

POSITIVE HENNESSY-MILNER LOGIC FOR BRANCHING BISIMULATION

HERMAN GEUVERS ^a AND KOMI GOLOV ^b

^a Radboud University Nijmegen, The Netherlands
e-mail address: herman@cs.ru.nl

^b JetBrains, The Netherlands
e-mail address: komi.golov@jetbrains.com

ABSTRACT. Labelled transition systems can be studied in terms of modal logic and in terms of bisimulation. These two notions are connected by Hennessy-Milner theorems, that show that two states are bisimilar precisely when they satisfy the same modal logic formulas. Recently, *apartness* has been studied as a dual to bisimulation, which also gives rise to a dual version of the Hennessy-Milner theorem: two states are apart precisely when there is a modal formula that distinguishes them.

In this paper, we introduce “directed” versions of Hennessy-Milner theorems that characterize when the theory of one state is included in the other. For this we introduce “positive modal logics” that only allow a limited use of negation. Furthermore, we introduce directed notions of bisimulation and apartness, and then show that, for this positive modal logic, the theory of s is included in the theory of t precisely when s is directed bisimilar to t . Or, in terms of apartness, we show that s is directed apart from t precisely when the theory of s is not included in the theory of t . From the directed version of the Hennessy-Milner theorem, the original result follows.

In particular, we study the case of branching bisimulation and Hennessy-Milner Logic with Until (HMLU) as a modal logic. We introduce “directed branching bisimulation” (and directed branching apartness) and “Positive Hennessy-Milner Logic with Until” (PHMLU) and we show the directed version of the Hennessy-Milner theorems. In the process, we show that every HMLU formula is equivalent to a Boolean combination of Positive HMLU formulas, which is a very non-trivial result. This gives rise to a sublogic of HMLU that is equally expressive but easier to reason about.

Concurrent systems are usually modeled as labelled transition systems (LTS), where a labelled transition step takes one from one state to the other in a non-deterministic way. Two states in such a system are considered equivalent in case the same labelled transitions can be taken to equivalent states. This is captured via the notion of bisimulation, which is a type of observable equality: two states are bisimilar if we can make the same observations on them, where the observations are the labelled transitions to equivalent states.

Often it is the case that some transitions cannot be observed and then they are modeled via a silent step, also called a τ -step, denoted by $t \rightarrow_\tau s$. The notion of bisimulation should take this into account and abstract away from the silent steps. This leads to a notion of weak bisimulation (as opposed to strong bisimulation, where all steps are observable). In [vGW89]

Key words and phrases: bisimulation apartness concurrency.

(and the later full version [vGW96]) it has been noticed that weak bisimulation ignores the branching in an LTS that can be caused by silent steps, and that is undesirable. Therefore the notion of branching bisimulation has been developed, which ignores the silent steps while taking into account their branching behavior. Branching bisimulation has been studied a lot as it is the finest equivalence in the well-known “van Glabbeek spectrum” [vG93]. Various algorithms have been developed for checking branching bisimulation and various tools have been developed that verify branching bisimulation for large systems [GV90, JGKW20].

There is a well-known connection between notions of bisimulation and various modal logics, first discovered by Hennessy and Milner [HM85] for strong bisimulation, and then extended to weak and branching bisimulation by De Nicola and Vaandrager [dNV95]. Let $\leftrightarrow$ be a bisimilarity relation, and let $\text{Th}(s)$ be the set of formulas in some modal logic that are true in a state s . A Hennessy-Milner theorem is a theorem relating the bisimulation and the logic by showing that two states are bisimilar precisely when their theories coincide:

$$s \leftrightarrow t \iff \text{Th}(s) = \text{Th}(t).$$

Hennessy and Milner proved such a theorem for Hennessy-Milner Logic and strong bisimulation, while De Nicola and Vaandrager extended this to Hennessy-Milner Logic¹ and weak bisimulation, and Hennessy-Milner Logic with Until and branching bisimulation.

