Weak Nominal Modal Logic

Joachim Parrow, Tjark Weber, Johannes Borgström, Lars-Henrik Eriksson

Weak Nominal Modal Logic

Joachim Parrow, Tjark Weber, Johannes Borgström, and Lars-Henrik Eriksson

Department of Information Technology, Uppsala University, Sweden

Abstract. Previous work on nominal transition systems explores strong bisimulation and a general kind of Hennessy-Milner logic with infinite but finitely supported conjunction, showing that it is remarkably expressive. In the present paper we treat weak bisimulation and the corresponding weak Hennessy-Milner logic, where there is a special unobservable action. We prove that logical equivalence coincides with bisimilarity and explore a few variants of the logic. In this way we get a general framework for weak bisimulation and logic in which formalisms such as the pi-calculus and its many variants can be uniformly represented.

1 Introduction

In many models of concurrent computation there is a fundamental distinction between two kinds of actions: on one hand, those that are strictly internal to a process, and thus cannot be observed by its environment; on the other hand, those that represent an interaction with the environment and thus are observable. The discriminatory power of the model must then be weak enough, roughly speaking, that unobservables do not count. This idea emerged in the early 1980s in a variety of concurrency models, for example in Milner's observation equivalence, Lamport's notion of stuttering, and the denotational models of Hoare [20,19,6]. A good example is weak bisimulation in numerous process calculi. Here the special action τ represents anything unobservable, and the bisimulation game requires a simulating process to mimic actions with the same observable content, i.e., it is allowed to have more or fewer τs. Similarly, the so called weak modal logics cannot express formulas to test for the presence or absence of τs.

In our earlier work [23] we develop a theory of nominal transition systems, bisimulation, and modal logic, with the goal to be as general as possible and subsume many models in the literature. The states of the transition systems may be tested by state predicates from an arbitrary logic. Transitions between states can take arbitrarily structured labels and also bind names (like in the scope extrusions of the pi-calculus). Thus we can uniformly represent not only the pi-calculus but also many of its high-level extensions. Our results include a treatment of bisimulation and an adequate Hennessy-Milner logic (HML) where logical equivalence coincides with bisimilarity. We make ample comparisons to other work to support our claim that their primitives can be encoded in our general framework. Main technical points include the use of nominal sets to represent how states, actions and predicates depend on names, and the use of finitely supported infinite conjunctions in the logic to represent a variety of quantifiers

and fixpoints. Section 2 below recapitulates the necessary background. All of that work is of the so-called strong variety: all actions are counted as observable.

In this paper we extend our investigation to nominal transition systems where there is a special unobservable action τ. In Section 3 we define and explore the notion of weak bisimulation. In comparison to existing work in process algebra there are subtleties in the interplay of unobservable actions and state predicates. In Section 4 we introduce a weak HML and prove that its induced logical equivalence coincides with weak bisimulation. The logic is formulated as a sublogic of our earlier logic [23] and contains only formulas that do not distinguish between weakly bisimilar states. Again the main subtlety is in the interplay between state predicates and action modalities. The logic does not admit disjunctions of state predicates, and in Section 5 we prove that this does not affect the expressive power. In Section 6 we demonstrate that state predicates can be encoded as additional transitions: self loops labelled with the predicate. Section 7 describes how our results can apply to existing models of computation, Section 8 relates to existing work on weak modal logics, and in Section 9 we conclude with a summary of the main insights gained and prospects for further work.

Our main results in Sections 3 and 4, including the adequacy of the weak logic, have been formalised in the interactive theorem prover Isabelle/HOL using the nominal datatype package. Our Isabelle theories, comprising approximately 1,300 lines of machine-readable definitions and proofs, are available from the Archive of Formal Proofs.[1] They extend an earlier formalisation [25] of nominal transition systems and our logic for strong bisimilarity, from which they re-use the definition of (strong) formulas.

2 Background

In this section we recapitulate the relevant definitions from our earlier work [23], to which we refer for more extensive explanations, examples, and relation to previous work on transition systems and Hennessy-Milner logics.

2.1 Nominal sets

Nominal sets [24] is a general theory of objects that contain names, and in particular formulates the notion of alpha-equivalence when names can be bound. The reader need not know nominal set theory to follow this paper, but some key definitions will make it easier to appreciate our work, and we recapitulate them here.

We assume a countably infinite multi-sorted set of atomic identifiers or *names* $\mathcal{N}$ ranged over by a, b, ... A *permutation* is a bijection on names that leaves all but finitely many names invariant. The singleton permutation that swaps names a and b and has no other effect is written $(a\,b)$, and the identity permutation, which swaps nothing, is written id. Permutations are ranged over by π, π'. The

[1] `https://devel.isa-afp.org/entries/Modal_Logics_for_NTS.shtml`

effect of applying a permutation π to an object X is written $\pi \cdot X$. Formally, the permutation action $\cdot$ can be any operation that satisfies $\mathrm{id} \cdot X = X$ and $\pi \cdot (\pi' \cdot X) = (\pi \circ \pi') \cdot X$, but a reader may comfortably think of $\pi \cdot X$ as the object obtained by permuting all names in X according to π.

A set of names N *supports* an object X if for all π that leave all elements of N invariant it holds $\pi \cdot X = X$. In other words, if N supports X then names outside N do not matter to X. If a finite set supports X then there is a unique minimal set supporting X, called the *support* of X, written $\mathrm{supp}(X)$, intuitively consisting of exactly the names that matter to X. In general, the support of a set is not the same as the union of the support of its members. An example is the set of all names; the support of each element a is the set $\{a\}$, but the whole set has empty support since $\pi \cdot \mathcal{N} = \mathcal{N}$ for any permutation π.

