x_j = y_j\right)\right)$.

By induction we have established that, for any history h, processes P and Q, and any n, if $P \nsim^h_n Q$ then we can construct ϕ_L such that $P \vDash^h \phi_L$ and $Q \nvDash^h \phi_L$; and also we can construct ϕ_R such that $Q \vDash^h \phi_R$ and $P \nvDash^h \phi_R$. The result then follows by observing that, since $\nsim$ is the least relation containing all $\nsim_n$ whenever $P \nsim Q$; there exists n such that $P \nsim^{x^i_1 \cdot \ldots \cdot x^i_n}_n Q$ and, where $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \left\{x^i_1, \ldots, x^i_n\right\}$; for which, there is ϕ^L such that $P \vDash^{x^i_1 \cdot \ldots \cdot x^i_n} \phi_L$ and $Q \nvDash^{x^i_1 \cdot \ldots \cdot x^i_n} \phi_L$; and also ϕ^R such that $Q \vDash^{x^i_1 \cdot \ldots \cdot x^i_n} \phi_R$ and $P \nvDash^{x^i_1 \cdot \ldots \cdot x^i_n} \phi_R$. Hence, by Definition 2.2, indeed $P \vDash \phi_L$, $Q \nvDash \phi_L$, $Q \vDash \phi_R$ and $P \nvDash \phi_R$ as required. □

4.3. **The proof of completeness.** Since open bisimilarity is decidable for finite π-calculus processes, Definition 4.1 coincides with the negation of open bisimilarity.

Lemma 4.10. *For finite processes, $P \nsim Q$ holds, according to non-bisimilarity in Definition 4.1, if and only if $P \sim Q$ does not hold.*

Combining Proposition 4.9 with Lemma 4.10 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, $P \vDash \phi$ iff $Q \vDash \phi$. Now for contradiction suppose that $P \sim Q$ does not hold. By Lemma 4.10, $P \nsim Q$ must hold. Hence by Proposition 4.9 there exists ϕ_L such that $P \vDash \phi_L$ but $Q \nvDash \phi_L$, but by the assumption above $Q \vDash \phi_L$, leading to a contradiction. Thereby $P \sim Q$. □

Notice that soundness (Theorem 3.2) and the non-bisimilarity algorithm (Proposition 4.9) also hold for infinite π-calculus processes (using replication for instance). However, for infinite π-calculus processes, open bisimilarity is undecidable; hence additional insight may be needed to justify whether Lemma 4.10 holds for infinite processes. Thereby in the infinite case, while it is impossible both $P \sim Q$ and $P \nsim Q$ hold, it may be possible neither holds. This is an open problem.

4.4. **Example runs of distinguishing formulae algorithm.** We provide further examples of non-bisimilar processes 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 non-bisimilar processes.

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

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 $\tau \models \langle\tau\rangle\mathtt{tt}$ is biased to the right.

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

$$P \triangleq \tau.\left(\overline{a}a + \overline{b}b\right) + [x = y]\tau.\overline{a}a \qquad \not\sim \qquad \tau.\left(\overline{a}a + \overline{b}b\right) + \tau.\overline{a}a \triangleq Q$$

To distinguishing these processes $Q \xrightarrow{\tau} \overline{a}a$ leads, which can only be matched by $P \xrightarrow{\tau} \overline{a}a + \overline{b}b$. By the previous example, we have distinguishing formulae for $\overline{a}a + \overline{b}b \not\sim \overline{a}a$. However, also observe, for substitutions θ such that $x\theta = y\theta$, there is an additional τ-transitions enabled: $P\theta \xrightarrow{\tau} \overline{a}a$.