In this paper, we explore what a directed variant of such a theorem would look like by considering what form of “directed bisimulation” relation corresponds to inclusion of theories. Hennessy-Milner logics typically have negation, so if the theory of one state is included in the other, then they are equal. As a consequence, Hennessy-Milner logics typically do not permit non-trivial inclusions of theories, so we must first replace the logic with a positive version. Writing $\Rightarrow$ for our new relation and $\text{Th}_P(s)$ for the theory of s in this new, positive, logic, we prove the property

$$s \Rightarrow t \iff \text{Th}_P(s) \subseteq \text{Th}_P(t). \quad (0.1)$$

Our constructions ensure that $s \leftrightarrow t$ precisely when $s \Rightarrow t$ and $t \Rightarrow s$, and that $\text{Th}_P(s) = \text{Th}_P(t)$ precisely when $\text{Th}(s) = \text{Th}(t)$. The undirected Hennessy-Milner theorem is thus an immediate consequence of our directed variant.

Our proof of Equation (0.1) uses the notion of apartness, which is the dual notion of bisimulation, studied previously by Geuvers and Jacobs [GJ21]. Two states are bisimilar precisely when they are not apart. The property we prove is thus

$$s \not\# t \iff \exists \varphi. \varphi \in \text{Th}_P(s) - \text{Th}_P(t), \quad (0.2)$$

where φ is called the distinguishing formula.

Unlike bisimulation, apartness is an inductive notion, meaning that we can perform our proofs by an induction on the derivation of an apartness. It turns out that the structure of this derivation closely mirrors the distinguishing formula, making the proof straightforward. This was previously shown by Geuvers in relation to strong and weak bisimulation in [Geu22], but a straightforward inductive proof does not work in the branching setting when the logic includes an *until* operator.

In this paper we study the directed constructions for branching bisimulation. Concretely, we define and study (Section 2) a logic of Positive Hennessy-Milner Logic with Until formulas which we refer to as PHMLU (Subsection 2.1), as well as a directed branching bisimulation relation $\Rightarrow_{db}$ (Subsection 2.2) and a directed branching apartness relation $\not\#_{db}$

¹This involved modifying the semantics to account for silent steps.

(Subsection 2.3). We then show that Equation (0.2) (for PHMLU and branching apartness) holds, and from that we conclude that Equation (0.1) holds as well. There are differences between HMLU and PHMLU that make it far from obvious that the logics are equally powerful. We show that the logics are equivalent in which states they distinguish, and moreover we show that every HMLU formula is equivalent to a Boolean combination of PHMLU formulas (Subsection 2.1). This gives rise to a logic that is equally expressive as HMLU, but for which the satisfaction relation is simpler.

To introduce the directed versions of bisimulation and apartness, and the corresponding positive modal logics, we will first start from the simpler setting of strong bisimulation and standard Hennessy-Milner Logic in Section 1. Here the construction and the proofs are rather straightforward, but it also suggests that a ‘directed approach’ to bisimulation, combined with a positive version of Hennessy-Milner logic and the use of apartness, should be applicable to other notions of bisimulation. For weak bisimulation, this is the case, which we have verified, but not included in the present paper.

Throughout this paper, we will work in a fixed labelled transition system (LTS) $(X, A, \rightarrow)$, consisting of a set of *states* X , a set of *actions* A , possibly with a *silent action* $\tau \notin A$, and a transition relation $\rightarrow_a$ for every $a \in A$ (and for τ , if applicable). Following [GJ21], we will use α to denote an arbitrary action from $A \cup \{\tau\}$, a to denote an arbitrary non-silent action (so $a \in A$), and use $\rightarrow_\alpha$ to denote the transitive reflexive closure of $\rightarrow_\alpha$. We use $\rightarrow_{(\alpha)}$ as a shorthand for $\rightarrow_a$ if $\alpha = a \neq \tau$, and for the reflexive closure of $\rightarrow_\tau$ if $\alpha = \tau$.

We would like to thank Jan Friso Groote and Jurriaan Rot for discussions on the topic of this paper. We also thank the referees, in particular also the referees of earlier versions of this paper, for their useful comments.

1. HENNESSY-MILNER LOGIC AND STRONG BISIMULATION

Definition 1.1. A *labelled transition system* (LTS) is a tuple $(X, A, \rightarrow)$, where X is a set of *states*, A is a set of *actions* and for every $a \in A$, $\rightarrow_a$ is a binary relation on X . When $s \rightarrow_a t$ holds, we say that s *a-steps to* t .

An LTS is *image-finite* if for all $s \in X$ and $a \in A$, the set $\{t \in X \mid s \rightarrow_a t\}$ is finite.

We will be particularly interested in image-finite LTSs, since the logics we study are finitary. We can characterize states in an LTS using Hennessy-Milner Logic (HML).