We write $a\#X$, pronounced "a is fresh for X," for $a \notin \mathrm{supp}(X)$. The intuition is that if $a\#X$ then X does not depend on a in the sense that a can be replaced with any fresh name without affecting X. If A is a set of names we write $A\#X$ for $\forall a \in A \,.\, a\#X$.

A *nominal set* S is a set with a permutation action such that $X \in S$ implies $\pi \cdot X \in S$, and where each member $X \in S$ has finite support. A main point is that then each member has infinitely many fresh names available for alpha-conversion.

A set of names N supports a function f on a nominal set if for all π that leave all elements of N invariant it holds $\pi \cdot f(X) = f(\pi \cdot X)$, and similarly for relations and functions of higher arity. Thus we extend the notion of support to finitely supported functions and relations as the minimal finite support, and can derive general theorems such as $\mathrm{supp}(f(X)) \subseteq \mathrm{supp}(f) \cup \mathrm{supp}(X)$.

An object that has empty support is called *equivariant*. For instance, a unary function f is equivariant if $\pi \cdot f(X) = f(\pi \cdot X)$ for all π, X. The intuition is that an equivariant object does not treat any name special.

2.2 Nominal transition systems

Definition 1. *A* nominal transition system *is characterised by the following*

- STATES*: A nominal set of* states *ranged over by P, Q.*
- PRED*: A nominal set of* state predicates *ranged over by φ.*
- *An equivariant binary relation $\vdash$ on* STATES *and* PRED*. We write $P \vdash \varphi$ to mean that in state P the state predicate φ holds.*
- ACT*: A nominal set of* actions *ranged over by α.*
- *An equivariant function* bn *from* ACT *to finite sets of names, which for each α returns a subset of* $\mathrm{supp}(\alpha)$*, called the* binding names*.*
- *An equivariant transition relation $\rightarrow$ on states and residuals. A residual is a pair of action and state. For $\rightarrow (P, (\alpha, P'))$ we write $P \xrightarrow{\alpha} P'$. The transition relation must satisfy alpha-conversion of residuals: If $a \in \mathrm{bn}(\alpha)$, $b\#\alpha, P'$ and $P \xrightarrow{\alpha} P'$ then also $P \xrightarrow{(a\,b)\cdot\alpha} (a\,b) \cdot P'$.*

In [23] we motivate and demonstrate many examples of nominal transition systems, including the pi-calculus and several extensions of it. Here states, actions and transitions are familiar, and the binding names correspond to the names in scope extrusions. State predicates represent what the environment can perceive of a state, for example equality tests of expressions, or connectivity between communication channels.

Definition 2. *A* bisimulation R *is a symmetric binary relation on states in a nominal transition system satisfying the following two criteria:* $R(P, Q)$ *implies*

1. Static implication*:* $P \vdash \varphi$ *implies* $Q \vdash \varphi$.
2. Simulation*: For all* α, P' *such that* $\mathrm{bn}(\alpha)\#Q$ *there exist* Q' *such that if* $P \xrightarrow{\alpha} P'$ *then* $Q \xrightarrow{\alpha} Q'$ *and* $R(P', Q')$

We write $P \dot{\sim} Q$ *to mean that there exists a bisimulation* R *such that* $R(P, Q)$.

Static implication and symmetry means that bisimilar states must satisfy the same state predicates. The simulation requirement is familiar from the pi-calculus.

2.3 Hennessy-Milner logic

We define a Hennessy-Milner logic including infinitary conjunctions; as demonstrated in [23] this results in high expressiveness using a very compact formal definition. In order to avoid set-theoretic paradoxes we begin by fixing some infinite cardinal κ to bound the cardinality of conjunctions. We define the formulas, ranged over by $A, B, \ldots$, and the validity of a formula A in a state P, written $P \models A$, by induction as

Definition 3.

$$
\begin{array}{ll}
P \models \bigwedge_{i \in I} A_i & \text{if for all } i \in I \text{ it holds that } P \models A_i \\
P \models \neg A & \text{if not } P \models A \\
P \models \varphi & \text{if } P \vdash \varphi \\
P \models \langle \alpha \rangle A & \text{if there exists } P' \text{ such that } P \xrightarrow{\alpha} P' \text{ and } P' \models A
\end{array}
$$

Support and name permutation are defined as usual (permutation distributes over all formula constructors). In $\bigwedge_{i \in I} A_i$ it is required that the indexing set I has bounded cardinality, by which we mean that $|I| < \kappa$. We assume that κ is sufficiently large; specifically, we require $\kappa > \aleph_0$ (so that we may form countable conjunctions) and $\kappa > |\textsc{states}|$. It is also required that the set of conjuncts $\{A_i \mid i \in I\}$ has finite support; this is then the support of the conjunction. This is strictly weaker than requiring the set to be uniformly bounded, i.e., that there is a finite set of names supporting all members. Alpha-equivalent formulas are identified; the only binding construct is in $\langle \alpha \rangle A$ where $\mathrm{bn}(\alpha)$ binds into A. In the last clause we assume that $\langle \alpha \rangle A$ is a representative of its alpha-equivalence class such that $\mathrm{bn}(\alpha)\#P$.

We write $\top$ for the empty conjunction and $A_0 \wedge A_1$ for the binary conjunction $\bigwedge_{i\in\{0,1\}} A_i$. Bounded and finitely supported disjunction $\bigvee$ is defined in the usual way as the dual of conjunction. Universal and existential quantifiers are defined as conjunction and disjunction over the set of instances. In [23] we expand on the expressive power and relate to existing logics.

Definition 4. *Two states P and Q are* logically equivalent*, written $P \dot{=} Q$, if for all A it holds that $P \models A$ iff $Q \models A$.*

Theorem 1. *(Theorems 6 and 9 in [23]) $P \dot{\sim} Q$ iff $P \dot{=} Q$*

The implication from left to right is by induction over formulas. The other direction is by contraposition: if not $P \dot{\sim} Q$ then there is a distinguishing formula A such that $P \models A$ and not $Q \models A$.