This leads to the following distinguishing formula for the left side, consisting of a box τ followed by a disjunction comprised of the left distinguishing formula for $\overline{a}a + \overline{b}b \not\sim \overline{a}a$, and the postcondition $x = y$ guaranteed after the additional τ transition.

$$\tau.\left(\overline{a}a + \overline{b}b\right) + [x = y]\tau.\overline{a}a \models [\tau]\left(\langle\overline{b}b\rangle\mathtt{tt} \vee x = y\right)$$

The distinguishing formula for the right process is diamond τ followed by the right distinguishing formula for $\overline{a}a + \overline{b}b \not\sim \overline{a}a$, as follows: $\tau.\left(\overline{a}a + \overline{b}b\right) + \tau.\overline{a}a \models \langle\tau\rangle[\overline{b}b](a = b)$.

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 non-bisimilar processes: $\overline{a}a \not\sim \overline{a}b$. A distinguishing strategy is where process $\overline{a}b$ makes a $\overline{a}b$ transition, which cannot be matched by $\overline{a}a$. However, $(\overline{a}a)\sigma \xrightarrow{(\overline{a}b)\sigma} 0$ for any substitution such that $a\sigma = b\sigma$, leading to distinguishing formula $[\overline{a}b](a = b)$ biased to $\overline{a}a$. Notice substitution σ is applied to both the process and the label.

For a trickier example consider the following processes.

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

After two actions, the above problem reduces to base case $[x = b]\overline{x}x \not\sim^{a^i \cdot b^o \cdot x^i} \overline{x}x$, where $\overline{x}x$ can perform a $\overline{x}x$ action, but $[x = b]\overline{x}x$ cannot. However, $([x = b]\overline{x}x)\{^b/_x\} \xrightarrow{\overline{x}x\{^b/_x\}} 0$ 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.

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

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 $\langle\tau\rangle\mathtt{tt}$ 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 $x = y \supset [\tau]\langle\tau\rangle\mathtt{tt}$ biased to the process on the right, which is indeed distinguishing.

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

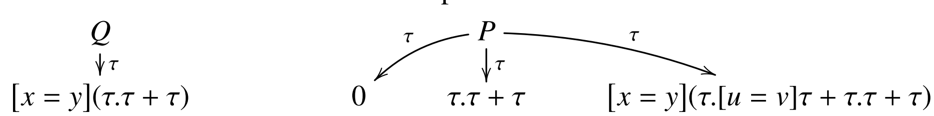
$$[x = y]\tau.\tau + \tau \quad \not\sim \quad \tau.\tau + \tau$$

The following is a distinguishing formula biased to the left process: $[\tau][\tau](x = y)$. However, this is different from the left-biased formula $[\tau]([\tau]\mathtt{ff} \vee (x = y))$ generated by the algorithm. Thus, there exist alternative distinguishing formulae . . . and alternative algorithms.

4.4.4. *A more elaborate example.* Consider the following processes.

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

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.



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.

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

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

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. Mechanised Proofs 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 λProlog [MN12]. The formalisation of the modal logic $\mathcal{OM}$ in this section is built on top existing work on the formalisation of the pi-calculus and bisimulation based on the higher-order abstract syntax (HOAS) approach [BGM+07, TM10, BCG+14]. We give an overview of the λProlog specification of the

π-calculus labelled transition semantics (Section 5.1) and the coinductive definition of open bisimilarity in Abella (Section 5.2); for more details, the reader is referred to the literature, in particular [TM10, BCG+14]. We present the formalisation of the semantics of the modal logic $\mathcal{OM}$ and explain our mechanised proof of the soundness theorem (Theorem 3.2) in Section 5.3.

5.1. Specification of the syntax and labelled transition semantics of the π-calculus in λProlog.

Recall the syntax and labelled transition semantics in Fig. 1 from Section 2. Fig. 3 is a transcription of the syntactic constructs and transition rules in λProlog. 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 λProlog where some of which may require multiple arguments to construct the desired syntactic category. For instance, `plus` needs two process arguments to construct a process as its type (`p`→`p`→`p`) suggests. The table below summarises the process syntax in λProlog and the notations used in the previous sections.