Definition 1.2. The formulas of HML are given inductively by the following definition:

$$\varphi := \top \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle a \rangle \varphi.$$

We use $\perp$ as a shorthand for $\neg\top$ and $\varphi_1 \vee \varphi_2$ as a shorthand for $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$. The modality binds weakly; $\langle a \rangle \varphi \wedge \psi$ denotes $\langle a \rangle (\varphi \wedge \psi)$.

Definition 1.3. The relation $s \models \varphi$ is defined inductively by the following rules:

$$\begin{array}{lll} s \models \top & \text{always} & s \models \varphi_1 \wedge \varphi_2 \iff s \models \varphi_1 \text{ and } s \models \varphi_2 \\ s \models \neg\varphi & \iff s \not\models \varphi & s \models \langle a \rangle \varphi \iff \exists s'. s \rightarrow_a s' \wedge s' \models \varphi. \end{array}$$

Given a state s we define $\text{Th}(s) := \{\varphi \in \text{HML} \mid s \models \varphi\}$, the set of HML formulas true in s . We call a formula in $\text{Th}(s) - \text{Th}(t)$ or $\text{Th}(t) - \text{Th}(s)$ a *distinguishing formula* for s and t .

To compare two states, one considers if they can recursively simulate one another. This notion is known as bisimulation.

Definition 1.4. A symmetric relation R is a strong bisimulation if whenever $R(s, t)$ and $s \rightarrow_a s'$, there exists some t' such that $R(s', t')$ and $t \rightarrow_a t'$.

We say two states are strongly bisimilar (notation $s \leftrightarrow_s t$) if there exists a strong bisimulation that relates them.

A key result of Hennessy and Milner [HM85] is that these two views of labelled transition systems are related in the following sense:

Theorem 1.5 [HM85]. *In an image-finite LTS, two states s and t are strongly bisimilar precisely when they satisfy the same HML formulas; that is,*

$$s \leftrightarrow_s t \Leftrightarrow \text{Th}(s) = \text{Th}(t).$$

We are interested in a similar theorem that considers inclusion of theories rather than equality. However, in the context of Hennessy-Milner Logic such a theorem would not be meaningfully different, since the presence of negation gives us the following result:

Proposition 1.6. *If $\text{Th}(s) \subseteq \text{Th}(t)$, then $\text{Th}(s) = \text{Th}(t)$.*

Proof. Suppose, on the contrary, that there is some $\varphi \in \text{Th}(t) - \text{Th}(s)$. Then $\neg\varphi \in \text{Th}(s) \subseteq \text{Th}(t)$, hence $t \models \varphi$ and $t \models \neg\varphi$, a contradiction. $\square$

We will thus introduce Positive Hennessy-Milner Logic (PHML), where negation is only permitted under a modality.²

Definition 1.7. The formulas of PHML are given inductively by the following definition:

$$\varphi := \top \mid \perp \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \langle a \rangle(\varphi_1 \wedge \neg\varphi_2).$$

We view a formula of PHML as also being a formula of HML, and can thus reuse the semantics from Definition 1.3. We define $\text{Th}_P(s) := \{\varphi \in \text{PHML} \mid s \models \varphi\}$. We call a $\varphi \in \text{Th}_P(s) - \text{Th}_P(t)$ a *positive distinguishing formula* for s and t .

There are various syntactic classes related to PHML that will be useful for our proofs, and for these we introduce some special notation.

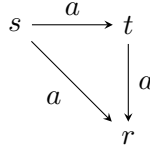
Notation 1.8.

- $\pm\text{PHML}$ denotes the set of PHML formulas and their negations.
- $\bigwedge \pm\text{PHML}$ denotes the set of conjunctions of $\pm\text{PHML}$ formulas.
- Given a formula $\varphi \in \bigwedge \pm\text{PHML}$, we will use φ^+ and φ^- to refer to the PHML formulas such that φ is equivalent to $\varphi^+ \wedge \neg\varphi^-$.

So φ^+ is the conjunction of all ‘positive’ conjuncts of φ , the *positive segment* of φ , and φ^- is the disjunction of ψ for every ‘negative’ conjunct $\neg\psi$ of φ , the *negative segment* of φ . Concretely, for $\varphi \in \bigwedge \pm\text{PHML}$ with $\varphi = \varphi_1 \wedge \varphi_2 \wedge \neg\varphi_3 \wedge \neg\varphi_4$, we have $\varphi^+ = \varphi_1 \wedge \varphi_2$ and $\varphi^- = \varphi_3 \vee \varphi_4$, and so φ is equivalent to $\varphi^+ \wedge \neg\varphi^-$. We write $\langle a \rangle\varphi$ for $\langle a \rangle\varphi^+ \wedge \neg\varphi^-$. Recall that by our conventions, this is read as $\langle a \rangle(\varphi^+ \wedge \neg\varphi^-)$.