3 Weak bisimulation

The logics and bisimulations considered in [23] are of the strong variety, in the sense that all transitions are regarded as equally significant. In many models of concurrent computation there is a special action that is *unobservable* in the sense that in a bisimulation, and also in the definition of the action modalities, the presence of extra such transitions does not matter. This leads to notions of *weak* bisimulation and accompanying weak modal logics. For example, a process that has no transitions is weakly bisimilar to any process that has only unobservable transitions, and these satisfy the same weak modal logic formulas. We shall here introduce these ideas into the nominal transition systems, where the presence of state predicates requires some care in the definitions.

To cater for unobservable transitions assume a special action τ with empty support. The following definitions are standard:

Definition 5.

1. *$P \Rightarrow P'$ is defined by induction to mean $P = P'$ or $P \xrightarrow{\tau} \circ \Rightarrow P'$.*
2. *$P \stackrel{\alpha}{\Rightarrow} P'$ means $P \Rightarrow \circ \xrightarrow{\alpha} \circ \Rightarrow P'$.*
3. *$P \stackrel{\hat{\alpha}}{\Rightarrow} P'$ means $P \Rightarrow P'$ if $\alpha = \tau$ and $P \stackrel{\alpha}{\Rightarrow} P'$ otherwise.*

Intuitively $P \stackrel{\hat{\alpha}}{\Rightarrow} P'$ means that P can evolve to P' through transitions with the only observable content α. We call this a weak action α and it will be the basis for the semantics in this section.

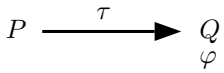
The normal way to define weak bisimilarity is to weaken $Q \xrightarrow{\alpha} Q'$ to $Q \stackrel{\hat{\alpha}}{\Rightarrow} Q'$ in the simulation requirement. This results in the weak simulation criterion:

Definition 6. *A binary relation R on* STATES *is a* weak simulation *if $R(P, Q)$ implies that for all α, P' with $\mathrm{bn}(\alpha)\#Q$ there exists Q' such that*

$$\textit{if } P \xrightarrow{\alpha} P' \textit{ then } Q \stackrel{\hat{\alpha}}{\Rightarrow} Q' \textit{ and } R(P', Q')$$

However, just replacing the simulation requirement with weak simulation in Definition 2 will not suffice. The reason is that through the static implication criterion in Definition 2, an observer can still observe the state predicates directly, and thus distinguish between a state that satisfies φ and a state that does not but can silently evolve to another state that satisfies φ:

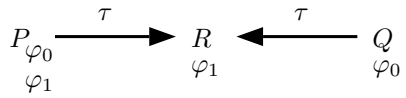
Example 1.



Certainly $\{(P,Q),(Q,Q)\}$ is a weak simulation according to Definition 6. But $P \not\vdash \varphi$ and $Q \vdash \varphi$, thus they are in no static implication. We argue that if φ is the *only* state predicate (in particular, there is no predicate $\neg\varphi$), then the only test that an observer can apply is "if φ then ...," and here P and Q will behave the same; P can pass the test after an unobservable delay. Thus P and Q should be deemed weakly bisimilar, and static implication as in Definition 2 is not appropriate.

Therefore we need a weak counterpart of static implication where τ transitions are admitted before checking predicates, that is, if $P \vdash \varphi$ then $Q \Rightarrow Q' \vdash \varphi$. In other words, Q can unobservably evolve to a state that satisfies φ. However, this is not quite enough by itself. Consider the following example where $P \vdash \varphi_0$, $P \vdash \varphi_1$, $R \vdash \varphi_1$ and $Q \vdash \varphi_0$, with transitions $P \xrightarrow{\tau} R$ and $Q \xrightarrow{\tau} R$:

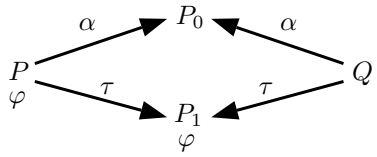
Example 2.



Here we do not want to regard P and Q as weakly bisimilar. They do have the same transitions and can satisfy the same predicates, possibly after a τ transition. But an observer of P can first determine that φ_1 holds, and then determine that φ_0 holds. This is not possible for Q: an observer who concludes φ_1 must already have evolved to R.

Similarly, consider the following example where the only difference between P and Q is that $P \vdash \varphi$ but not $Q \vdash \varphi$:

Example 3.



Again we do not want to regard P and Q as weakly bisimilar. Intuitively, an observer of Q that determines that φ holds must already be at P_1 and thus have preempted the possibility to do α, whereas for P, the predicate φ holds while retaining the possibility to do α. For instance, P in parallel with a process of kind "if φ then γ" can perform γ followed by α, but Q in parallel with the same cannot do that sequence.

In conclusion, the weak counterpart of static implication should allow the simulating state to proceed through unobservable actions to a state that *both* satisfies the same predicate *and* continues to bisimulate. This leads to the following:

Definition 7. *A binary relation R on states is a* weak static implication *if $R(P, Q)$ implies that for all φ there exists Q' such that*

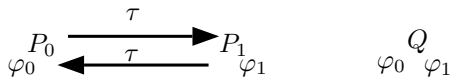
$$\textit{if } P \vdash \varphi \textit{ then } Q \Rightarrow Q' \textit{ and } Q' \vdash \varphi \textit{ and } R(P, Q')$$

Definition 8. *A* weak bisimulation *is a symmetric binary relation on states satisfying both weak simulation and weak static implication. We write $P \dot{\approx} Q$ to mean that there exists a weak bisimulation R such that $R(P, Q)$.*

In Example 1, $\{(P, Q), (Q, P), (Q, Q)\}$ is a weak bisimulation. In Examples 2 and 3, P and Q are not weakly bisimilar.