λProlog syntax	mathematical notation	
`null`	0	
`nu x\P x` (or `nu P`)	$\nu z.P$	`(P z)` corresponds to P
`taup P`	$\tau.P$	
`out x z P`	$\overline{x}z.P$	
`in x z\P z` (or `in x P`)	$x(z).P$	`(P z)` corresponds to P
`match x y P`	$[x = y]P$	
`par P Q`	$P \parallel Q$	
`plus P Q`	$P + Q$	

Fig. 3 has few stylistic differences from Fig. 1. Firstly, distinct constants are used for actions and its related processes (e.g., `tau` action for `taup` prefixed process) because the constants cannot be overloaded as in mathematical notations. 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 ($\overline{x}z$ and $\overline{x}(z)$) 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 set of transition relations are defined: `one` relating a process (`p`) with another process (`p`) via a free action (`a`) and `oneb` relating a process (`p`) with a bound process (`n`→`p`) via a bound action (`n`→`a`).

One advantage of using λProlog [MN12] is that we can rely on its native support for a variant of HOAS, known as *λ-tree syntax* [MP99], for handling bound names and $\alpha\beta\eta$-equivalence automatically. For instance, consider the bound output transition rule for name restricted process from both Fig. 3 and Fig. 1:

```
oneb (nu z\P z) (up X) Q :-
   pi z\one (P z) (up X z) (Q z)
```

$$\frac{P \xrightarrow{\overline{x}z} Q}{\nu z.P \xrightarrow{\overline{x}(z)} Q}\; x \neq z$$

There is no need to explicitly state and keep track of the side conditions such as $x \neq z$ and $x \notin \mathrm{n}(\pi)$ in λProlog definitions. For example, consider `pi z\one (P z) (up X z) (Q z)` from above. Here, it is guaranteed that `z` does not to 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

```
sig finite-pic. % file: finite-pic.sig

kind n  type. % names

kind p  type. % processes
type null         p.                 % deadlock
type taup         p → p.             % progress action
type plus, par    p → p → p.         % choice, par
type match, out   n → n → p → p.     % match, output action
type in           n → (n → p) → p.   % input action
type nu           (n → p) → p.       % nu

kind a  type. % actions (transition labels)
type tau          a.
type up, dn       n → n → a.

type one     p →     a   →     p   → o.  % one step  free transition
type oneb    p → (n → a) → (n → p) → o.  % one step bound transition


module finite-pic. % file: finite-pic.mod

oneb (in X M) (dn X) M.      % bound input
one (out X Y P) (up X Y) P.  % free output
one (taup P) tau P.          % tau
% match prefix
one (match X X P) A Q :- one  P A Q.     oneb (match X X P) A M :- oneb P A M.
% sum
one (plus P Q) A R :- one  P A R.        oneb (plus P Q) A M :- oneb P A M.
one (plus P Q) A R :- one  Q A R.        oneb (plus P Q) A M :- oneb Q A M.
% par
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.
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.
% restriction
one  (nu z\P z) A (nu z\Q z)     :- pi z\ one  (P z) A (Q z).
oneb (nu z\P z) A (y\nu z\Q z y) :- pi z\ oneb (P z) A (y\Q x z).
% open (bound output)
oneb (nu x\P x) (up X) Q :- pi z\ one (P y) (up X y) (Q y).
% close
one (par P Q) tau (nu z\par (M z) (N z)) :- oneb P (dn X) M, oneb Q (up X) N.
one (par P Q) tau (nu z\par (M z) (N z)) :- oneb P (up X) M, oneb Q (dn X) N.
% comm (interaction)
one (par P Q) tau (par (M Y) T) :- oneb P (dn X) M, one Q (up X Y) T.
one (par P Q) tau (par R (M Y)) :- oneb Q (dn X) M, one P (up X Y) R.
```

Figure 3: λProlog specification of the syntax and semantics of the π-calculus. (Adopted from one of the examples distributed with Abella.)

```
1  Specification "finite-pic". % load the finite pi-calc. spec. in Fig. 3
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 ∨, ∧, ⊃         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 (∨ 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 6.1 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 6.1 for related discussion)
32  ; sat P (◇↓X A) := ∃ Q, {oneb P (dn X) Q} ∧  ∀ z, sat (Q z) (A z).
```

Figure 4: A coinductive definition of open bisimulation and an inductive definition of the modal logic $\mathcal{OM}$ in Abella.