Example 1.9. If we consider PHML, we can have non-trivial inclusion of theories. This is illustrated by the following LTS. (The fact the inclusions hold in the first place is most easily shown by constructing a strong directed bisimulation as defined below. However, the rigorous reader can perform a proof by induction.)

²Prohibiting negation entirely would produce a logic that is too weak; it would correspond to simulation. Two states simulating each other is weaker than two states being bisimilar, while we require that two states directed bisimulating each other be equivalent to the states being bisimilar.



In this LTS, we have $\text{Th}_P(r) \subsetneq \text{Th}_P(t) \subsetneq \text{Th}_P(s)$. Examples of PHML formulas that distinguish s , t and r are the following.

- $\langle a \rangle \langle a \rangle \top$, which holds in s but not in t and r .
- $\langle a \rangle \neg \langle a \rangle \top$ and $\langle a \rangle \top$, which hold in s and t , but not in r

Note that $\langle a \rangle \perp$ does not hold in any state, but for different reasons: in r there is no a -step possible, whereas in s and t there is an a -step possible, but not to a state where $\perp$ holds (because $\perp$ never holds).

Theorem 1.10. *Every HML formula φ is equivalent to a disjunction of $\bigwedge \pm\text{PHML}$ formulas.*

Proof. The proof proceeds by induction on φ . We treat the cases for φ being a negation and for φ being a modality.

Suppose that $\varphi = \neg\psi$, then by induction there is a formula $\bigvee_{i=1}^n \Psi_i$ equivalent to ψ and we write $\Psi_i = \Psi_i^+ \wedge \neg\Psi_i^-$ (for $i \in \{1, \dots, n\}$). Then $\varphi = \neg\psi$ is equivalent to $\bigwedge_{i=1}^n \neg(\Psi_i^+ \vee \Psi_i^-)$. Note that $\Psi_i^- \in \text{PHML}$, but $\neg(\Psi_i^+)$ is a negation of a PHML-formula. We can let $\wedge$ distribute over $\vee$ and we find that φ is equivalent to a disjunction of $\bigwedge \pm\text{PHML}$ formulas.

Suppose that $\varphi = \langle a \rangle \psi$, then by induction there is a formula $\bigvee_{i \in I} \Psi_i$ equivalent to ψ . Since modality distributes over disjunction, $\langle a \rangle \bigvee_i \Psi_i$ is equivalent to $\bigvee_i \langle a \rangle \Psi_i$, which is the desired formula. $\square$

Corollary 1.11. *For all states s, t , we have $\text{Th}(s) = \text{Th}(t)$ precisely when $\text{Th}_P(s) = \text{Th}_P(t)$.*

Proof. The left to right direction is easy: $\text{Th}_P(s) = \text{Th}(s) \cap \text{PHML}$, so $\text{Th}(s) = \text{Th}(t)$ implies $\text{Th}_P(s) = \text{Th}_P(t)$.

In the other direction, suppose $\text{Th}_P(s) = \text{Th}_P(t)$ and $\varphi \in \text{Th}(s)$. By Theorem 1.10, there is an equivalent formula $\bigvee_i \Psi_i$ where $\Psi_i \in \bigwedge \pm\text{PHML}$. As $s \models \varphi$, there is some i such that $s \models \Psi_i$. Writing $\Psi_i = \bigwedge_j \Psi_j^+ \wedge \neg\Psi_j^-$, we have that for all j , $s \models \Psi_j^+$ and $s \not\models \Psi_j^-$. Since each Ψ_j^+ and Ψ_j^- is positive, it follows that $t \models \Psi_j^+$ and $t \not\models \Psi_j^-$ and thus $t \models \Psi_i$ and $t \models \varphi$, as required. $\square$

This concludes our construction on the logic side. Let us now consider the bisimulation side of the question.

Definition 1.12. A relation R is a *directed strong bisimulation* if whenever $R(s, t)$ and $s \rightarrow_a s'$, there exists some t' such that $t \rightarrow_a t'$ and $R(s', t')$ and $R(t', s')$.