It is interesting to compare this with weak bisimilarities defined for psi-calculi [16]. A psi-calculus contains a construct of kind "if φ then ..." to test if a state predicate is true. These constructs may be nested; for instance, "if φ_0 then if φ_1 then ..." effectively tests if both φ_o and φ_1 are true simultaneously. If state predicates are closed under conjunction, Definition 8 coincides with the definition of simple weak bisimulation in [16]. In general, however, Definition 8 is less discriminating. Consider $P_0 \xrightarrow{\tau} P_1 \xrightarrow{\tau} P_0$ where for $i = 0, 1$: $P_i \vdash \varphi_i$. Compare it to Q with no transitions where both $Q \vdash \varphi_0$ and $Q \vdash \varphi_1$:

Example 4.



Here all of P_0, P_1 and Q are weakly bisimilar, unless the predicates are closed under conjunction, in which case the predicate $\varphi_0 \wedge \varphi_1$ distinguishes between them. In psi-calculi Q would not be simply weakly bisimilar to P_0 or P_1 for the same reason.

We proceed to establish some expected properties of weak bisimilarity.

Lemma 1. *If $P \dot{\approx} Q$ and $P \stackrel{\hat{\alpha}}{\Rightarrow} P'$ with $\mathrm{bn}(\alpha)\#Q$ then for some Q' it holds $P' \dot{\approx} Q'$ and $Q \stackrel{\hat{\alpha}}{\Rightarrow} Q'$.*

Proof. The proof has been formalised in Isabelle; it is by induction and case analysis according to Definition 5.

Lemma 2. $\dot{\approx}$ *is an equivariant equivalence relation.*

Proof. The proofs of equivariance, reflexivity, symmetry, and transitivity have been formalised in Isabelle.

4 Weak logic

We here define a Hennessy-Milner logic adequate for weak bisimilarity. Since weak bisimilarity identifies more states than strong bisimilarity, the logic needs to be correspondingly less expressive: it must not contain formulas that distinguish between weakly bisimilar states. Our approach is to keep the definition of formulas (Definition 3) and identify an adequate sublogic.

One main idea is to restrict the action modalities $\langle\alpha\rangle$ to occur only in accordance with the requirement of a weak bisimulation, thus checking for $\stackrel{\hat{\alpha}}{\Rightarrow}$ rather than for $\stackrel{\alpha}{\rightarrow}$. We therefore define the derived *weak action* modal operator $\langle\!\langle\alpha\rangle\!\rangle$ in the following way, where $\langle\tau\rangle^i A$ is defined to mean A if $i = 0$ and $\langle\tau\rangle\langle\tau\rangle^{i-1}A$ otherwise.

Definition 9 (Weak action modality).

$$\langle\!\langle\tau\rangle\!\rangle A = \bigvee_{i\in\omega} \langle\tau\rangle^i A \qquad\qquad \langle\!\langle\alpha\rangle\!\rangle A = \langle\!\langle\tau\rangle\!\rangle\langle\alpha\rangle\langle\!\langle\tau\rangle\!\rangle A \quad \textit{for } \alpha \neq \tau$$

Note that in $\langle\!\langle\alpha\rangle\!\rangle A$ the names in $\mathrm{bn}(\alpha)$ bind into A. As usual we consider formulas up to alpha-conversion in the standard sense, i.e., to prove a property of a formula it is enough to prove a property of an alpha-variant. It is then straightforward to show (and formalise in Isabelle) that $\langle\!\langle\alpha\rangle\!\rangle A$ corresponds to the weak transitions used in the definition of weak bisimilarity:

Proposition 1. *Assume* $\mathrm{bn}(\alpha)\#P$. *Then*

$$P \models \langle\!\langle\alpha\rangle\!\rangle A \quad \textit{iff} \quad \exists P'.\, P \stackrel{\hat{\alpha}}{\Rightarrow} P' \textit{ and } P' \models A$$

In particular, for $\alpha = \tau$, we have that $\langle\!\langle\tau\rangle\!\rangle A$ holds iff A holds after zero or more τ transitions.

Thus a first step towards a weak sublogic is to replace $\langle\alpha\rangle$ by $\langle\!\langle\alpha\rangle\!\rangle$ in Definition 3. By itself this is not enough; that sublogic is still too expressive. For instance, the formula φ asserts that φ holds in a state; this holds for Q but not for P in Example 1, even though they are weakly bisimilar.

To disallow φ as a weak formula we require that state predicates only occur guarded by a weak action $\langle\!\langle\tau\rangle\!\rangle$. This solves part of the problem. In Example 1 we can no longer use φ as a formula, and the formula $\langle\!\langle\tau\rangle\!\rangle\varphi$ holds of both P and Q. Still, in Example 1 there would be the formula $\langle\!\langle\tau\rangle\!\rangle\neg\varphi$ which holds for P but not for Q, and in Example 4 the formula $\langle\!\langle\tau\rangle\!\rangle(\varphi_0 \wedge \varphi_1)$ holds for Q but not for P_0. Clearly a logic adequate for weak bisimulation cannot have such formulas. The more draconian restriction that state predicates occur *immediately* under $\langle\!\langle\tau\rangle\!\rangle$

would indeed disallow both $\langle\!\langle\tau\rangle\!\rangle\neg\varphi$ and $\langle\!\langle\tau\rangle\!\rangle(\varphi_0 \wedge \varphi_1)$ but would also disallow any formula distinguishing between P and Q in Examples 2 and 3.