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`.

5.2. **Coinductive definition of open bisimulation in Abella's reasoning logic.** Open bisimulation relation `bisim` is coinductively defined in Fig. 4. 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., `{one P A P1}`) are used for referring to the object logic proposition (i.e., λProlog proposition) from the reasoning logic of Abella. Recall the type of the λProlog relation `one : p→a→p→o` in Fig. 3. When applied to three arguments, it becomes an object logic proposition `one P A P1 : o`. When we need to to refer to the λProlog proposition from Abella's reasoning logic, we must convert form the λProlog proposition type (`o`) to the reasoning logic proposition type (`prop`) using curly braces. For example, `{one P A P1} : prop`. Abella's reasoning logic is richer than the object logic. It supports coinductive definitions, which we used to define `bisim`. It also supports a nominal quantifier (∇), which will be discussed shortly, in addition to universal (∀) and existential (∃) 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 `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 lead the explanation on 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 below:

```
     ∀x,∇z,∀y, bisim null null              ⋮
   ----------------------------------------------
   ∀x,∇z, bisim (in x y\null) (in x y\null)                ⋮
-----------------------------------------------------------------------
∀x, bisim (nu z\out x z (in x y\null)) (nu z\out x z (in x y\null))
```

$$\dfrac{\dfrac{0 \sim^{x^i \cdot z^o \cdot y^i} 0 \quad \vdots}{x(y).0 \sim^{x^i \cdot z^o} x(y).0} \quad \vdots}{\nu z.\overline{x}z.x(y).0 \sim^{x^i} \nu z.\overline{x}z.x(y).0}$$

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 `∇z` in Abella, which corresponds to extending the history with z^o. The input step extends the environment with `∀y` in Abella, which corresponds to extending the history with y^i. These quantified variables come from the definition of `bisim` in Fig. 4, 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 output variable 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 with respect to all the previously introduced variables. For instance, consider the environment `∀x,∇z,∀y,···`. Abella ensures that `z` cannot occur free in `x`, hence, `x` cannot be unified with `z`; however, `y` can be unified with `z` because `y` is introduced after `z`.

Intuitively, universal quantification represents all possible substitutions over universally quantified variables. For example, consider `∀ x, pred x`. Proving this in Abella means that the predicate `pred` holds for all possible substitutions over `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 names before the `bisim` relation in Abella not only transcribes the history but also represents all possible respectful substitutions.

5.3. **Embedding of $\mathcal{OM}$ in Abella and the soundness proof.** The latter part of Fig. 4 is an embedding of $\mathcal{OM}$ syntax and semantics 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. 4). There are similar stylistic difference regarding $\mathcal{OM}$ formulae in the Abella embedding (Fig. 4) and notation from 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.[2]

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 on $\mathcal{OM}$ semantics are handled exactly the same manner in Abella and enjoy the closure properties regarding respectful substitutions. For example, `∀x,∇z,∀y, sat (P x y z)` ϕ 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. 4 is an embedding of $\mathcal{OM}$ semantics ($\models$) in Abella. There is no explicit handing of substitutions in the `sat` relation definition 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 : ∀ P Q F, form F →
                                  bisim P Q → ((sat P F → sat Q F) ∧ (sat Q F → sat P F)).
```

where `form: o′→prop` is an inductive predicate for well-formed $\mathcal{OM}$ formulae, defined as follows:

```
Define form : o′ → prop
by form tt
 ; form ff
 ; form (∨ A B) := form A ∧ form B
 ; form (∧ A B) := form A ∧ form B
 ; form (⊃ A B) := form A ∧ form B
 ; form (≠ X Y) := form A
 ; form (□ X A) := form A
 ; form (◇ X A) := form A
 ; form (□↑X A) := ∀ w, form (A w)
 ; form (◇↑X A) := ∀ w, form (A w)
 ; form (□↓X A) := ∀ w, form (A w)
 ; form (◇↓X A) := ∀ w, form (A w).
```

The proof proceeds by induction on the structure of the modal formulae and case analyses on the definition of the satisfiability relation `sat`. The soundness theorem is a simple corollary of `bisim_sat`:

```
Theorem soundness: ∀ P Q,
  bisim P Q → ∀ F, form F → (sat P F → sat Q F) ∧ (sat Q F → sat P F).
```

6. 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.

[2] In the Abella proof scripts, $\Box$, $\Box^{\uparrow}$ and $\Box^{\downarrow}$ are represented using keywords `boxAct`, `boxOut` and `boxIn`, respectively.

6.1. **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, that is, the law of excluded middle for name equalities is induced by eagerly applying substitutions and assuming all names are distinct. Secondly, the late input box modality is defined differently, involving an existential quantification over substitutions. Moving to the intuitionistic setting, this gives rise to the following variant of the box input modality.

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

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.

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

Recall from Sec. 5, 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. 4).

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 non-bisimilar processes.

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

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

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

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 $[a(x)]^L(\langle\tau\rangle\mathtt{tt} \vee [\tau]\mathtt{ff})$. To see why $a(x).\tau + a(x) + a(x).[x = a]\tau \models [a(x)]^L(\langle\tau\rangle\mathtt{tt} \vee [\tau]\mathtt{ff})$ holds, observe 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 $([x = a]\tau)\{^a/_x\} \models \langle\tau\rangle\mathtt{tt} \vee [\tau]\mathtt{ff}$.

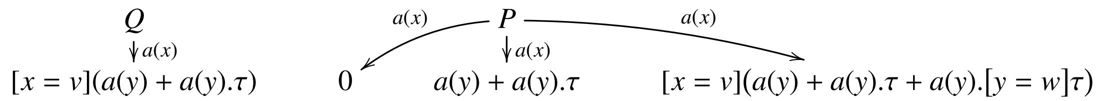
Also, note the formula $[a(x)]^L((x = a \wedge [\tau]\mathtt{ff}) \vee \langle\tau\rangle\neg(x = a))$ fails in intuitionistic $\mathcal{LM}$ for both processes, despite being distinguishing for these processes in classical $\mathcal{LM}$. Although, if we use triple negation the formula becomes distinguishing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

$$a(x).\tau + a(x) \models [a(x)]^L((x = a \wedge [\tau]\mathtt{ff}) \vee \langle\tau\rangle\neg\neg\neg(x = a)) \tag{6.1}$$

6.1.1. *Example where box input is necessary.* Notice, for the above example processes, formula $\langle a(x)\rangle(x = a \supset \langle\tau\rangle\mathtt{tt} \wedge [\tau](x = a))$ 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.



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.

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.

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

For a formula biased to Q we obtain the following.

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

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.

6.1.2. *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:

```
Type □↓L      n → (n → o′) → o′.
```

The clauses for the satisfaction relation (encoded as the predicate `satLM`) are those for $\mathcal{OM}$ (Figure 4) without the "basic" box operator, but with the following clause for `satLM` :