We say two states are *directed strongly bisimilar* (notation $s \rightrightarrows_{ds} t$) if there exists a directed strong bisimulation that relates them.

Theorem 1.13. *Two states are strongly bisimilar if and only if they are directed strongly bisimilar in each direction.*

Proof. The left-to-right direction follows from the fact that every strong bisimulation is a directed strong bisimulation. For the other direction, define $R(p, q) = p \rightrightarrows_{ds} q \wedge q \rightrightarrows_{ds} p$. This is a strong bisimulation, and hence $R(p, q)$ implies that p and q are strongly bisimilar. $\square$

We can now state our directed Hennessy-Milner theorem for strong bisimulation.

Theorem 1.14. *In an image-finite LTS, for all s, t , $s \rightarrow_{ds} t$ iff $\text{Th}_P(s) \subseteq \text{Th}_P(t)$.*

In order to prove this theorem we introduce one more directed notion: directed strong apartness, the dual of directed strong bisimulation. This will simplify the proof to an inductive argument in each direction, one induction on the formula that distinguishes the states, and the other induction on the derivation of the apartness.

Definition 1.15. A relation Q is a *directed strong apartness* if the following rule holds for Q .

$$\frac{s \rightarrow_a s' \quad \forall t'. t \rightarrow_a t' \implies Q(s', t') \vee Q(t', s')}{Q(s, t)} \text{IN}_{ds}$$

Two states s, t are *directed strong apart*, notation $s \not\rightarrow_{ds} t$, if for every directed strong apartness Q , it holds that $Q(s, t)$.

We want to stress that directed strong apartness is an *inductive notion*, like other notions of apartness, see [GJ21]. This implies that it can be characterized via a *derivation system*: $s \not\rightarrow_{ds} t$ if and only if this can be derived with the derivation rule IN_{ds} . It is also immediate that directed strong apartness is the complement of directed strong bisimulation. We collect these facts in the following easy to prove lemma.

Lemma 1.16. *We have that $s \not\rightarrow_{ds} t$ if and only if this can be derived using the following rule.*

$$\frac{s \rightarrow_a s' \quad \forall t'. t \rightarrow_a t' \implies s' \not\rightarrow_{ds} t' \vee t' \not\rightarrow_{ds} s'}{s \not\rightarrow_{ds} t} \text{IN}_{ds}$$

Furthermore, $s \not\rightarrow_{ds} t \iff \neg(s \rightarrow_{ds} t)$.

Proof. Phrased differently: if we redefine $s \not\rightarrow_{ds} t$ as “there is a derivation of $s \not\rightarrow_{ds} t$ using only rule IN_{ds} ”, then we can show that $s \not\rightarrow_{ds} t$ if and only if $Q(s, t)$ holds for all apartness relations Q .

The first observation is that the redefined $s \not\rightarrow_{ds} t$ is itself an apartness relation. Hence, if $Q(s, t)$ for all apartness relations Q , then $s \not\rightarrow_{ds} t$. The other way around, if $s \not\rightarrow_{ds} t$ by a derivation, and Q is an apartness relation, then we can prove $Q(s, t)$ by induction on the derivation.

For the second part: For $\Rightarrow$ we derive a contradiction from $s \not\rightarrow_{ds} t$ and $s \rightarrow_{ds} t$ by induction on the derivation of $s \not\rightarrow_{ds} t$. For $\Leftarrow$, we show that $\neg(s \not\rightarrow_{ds} t)$ is a directed bisimulation relation, so $\neg(s \not\rightarrow_{ds} t)$ implies $s \rightarrow_{ds} t$. $\square$

One may wonder what the “base case” is for the inductive definition of $s \not\rightarrow_{ds} t$, because every application of rule IN_{ds} seems to have an apartness hypothesis again. In case $s \rightarrow_a s'$ and there is no a -step from t , then the $\forall t'. t \rightarrow_a t' \implies \dots$ trivially holds, and we find that $s \not\rightarrow_{ds} t$.

Every two states are either directed strong bisimilar or directed strong apart, depending on whether there is any directed strong bisimulation that relates them. We can thus restate Theorem 1.14 as follows.

Theorem 1.17. *In an image-finite LTS, for all s, t , $s \not\rightarrow_{ds} t$ iff there is some positive distinguishing formula for s and t .*

Proof. Both directions of the proof proceed by induction: from left to right by induction on the derivation of apartness, and from right to left by induction on the positive distinguishing formula.