A solution is to allow state predicates under $\langle\!\langle\tau\rangle\!\rangle$, and never directly under negation or in conjunction with another state predicate. The logic is:

Definition 10 (Weak formulas). *The set of* weak formulas *is the sublogic of Definition 3 given by*

$$A \quad ::= \quad \bigwedge_{i\in I} A_i \quad | \quad \neg A \quad | \quad \langle\!\langle\alpha\rangle\!\rangle A \quad | \quad \langle\!\langle\tau\rangle\!\rangle(A \wedge \varphi)$$

Note that since $P \stackrel{\hat{\alpha}}{\Rightarrow} \circ \Rightarrow P'$ holds iff $P \stackrel{\hat{\alpha}}{\Rightarrow} P'$ we have that $\langle\!\langle\alpha\rangle\!\rangle\langle\!\langle\tau\rangle\!\rangle A$ is logically equivalent to $\langle\!\langle\alpha\rangle\!\rangle A$. We thus abbreviate $\langle\!\langle\alpha\rangle\!\rangle\langle\!\langle\tau\rangle\!\rangle(A\wedge\varphi)$ to $\langle\!\langle\alpha\rangle\!\rangle(A\wedge\varphi)$. We also abbreviate $\langle\!\langle\alpha\rangle\!\rangle(\top \wedge \varphi)$ to $\langle\!\langle\alpha\rangle\!\rangle\varphi$.

Compared to Definition 3, the state predicates can now only occur in formulas of the form $\langle\!\langle\tau\rangle\!\rangle(A \wedge \varphi)$, i.e., under a weak action, and not under negation or conjunction with another predicate. For instance, in Example 1 above, neither φ nor $\langle\!\langle\tau\rangle\!\rangle\neg\varphi$ are weak formulas, and in fact there is no weak formula to distinguish between P and Q. Similarly, in Example 4 $\langle\!\langle\tau\rangle\!\rangle(\varphi_0 \wedge \varphi_1)$ is not a weak formula, and no weak formula distinguishes between Q and P_i.

To argue that the logic still is expressive enough to provide distinguishing formulas for states that are not weakly bisimilar, consider Example 2 and the formula $\langle\!\langle\tau\rangle\!\rangle((\langle\!\langle\tau\rangle\!\rangle\varphi_0) \wedge \varphi_1)$ which holds for P but not for Q. Similarly, in Example 3 $\langle\!\langle\tau\rangle\!\rangle((\langle\!\langle\alpha\rangle\!\rangle\top) \wedge \varphi)$ holds for P but not for Q.

Definition 11. *Two states P and Q are* weakly logically equivalent*, written $P \dot{\equiv} Q$, if for all weak formulas A it holds that $P \models A$ iff $Q \models A$.*

Theorem 2. *If $P \dot{\approx} Q$ then $P \dot{\equiv} Q$*

Proof. The proof has been formalised in Isabelle. It is by induction over weak formulas.

Theorem 3. *If $P \dot{\equiv} Q$ then $P \dot{\approx} Q$*

Proof. The proof has been formalised in Isabelle. The idea is to prove that $\dot{\equiv}$ is a bisimulation by contraposition: for any non-bisimilar pair of states there exists a distinguishing weak formula.

5 Disjunction elimination

As defined in Section 2, disjunction is a derived logical operator, expressed through conjunction and negation. This is still true in the weak modal logic, but there is a twist in that neither general conjunctions nor negations may be applied to unguarded state predicates. The examples in Section 3 demonstrate why this restriction is necessary: negated or conjoined state predicates in formulas would mean that adequacy no longer holds. Interestingly, we can allow

disjunctions of unguarded predicates while maintaining adequacy; in fact, adding disjunction would not increase the expressive power of the logic. In this section we demonstrate this.

The *extended* weak logic is as follows, where a simultaneous induction defines both extended weak formulas (ranged over by E) and preformulas (ranged over by B) corresponding to subformulas with unguarded state predicates.

Definition 12 (Extended weak formulas E and preformulas B).

$$E \quad ::= \quad \textstyle\bigwedge_{i\in I} E_i \quad | \quad \neg E \quad | \quad \langle\!\langle \alpha \rangle\!\rangle E \quad | \quad \langle\!\langle \tau \rangle\!\rangle B$$

$$B \quad ::= \quad E \wedge B \quad | \quad \varphi \quad | \quad \textstyle\bigvee_{i\in I} B_i$$

The last clause in the definition of preformulas is what distinguishes this logic from the logic in Definition 10. (Thus an extended weak formula is also an ordinary weak formula if it does not contain a disjunction of unguarded state predicates.) For instance, $\langle\!\langle \tau \rangle\!\rangle(\varphi_0 \vee \varphi_1)$ is an extended weak formula, as is

$$\langle\!\langle \tau \rangle\!\rangle(((\langle\!\langle \beta \rangle\!\rangle \top) \wedge \varphi_0) \vee ((\langle\!\langle \gamma \rangle\!\rangle \top) \wedge \varphi_1))$$

saying that it is possible to do a sequence of unobservable actions such that either continuing with β and satisfying φ_0 hold, or continuing with γ and satisfying φ_1 hold.

Theorem 4. *For any extended weak formula E there is an (ordinary) weak formula $\Delta(E)$ such that $E \mathrel{\dot{\equiv}} \Delta(E)$.*

Proof. The idea is to push disjunctions in preformulas to top level using the fact that (finite) conjunction distributes over disjunction, and then use the fact that the action modality distributes over disjunction to transform disjunctions of preformulas into disjunctions of weak formulas.

6 State predicates as actions

We shall here demonstrate that omitting state predicates does not really entail a loss of expressiveness: for any transition system $\mathbf{T}$ there is another transition system $\mathcal{S}(\mathbf{T})$ where state predicates are replaced by self-loops. In this section we formally define this transformation $\mathcal{S}$ and derive some of its properties. To formulate this idea we introduce the notation $\textsc{states}_{\mathbf{T}}$ to mean the states in the transition system $\mathbf{T}$, and similarly for actions, bn, transitions, bisimilarity, etc.