```
satLM P (□↓L X A) := ∀ Q, {oneb P (dn X) Q} → ∃ z, sat (Q z) (A z).
```

The example involving triple negation above (6.1) 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 principals 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.

6.2. **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. Fortunately, this criteria holds for $\mathcal{OM}$ — the classical counterpart to $\mathcal{OM}$ characterises late bisimilarity. For convenience we, recall the definition of late bisimilarity.

Definition 6.1. *A late bisimulation $\mathcal{R}$ is a symmetric relation, such that, whenever $P \mathrel{\mathcal{R}} Q$:*

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

Late bisimilarity $\dot{\sim}_L$ is the greatest late bisimulation.

$$
\begin{array}{lll}
P \mathrel{\dot{\models}_L} \mathtt{tt} & \text{and} \quad P \mathrel{\dot{\models}_L} x = x & \text{always hold.} \\
P \mathrel{\dot{\models}_L} \phi_1 \supset \phi_1 & \text{iff} & P \mathrel{\dot{\models}_L} \phi_1 \implies P \mathrel{\dot{\models}_L} \phi_2. \\
P \mathrel{\dot{\models}_L} \langle \alpha \rangle \phi & \text{iff} & \exists Q,\ P \xrightarrow{\alpha} Q \text{ and } Q \mathrel{\dot{\models}_L} \phi. \\
P \mathrel{\dot{\models}_L} \langle a(z) \rangle \phi & \text{iff} & \exists Q,\ P \xrightarrow{a(z)} Q \text{ and, } \forall y,\ Q\{^y\!/\!_z\} \mathrel{\dot{\models}_L} \phi\{^y\!/\!_z\}. \\
P \mathrel{\dot{\models}_L} [a(z)]\phi & \text{iff} & \forall Q, P \xrightarrow{a(z)} Q \implies \forall z, Q \mathrel{\dot{\models}_L} \phi.
\end{array}
$$

Figure 5: Semantics of "classical $\mathcal{OM}$", where α is τ, $\overline{a}b$ or $\overline{a}(z)$.

A direct semantics of classical $\mathcal{OM}$, in the style of Milner, Parrow and Walker [MPW93], is presented in Fig. 5. Observe histories are not employed in the classical semantics since inputs are instantiated eagerly, immediately after performing an input transition. Also, missing operators (conjunction, disjunction, and $[\alpha]\phi$) are derivable using classical negation; whereas in an intuitionistic modal logic they have independent interpretations. Classical $\mathcal{OM}$ characterises late bisimilarity.

Corollary 6.2 (characterisation)**.** *$P \mathrel{\dot{\sim}_L} Q$ if and only if, for all $\mathcal{OM}$ formulae ϕ, we have $P \mathrel{\dot{\models}_L} \phi$ iff $Q \mathrel{\dot{\models}_L} \phi$, according to the classical semantics for $\mathcal{OM}$ in Fig. 5.*

Proof. Observe that the definition of $\langle a(x) \rangle \phi$ in Figure 5 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. The original paper on modal logics for mobile processes establishes that, classical $\mathcal{LM}$ characterises late bisimilarity, and also $\mathcal{LM}$ extend with basic modalities has the same expressive power at $\mathcal{LM}$. □

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 has a simpler characterisation in terms of closure under substitutions defined as follows.

Definition 6.3. *P is late equivalent to Q, written $P \sim_L Q$, whenever there exists a late bisimulation $\mathcal{R}$ such that for all σ, $P\sigma \mathrel{\mathcal{R}} Q\sigma$. Define $P \models_L \phi$ whenever for all σ, $P\sigma \mathrel{\dot{\models}_L} \phi\sigma$.*

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

Proof. This follows immediately from Corollary 6.2 and the following facts. For a fresh for P, Q and ϕ, $a(x_1)\dots a(x_n).P \mathrel{\dot{\sim}_L} a(x_1)\dots a(x_n).Q$ if and only if $P \sim_L Q$, and $P \models_L \phi$ if and only if $a(x_1)\dots a(x_n).P \mathrel{\dot{\models}_L} \langle a(x_1)\rangle\dots\langle a(x_n)\rangle\phi$ where $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \cup \mathrm{fn}(\phi) \subseteq \{x_1, \dots, x_n\}$. □

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: $P \models_L x = y \supset \langle\tau\rangle\mathtt{tt}$ and $Q \models_L [\tau]\mathtt{ff}$.