For the left to right case, $s \not\#_{ds} t$ has been concluded by the IN_{ds} rule (see Lemma 1.16), so say we have $s \rightarrow_a s'$ and we know $s' \not\#_{ds} t' \vee t' \not\#_{ds} s'$ for all (finitely many!) t' for which $t \rightarrow_a t'$. By induction we have, for each of these t' , either a formula $\varphi' \in \text{Th}_P(s') - \text{Th}_P(t')$ or a formula $\psi' \in \text{Th}_P(t') - \text{Th}_P(s')$. Taking Φ to be the conjunction of all φ' and Ψ to be the disjunction of all ψ' , we find that $\varphi := \langle a \rangle (\Phi \wedge \neg \Psi)$ is in $\text{Th}_P(s) - \text{Th}_P(t)$.

For the right to left case, we do induction on $\varphi \in \text{Th}_P(s) - \text{Th}_P(t)$. The interesting case is when φ is of the form $\langle a \rangle (\varphi_1 \wedge \neg \varphi_2)$. Then $s \rightarrow_a s'$ with $s' \models \varphi_1 \wedge \neg \varphi_2$ for some s' . Also, there is no t' with $t \rightarrow_a t'$ and $t' \models \varphi_1 \wedge \neg \varphi_2$. So for all t' with $t \rightarrow_a t'$ we either have $t' \not\models \varphi_1$ or $t' \models \varphi_2$. By induction we find that $\forall t'. t \rightarrow_a t' \implies s' \not\#_{ds} t' \vee t' \not\#_{ds} s'$, and we can conclude $s \not\#_{ds} t$ by the rule IN_{ds} . $\square$

2. HENNESSY-MILNER LOGIC WITH UNTIL AND BRANCHING BISIMULATION

Let us now extend our notion of a labelled transition system to permit silent steps. We extend our system with a special ‘silent’ action $\tau \notin A$ with a corresponding silent step relation $\rightarrow_\tau$.

Definition 2.1. A *labelled transition with silent steps* (LTS_τ) is a labelled transition system $(X, A, \rightarrow)$ with a special action $\tau \notin A$ and a corresponding relation $\rightarrow_\tau$ on X .

Our presentation of branching bisimulation follows the style of Basten [Bas96] and van Glabbeek et al. [vGLT09], which has also been described already in [vG93]. We use the abbreviation $s \rightarrow_{(\alpha)} t$ as a shorthand for $s \rightarrow_\alpha t \vee (s = t \wedge \alpha = \tau)$.

We will refer to a sequence $s \twoheadrightarrow_\tau s' \rightarrow_{(\alpha)} s''$ as an *eventual α -step*, where the $\rightarrow_{(\alpha)}$ indicates that for $\alpha = \tau$ the last step is optional.

We adapt our notion of image-finiteness to this new setting as follows, effectively requiring that for every $\alpha \in A \cup \{\tau\}$, every state only has finitely many outgoing eventual α -steps.

Definition 2.2. An $\text{LTS}_\tau (X, A, \rightarrow)$ is *image-finite* if the underlying LTS is image-finite and moreover the image of the transitive closure of $\rightarrow_\tau$ is finite.

The Hennessy-Milner theorem for branching bisimulation and modal logic was found by De Nicola and Vaandrager in [dNV95]. The modal logic in question is Hennessy-Milner Logic with Until, where a diamond formula can impose a requirement not only on the state after the step, but also on all states on the path leading up to this state. We now briefly recap this logic, using the semantics introduced by van Glabbeek [vG93].

Definition 2.3. Formulas of Hennessy-Milner Logic with Until (HMLU) are given by the following inductive definition

$$\varphi := \top \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \langle \alpha \rangle \varphi_2.$$

We will use $\varphi_1 \vee \varphi_2$ as a shorthand for $\neg(\neg \varphi_1 \wedge \neg \varphi_2)$ and $\langle \alpha \rangle \varphi$ as a shorthand for $\top \langle \alpha \rangle \varphi$. The modality binds tightly on the left and loosely on the right, so $\delta \wedge \gamma \langle \alpha \rangle \varphi \wedge \psi$ reads as $\delta \wedge (\gamma \langle \alpha \rangle (\varphi \wedge \psi))$, and $\delta \langle \alpha \rangle \varphi \langle \beta \rangle \psi$ as $\delta \langle \alpha \rangle (\varphi \langle \beta \rangle \psi)$.