Definition 13. *The function $\mathcal{S}$ from transition systems to transition systems is defined as follows:*

- $\textsc{states}_{\mathcal{S}(\mathbf{T})} = \textsc{states}_{\mathbf{T}}$
- $\textsc{act}_{\mathcal{S}(\mathbf{T})} = \textsc{act}_{\mathbf{T}} \uplus \textsc{pred}_{\mathbf{T}}$
- $\mathrm{bn}_{\mathcal{S}(\mathbf{T})}(\alpha) = \mathrm{bn}_{\mathbf{T}}(\alpha)$ *if* $\alpha \in \textsc{act}_{\mathbf{T}}$; $\mathrm{bn}_{\mathcal{S}(\mathbf{T})}(\varphi) = \emptyset$ *if* $\varphi \in \textsc{pred}_{\mathbf{T}}$
- $\textsc{pred}_{\mathcal{S}(\mathbf{T})} = \vdash_{\mathcal{S}(\mathbf{T})} = \emptyset$

– $P \xrightarrow{\alpha}_{\mathcal{S}(\mathbf{T})} P'$ *if* $P \xrightarrow{\alpha}_{\mathbf{T}} P'$ *(for* $\alpha \in \text{ACT}_{\mathbf{T}}$*);* $P \xrightarrow{\varphi}_{\mathcal{S}(\mathbf{T})} P$ *if* $P \vdash_{\mathbf{T}} \varphi$ *(for* $\varphi \in \text{PRED}_{\mathbf{T}}$*)*

It is easy to see that if $\mathbf{T}$ is a transition system then so is $\mathcal{S}(\mathbf{T})$. In particular equivariance of $\rightarrow_{\mathcal{S}(\mathbf{T})}$ follows from equivariance of $\rightarrow_{\mathbf{T}}$ and $\vdash_{\mathbf{T}}$ and the fact that the union of equivariant relations is equivariant.

Theorem 5. *If* $P \dot{\approx}_{\mathbf{T}} Q$ *then* $P \dot{\approx}_{\mathcal{S}(\mathbf{T})} Q$.

Proof. We prove that $\dot{\approx}_{\mathbf{T}}$ is a weak $\mathcal{S}(\mathbf{T})$-bisimulation.

Theorem 6. *If* $P \dot{\approx}_{\mathcal{S}(\mathbf{T})} Q$ *then* $P \dot{\approx}_{\mathbf{T}} Q$.

Proof. We prove that $\dot{\approx}_{\mathcal{S}(\mathbf{T})}$ is a weak $\mathbf{T}$-bisimulation. It needs a lemma that if $P \Rightarrow Q \Rightarrow R$ and $P \dot{\approx} R$ then $Q \dot{\approx} R$.

A corresponding transformation of weak formulas turns state predicates into actions in the following way.

Definition 14. *The partial function* $\mathcal{S}$ *from weak formulas on the transition system* $\mathbf{T}$ *to weak formulas on the transition system* $\mathcal{S}(\mathbf{T})$ *is defined by*

$$\mathcal{S}(\langle\!\langle\tau\rangle\!\rangle((\langle\!\langle\tau\rangle\!\rangle A) \wedge \varphi)) = \langle\!\langle\varphi\rangle\!\rangle\mathcal{S}(A)$$

and is homomorphic on the first three cases in Definition 10.

$\mathcal{S}$ is not total since a formula $\langle\!\langle\tau\rangle\!\rangle(A \wedge \varphi)$ is in its domain only when $A = \langle\!\langle\tau\rangle\!\rangle A'$ for some A'. It is easy to see that $\mathcal{S}$ is injective and surjective, i.e., every weak formula A on $\mathcal{S}(\mathbf{T})$ has a unique formula B on $\mathbf{T}$ such that $\mathcal{S}(B) = A$. We write $\mathcal{S}^{-1}$ for the inverse of $\mathcal{S}$. Thus

$$\mathcal{S}^{-1}(\langle\!\langle\varphi\rangle\!\rangle A) = \langle\!\langle\tau\rangle\!\rangle((\langle\!\langle\tau\rangle\!\rangle\mathcal{S}^{-1}(A)) \wedge \varphi)$$

and $\mathcal{S}^{-1}$ is homomorphic on all other operators.

Theorem 7. $P \models_{\mathcal{S}(\mathbf{T})} A$ *iff* $P \models_{\mathbf{T}} \mathcal{S}^{-1}(A)$

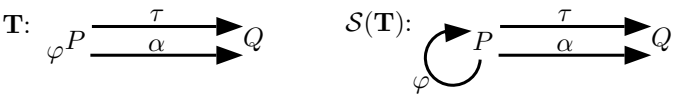
Proof. By induction over weak formulas on $\mathcal{S}(\mathbf{T})$.

An interesting consequence is that to express the distinguishing formulas guaranteed by Theorem 3, it is enough to consider formulas in $\text{dom}(\mathcal{S})$, i.e., in the last clause of Definition 10, it is enough to consider $A = \langle\!\langle\tau\rangle\!\rangle A'$. The reason is that if $P \not\dot{\approx}_{\mathbf{T}} Q$ then by Theorem 6 also $P \not\dot{\approx}_{\mathcal{S}(\mathbf{T})} Q$, which by Theorem 3 means there is a distinguishing formula B for P and Q in $\mathcal{S}(\mathbf{T})$, which by Theorem 7 means that $\mathcal{S}^{-1}(B)$ is a distinguishing formula in $\mathbf{T}$.