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 $\dot{\models}_L$, characterising late bisimilarity.

6.3. **A sharpened picture of the classical/intuitionistic spectrum.** We emphasise here that open bisimilarity is not the only bisimilarity congruence. Another, strictly coarser, bisimilarity congruence for the π-calculus is *open barbed bisimilarity* [SW01], which is defined to be the greatest bisimilarity congruence, defined as follows.

Definition 6.5. *Process P has a barb x, written $P\!\downarrow\! x$, whenever $P \xrightarrow{x(z)} P'$ or $P \xrightarrow{\overline{x}y} P'$ or $P \xrightarrow{\overline{x}(z)} P'$. An open barbed bisimulation $\mathcal{R}$ is a symmetric relation such that, whenever $P \mathrel{\mathcal{R}} Q$ we have:*

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

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.

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

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

Definition 6.6. *A intermediate bisimulation $\mathcal{R}$ is a symmetric relation indexed by a set of names, such that, whenever $P \mathrel{\mathcal{R}^{\mathcal{E}}} Q$ the following hold:*

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

Intermediate bisimilarity $\sim_I$ is the greatest intermediate bisimulation.

Intermediate bisimilarity, 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 $\nu k.\overline{a}k.(a(x).\tau + a(x))$ and $\nu k.\overline{a}k.(a(x).[x = k]\tau + a(x).\tau + a(x))$ are distinguished by intermediate bisimilarity, as witnessed by the following strategy.

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

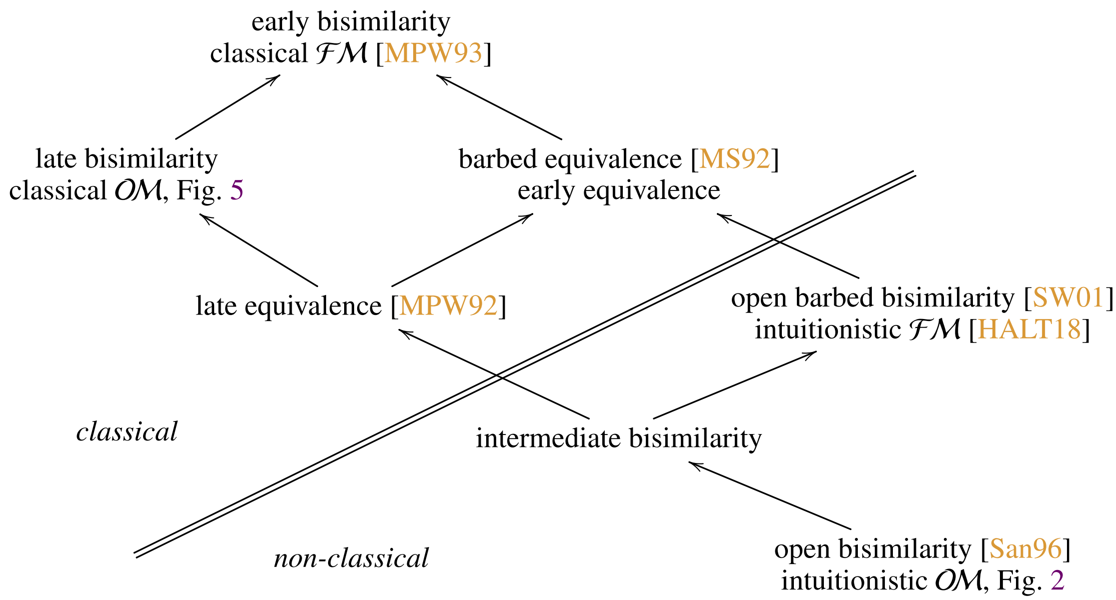


Figure 6: The line between classical and non-classical notions of bisimilarity.

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.

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

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

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

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 the spectrum surrounding open bisimilarity is provided in Fig. 6. To complete this picture, 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 argues the merits of open barbed bisimilarity due to its more objective definition and granularity suitable for verifying privacy properties.

6.4. **Related work: a generic formalisation 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 the abstract framework for open bisimilarity in the π-calculus, but for a fragment without input prefixes.

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 semantic 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.

7. 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 5. The details of the completeness, Theorem 3.3, are provided in Section 4.

Intuitionistic modal logic $\mathcal{OM}$ satisfies the following established criteria [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 all substitutions respectful of the environment.
- 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 6.4, 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.

There are further criteria expected for an intuitionistic logic. A more direct proof calculus for $\mathcal{OM}$ is left as an open problem, in place of the Kripke semantics and intuitionistic meta-framework embedding of Figures 2 and 4. A proof calculus 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 calculus would be a step towards addressing the following, more philosophical, criterion:

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

We remark that previous work on intuitionistic modal logic for program analysis [PS86, SI94], is motivated by the undecidability of termination, and preserving liveness/safety properties. This is quite different to this work, where the source of intuitionistic content is the lazy approach to inputs.