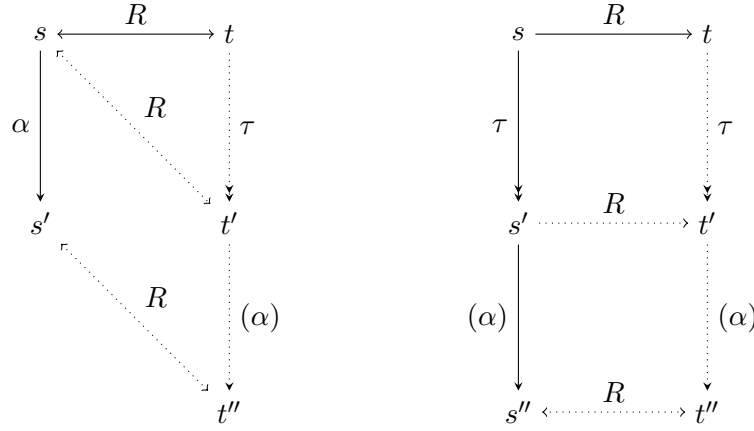


Figure 1: Branching Bisimulation (left), Directed Branching Bisimulation (right)

The semantics of a formula φ in a state s of an LTS are as follows. We write $s \models \varphi$ to mean φ holds in s , and we write $s \xrightarrow{\varphi}_{\alpha} s'$ to indicate that there exists a sequence $s = s_1 \rightarrow_{\alpha} \dots \rightarrow_{\alpha} s_n = s'$ such that for all $i \leq n$, $s_i \models \varphi$.

Definition 2.4. The relation $s \models \varphi$ is defined inductively by the following rules:

$$\begin{array}{ll} s \models \top & \Leftrightarrow \text{always} \\ s \models \neg\varphi & \Leftrightarrow s \not\models \varphi \\ s \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow s \models \varphi_1 \text{ and } s \models \varphi_2 \\ s \models \varphi_1 \langle \alpha \rangle \varphi_2 & \Leftrightarrow \exists s', s''. s \xrightarrow{\varphi_1}_{\tau} s' \rightarrow_{(\alpha)} s'' \wedge s'' \models \varphi_2. \end{array}$$

Given a state s we define the *theory* of s , $\text{Th}(s)$ as $\{\varphi \in \text{HMLU} \mid s \models \varphi\}$. We say that φ is a *distinguishing formula* for s and t in case $\varphi \in \text{Th}(s) - \text{Th}(t)$ or $\varphi \in \text{Th}(t) - \text{Th}(s)$.

Our choice to require $s' \rightarrow_{(\alpha)} s''$ in the case of a formula with a modality follows van Glabbeek et al. [vGLT09] (and has also already been considered in [vG93]). It deviates from the definition by Vaandrager and de Nicola [dNV95], where a special case $\alpha = \tau$ is considered. It can be shown that both interpretations of the modality generate the same logical equivalence of states (where two states s and t are logical equivalent if they have the same theory, $\text{Th}(s) = \text{Th}(t)$).

Let us turn our attention to the semantic side of the question.

Definition 2.5. A symmetric relation R is a *branching bisimulation* if whenever $R(s, t)$ and $s \rightarrow_{\alpha} s'$, there exists an eventual α -step $t \xrightarrow{\tau} t' \rightarrow_{(\alpha)} t''$ such that $R(s, t')$ and $R(s', t'')$. Two states s and t are branching bisimilar (notation $s \xleftrightarrow{b} t$) if they are related by some branching bisimulation.

The situation is illustrated on the left in Figure 1.

In other words, a branching bisimulation is a relation that transfers α -steps into eventual α -steps. In this definition we again follow Basten [Bas96] and van Glabbeek et al. [vGLT09]. Basten calls these relations a “semi-branching bisimulation” and shows that the notion of being “semi-branching bisimilar” is equivalent to the original notion of branching bisimilar that can be found in the work of De Nicola and Vaandrager [dNV95] and van Glabbeek [vG93]. An important advantage of the notion of branching bisimulation of Definition 2.5 is that the composition of two branching bisimulations is again a branching bisimulation.

De Nicola and Vaandrager [dNV95] have shown that branching bisimulation and HMLU are related via a Hennessy-Milner theorem. This theorem also applies to the notion of

branching bisimulation of Definition 2.5 and the interpretation of HMLU formulas of Definition 2.4.