Finally, consider the apparently more appealing definition of $\mathcal{S}$ by

$$\mathcal{S}(\langle\!\langle\tau\rangle\!\rangle(A \wedge \varphi)) = \langle\!\langle\varphi\rangle\!\rangle\mathcal{S}(A)$$

Here $\mathcal{S}$ is total and a bijection, but with this definition, Theorem 7 fails. A counterexample is $A = \neg\langle\!\langle\alpha\rangle\!\rangle\top$, $P \vdash_{\mathbf{T}} \varphi$ with $P \xrightarrow{\tau}_{\mathbf{T}} Q$ and $P \xrightarrow{\alpha}_{\mathbf{T}} Q$ for some $\alpha \neq \tau$, where Q has no outgoing transitions, cf. the diagrams below:



Since $P \stackrel{\varphi}{\Rightarrow}_{\mathcal{S}(\mathbf{T})} Q$ and Q has no $\langle\langle\alpha\rangle\rangle$ action, we have that

$$P \models_{\mathcal{S}(\mathbf{T})} \langle\langle\varphi\rangle\rangle\neg\langle\langle\alpha\rangle\rangle\top$$

The only state that satisfies φ also has an $\langle\langle\alpha\rangle\rangle$ action, thus it does *not* hold that

$$P \models_{\mathbf{T}} \langle\langle\tau\rangle\rangle((\neg\langle\langle\alpha\rangle\rangle\top) \wedge \varphi)$$

7 Applications

In our earlier work [23] we outlined how several advanced process algebras can be given a semantics in terms of nominal transition systems. For all of these the present paper thus defines weak bisimulation, a weak HML, and an adequacy theorem. We here comment briefly on some of them.

The pi-calculus already has several notions of weak bisimulation, and Definition 8 corresponds to the early weak bisimulation. In the pi-calculus there are no state predicates, thus the weak static implication is unimportant. There is an HML adequate for strong bisimulation [22] but we are not aware of a weak HML. Our result here contributes a weak HML adequate for early weak bisimulation.

The applied pi-calculus [1] comes equipped with a labelled transition system and a notion of weak labelled bisimulation. States contain a record of emitted messages; this record has a domain and can be used to equate open terms M and N modulo some rewrite system. The definition of bisimulation requires bisimilar processes to have the same domain and equate the same open terms, i.e., to be strongly statically equivalent. In order to model this strong static equivalence in our weak logic, we add state predicates "$x \in \mathrm{dom}$" and "$M \equiv N$" to the labelled transition system. Since these are invariant under silent transitions, weak and strong static implication coincide, and our weak HML is adequate for Abadi and Fournet's early weak labelled bisimilarity.

The spi-calculus [2] has a formulation as an environment-sensitive labelled transition system [4] equipped with state formulae ϕ. As above, adding state predicates "$x \in \mathrm{dom}$" to this labelled transition system makes our weak HML adequate with respect to Boreale's weak bisimilarity.

Our earlier work also describes how to make nominal transition systems of multiple-labelled transition systems [11], the explicit fusion calculus [26], the concurrent constraint pi-calculus [7], and psi-calculi [16]. These calculi can become interesting applications of our ideas since they have actions with binders and nontrivial state predicates. Each of them has a special unobservable action, but until now only psi-calculi have a notion of weak labelled bisimulation (as remarked in Section 3), and none have a weak HML. Through this paper they all gain both bisimulation and logic, although more work is needed to establish how compatible the bisimulation equivalence is with their respective syntactic

constructs. A complication with all but the multiple-labelled systems is that the natural formulation of bisimulation makes use of substitutive effects (or in psi-calculi, the similar assertion extensions) which are bisimulation requirements on neither predicates nor actions. In order to map them into our framework these would need to be cast as actions. This could be an interesting area of further research.

8 Related work

The first published HML is by Hennessy and Milner (1980–1985) [13,21,14]. They work with image-finite CCS processes, where finite (binary) conjunction suffices for adequacy, and define both strong and weak versions of the logic. Milner et al. (1993) [22] give a strong HML for the pi-calculus.

Kozen's modal μ-calculus (1983) [18] subsumes several other weak temporal logics including CTL* (Cranen et al. 2011) [9], and can encode weak transitions using least fixed points. Dam (1996) [10] gives a modal μ-calculus for the pi-calculus, treating bound names using abstractions and concretions, and provides a model checking algorithm. Bradford and Stevens (1999) [5] give a generic framework for parameterising the μ-calculus on data environments, state predicates, and action expressions. The logic defined in the present paper can encode the weakest fixpoint operator of μ-calculi by a disjunction of its finite unrollings, in the same way as the strong version of our logic [23].

There are several weak HMLs for variants of the pi-calculus. Hüttel and Pedersen (2007) [15] define a weak HML for an applied pi-calculus with a subterm-convergent rewrite system augmented with test rules. Koutavas and Hennessey (2012) [17] give a weak HML for a higher-order pi-calculus with both higher-order and first-order communication using an environment-sensitive LTS. The conjunction operator of the logic is infinite, without an explicit bound on its cardinality. Without such a bound the set of formulas is not well-defined: let $\mathcal{F}$ be the set of all formulas, and consider the subset of formulas $\mathcal{S} := \{\bigwedge_{A\in I} A \mid I \subseteq \mathcal{F}\}$. By Cantor's Theorem $|\mathcal{S}| > |\mathcal{F}|$, which is a contradiction. Xu and Long (2015) [27] define a weak HML with countable conjunction for a purely higher-order pi-calculus. The adequacy proof uses stratification.

There are several extensions of HML with spatial modalities. The one most closely related to our logic is by Berger et al. (2008) [3]. They define an HML with both strong and weak action modalities, fixpoints, spatial conjunction and adjunction, and a scope extrusion modality, to study a typed value-passing pi-calculus with selection and recursion. The logic has three (may, must, and mixed) proof systems that are sound and relatively complete.