The main novelty of this paper is the completeness proof involving an algorithm, Proposition 4.9, constructing distinguishing formulae for non-bisimilar processes. To use this algorithm, firstly attempt to prove that two processes are open bisimilar. If they are non-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., $x = y \supset \langle \pi \rangle \mathtt{tt}$, and box and postconditions for the following process, e.g., $[\pi](x = y)$. Numerous examples involving post conditions are provided throughout, e.g., throughout Section 4.4.

Our logic $\mathcal{OM}$ is suitable for formal and automated reasoning; in particular, 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 [AHT17b].

Future work includes justifying whether or not $\mathcal{OM}$ is complete for infinite processes with replication or recursion, as discussed following Lemma 4.10. Another problem is to extend $\mathcal{OM}$ with fixed points, as in the μ-calculus [Koz83]. Such an extension could lead to intuitionistic model checkers invariant under open bisimulation, where the call-by-need approach to inputs is related to symbolic execution. We have already lifted open bisimulation and $\mathcal{OM}$ to the π-calculus with mismatch [HALT18]. Future work will lift $\mathcal{OM}$ and open bisimulation to the applied π-calculus [ABF18], generalising work on open bisimulation for the spi-calculus [TD10, BN06].

Acknowledgments

We are grateful to Sam Staton for providing an example that helped us in the completeness proof. We also appreciate the comments of anonymous reviewers significantly improving results. 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.

[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.

[AHT17b] Ki Yung Ahn, Ross Horne, and Alwen Tiu. Generating witness of non-bisimilarity for the pi-calculus. *CoRR*, abs/1705.10908, 2017.

[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.

[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.

[BN06] Sébastien Briais and Uwe Nestmann. Open bisimulation, revisited. *Electr. Notes Theor. Comput. Sci.*, 154(3):109–123, 2006.

[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.

[HALT18] Ross Horne, Ki Yung Ahn, Shang-Wei Lin, and Alwen Tiu. Quasi-open bisimilarity with mismatch is intuitionistic. In *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.

[HM85] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[Koz83] Dexter Kozen. Results on the propositional mu-calculus. *Theoretical Comput. Sci.*, 27(3):333–354, 1983.

[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.

[MN12] Dale Miller and Gopalan Nadathur. *Programming with Higher-Order Logic*. Cambridge University Press, 2012.

[MP99] Dale Miller and Catuscia Palamidessi. Foundational aspects of syntax. *ACM Computing Surveys*, 31, 1999.

[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.

[MPW93] Robin Milner, Joachim Parrow, and David Walker. Modal logics for mobile processes. *Theoretical Computer Science*, 114(1):149–171, 1993.

[MS92] Robin Milner and Davide Sangiorgi. Barbed bisimulation. In W. Kuich, editor, *Automata, Languages and Programming*, pages 685–695, 1992.

[NM88] Gopalan Nadathur and Dale Miller. An Overview of λProlog. In *Fifth International Logic Programming Conference*. MIT Press, 1988.

[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.

[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.

[San96] Davide Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33(1):69–97, 1996.

[SI94] Bernhard Steffen and Anna Ingolfsdottir. Characteristic formulas for processes with divergence. *Information and Computation*, 110(1):149 – 163, 1994.

[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.

[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.

[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.

[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.