9 Conclusion

Nominal transition systems include both labelled transitions and state predicates, and can therefore accommodate a wide variety of formalisms. We have defined weak bisimulation and a corresponding weak modal logic on nominal

transition systems, and proved the adequacy result: logical equivalence coincides with weak bisimilarity. The use of finitely supported infinite conjunctions is critical for this result.

A key insight is the notion of weak static implication: to bisimulate a state satisfying a state predicate it must be possible to take zero or more unobservable transitions to reach a state that *both* satisfies the predicate *and* continues to bisimulate. Another important conclusion is that in the logic, state predicates must be guarded by a weak action and cannot directly be combined conjunctively or negated. They may be combined disjunctively, but doing so does not increase expressiveness, since the action modality distributes over disjunction.

Many formalisms, among them most process algebras, feature labelled transitions but no state predicates. It is a folklore fact that this entails no loss of expressiveness. Here we formulate this as a theorem, showing that checking a predicate corresponds to executing a transition leading back to the same state. Formally this is done through a transformation that replaces predicates with loops, and showing that weak bisimilarity is precisely preserved. We also show how the so obtained weak modal logic correlates with the original one.

Nominal transition systems constitute a possible semantics for many formalisms, and an interesting idea for further work is to explore operators on them. For instance, a parallel composition operator would enable closer relations to existing process algebras. There are many different ways to approach this, and to gain general results it would be interesting to define classes of operators, for example through general formats, and explore their properties. There is a huge literature on operator formats for process algebras, of which a few are on nominal process algebras [12,8], but as we understand it none yet treat nominal transition systems in their full generality.

References

1. Martín Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In *Proceedings of POPL 2001*, pages 104–115. ACM, 2001.
2. Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, 1999.
3. Martin Berger, Kohei Honda, and Nobuko Yoshida. Completeness and logical full abstraction in modal logics for typed mobile processes. In *Proceedings of ICALP 2008*, volume 5126 of *LNCS*, pages 99–111. Springer, 2008.
4. Michele Boreale, Rocco De Nicola, and Rosario Pugliese. Proof techniques for cryptographic processes. *SIAM Journal on Computing*, 31(3):947–986, 2001.
5. Julian C. Bradfield and Perdita Stevens. Observational mu-calculus. Technical Report RS-99-5, BRICS, 1999.
6. Stephen D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, 1984.
7. Maria Grazia Buscemi and Ugo Montanari. CC-pi: A constraint-based language for specifying service level agreements. In *Proceedings of ESOP 2007*, volume 4421 of *LNCS*, pages 18–32, 2007.
8. Matteo Cimini, Mohammad R. Mousavi, Michel A. Reniers, and Murdoch J. Gabbay. Nominal SOS. *ENTCS*, 286:103–116, 2012.

9. Sjoerd Cranen, Jan Friso Groote, and Michel Reniers. A linear translation from CTL* to the first-order modal μ-calculus. *Theoretical Computer Science*, 412(28):3129 – 3139, 2011.
10. Mads Dam. Model checking mobile processes. *Information and Computation*, 129(1):35 – 51, 1996.
11. Rocco De Nicola and Michele Loreti. Multiple-labelled transition systems for nominal calculi and their logics. *Mathematical Structures in Computer Science*, 18(1):107–143, 2008.
12. Marcelo Fiore and Sam Staton. A congruence rule format for name-passing process calculi. *Information and Computation*, 207(2):209 – 236, 2009.
13. Matthew Hennessy and Robin Milner. On observing nondeterminism and concurrency. In *Proceedings of ICALP 1980*, volume 85 of *LNCS*, pages 299–309, 1980.
14. Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.
15. Hans Hüttel and Michael D. Pedersen. A logical characterisation of static equivalence. *ENTCS*, 173:139–157, 2007. Proceedings of MFPS XXIII.
16. Magnus Johansson, Jesper Bengtson, Joachim Parrow, and Björn Victor. Weak equivalences in psi-calculi. In *Proceedings of LICS 2010*, pages 322–331, 2010.
17. Vasileios Koutavas and Matthew Hennessy. First-order reasoning for higher-order concurrency. *Computer Languages, Systems & Structures*, 38(3):242–277, 2012.
18. Dexter Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27(3):333 – 354, 1983.
19. Leslie Lamport. What good is temporal logic? In *IFIP Congress*, pages 657–668, 1983.
20. Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer, 1980.
21. Robin Milner. A modal characterisation of observable machine-behaviour. In *Proceedings of CAAP 1981*, volume 112 of *LNCS*, pages 25–34, 1981.
22. Robin Milner, Joachim Parrow, and David Walker. Modal logics for mobile processes. *Theoretical Computer Science*, 114(1):149 – 171, 1993.
23. Joachim Parrow, Johannes Borgström, Lars-Henrik Eriksson, Ramunas Gutkovas, and Tjark Weber. Modal logics for nominal transition systems. In *Proceedings of CONCUR 2015*, volume 42 of *LIPIcs*, pages 198–211. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.
24. Andrew M. Pitts. *Nominal Sets*. Cambridge University Press, 2013.
25. Tjark Weber, Lars-Henrik Eriksson, Joachim Parrow, Johannes Borgström, and Ramunas Gutkovas. Modal logics for nominal transition systems. *Archive of Formal Proofs*, October 2016. `http://isa-afp.org/entries/Modal_Logics_for_NTS.shtml`, Formal proof development.
26. Lucian Wischik and Philippa Gardner. Explicit fusions. *Theoretical Computer Science*, 340(3):606–630, 2005.
27. Xian Xu and Huan Long. A logical characterization for linear higher-order processes. *Journal of Shanghai Jiaotong University (Science)*, 20(2):185–194, 2015.