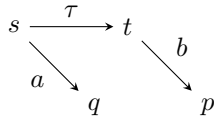
Theorem 2.6 [dNV95]. $s \leftrightarrow_b t$ precisely when $\text{Th}(s) = \text{Th}(t)$.

2.1. Positive Hennessy-Milner Logic with Until. As was the case with Hennessy-Milner Logic, Hennessy-Milner Logic with Until needs to be adapted to a positive variant to allow for non-trivial inclusions of theories. Unlike HML, however, simply requiring that all negations be under a modality is insufficient for our purposes. Instead, we require that all negations be on the right-hand side of a modality. The motivation for this is as follows.

A positive formula expresses what eventual α -steps can be taken from some state s . If s is a state that contains a silent step to t , then every eventual α -step from t can also be taken from s . Hence any positive formula that holds in t should also hold in s ; that is, for any positive formula φ , we expect the following implication to hold:

$$s \rightarrow_\tau t \text{ and } t \models \varphi \quad \Rightarrow \quad s \models \varphi. \quad (2.1)$$

Example 2.7. There exist formulas where all negations are under a modality that nevertheless do not satisfy implication (2.1). Consider $\varphi = (\neg\langle a \rangle \top) \langle b \rangle \top$ in the following LTS.



We have $t \models \varphi$ but $s \not\models \varphi$.

To resolve this, we specify that a modality is only a positive formula if the formula on its left is positive, giving us the following definition.

Definition 2.8. We define *positive* and *negative* formulas by induction as follows:

$\text{pos}(\top)$	$\Leftrightarrow$ always	$\text{neg}(\top)$	$\Leftrightarrow$ always
$\text{pos}(\neg\varphi)$	$\Leftrightarrow \text{neg}(\varphi)$	$\text{neg}(\neg\varphi)$	$\Leftrightarrow \text{pos}(\varphi)$
$\text{pos}(\varphi_1 \wedge \varphi_2)$	$\Leftrightarrow \text{pos}(\varphi_1) \text{ and } \text{pos}(\varphi_2)$	$\text{neg}(\varphi_1 \wedge \varphi_2)$	$\Leftrightarrow \text{neg}(\varphi_1) \text{ and } \text{neg}(\varphi_2)$
$\text{pos}(\varphi_1 \langle \alpha \rangle \varphi_2)$	$\Leftrightarrow \text{pos}(\varphi_1)$	$\text{neg}(\varphi_1 \langle \alpha \rangle \varphi_2)$	$\Leftrightarrow \text{never.}$

Note that all modality-free formulas are both positive and negative.

Positive and negative formulas satisfy the kind of transfer property we express in (2.1).

Lemma 2.9. For s, t states with $s \rightarrow_\tau t$ we have

- for every positive formula φ , if $t \models \varphi$ then $s \models \varphi$;
- for every negative formula φ , if $s \models \varphi$ then $t \models \varphi$.

Proof. By induction on the formula. The case when $\varphi = \delta \langle \alpha \rangle \psi$ goes as follows: if $t \models \delta \langle \alpha \rangle \psi$ then there exists some $t \xrightarrow{\delta}_\tau t' \xrightarrow{(\alpha)} t''$. Since δ is positive and $t \models \delta$ we have $s \models \delta$ by induction. The path $s \rightarrow_\tau t \xrightarrow{\delta}_\tau t' \xrightarrow{(\alpha)} t''$ witnesses that $s \models \delta \langle \alpha \rangle \psi$, as desired. $\square$

A class of formulas of some interest is the following:

Definition 2.10. A formula φ is *left-positive* if in all of its subformulas of the form $\delta \langle \alpha \rangle \psi$, δ is positive.

Note that not every left-positive formula is positive, and not every positive formula is left-positive. For example, $\neg(\top\langle\alpha\rangle\top)$ is left-positive, since the formula $\top$ on the left of the modality is positive, but is as a whole not positive. On the other hand, $\langle\alpha\rangle(\neg\langle\gamma\rangle\top)\langle\beta\rangle\top$ is positive, but is not left-positive since $\neg\langle\beta\rangle\top$ appears on the left of a modality but is not positive.³ As we shall see in Corollary 2.16, every HMLU formula is equivalent to a left-positive formula. Our experience is that proofs about left-positive formulas are simpler than about arbitrary formulas. This is largely due to the following simplification of the semantics:

Lemma 2.11. *When restricted to left-positive formulas, the satisfaction relation is equivalent to the one generated by the following inductive definition:*

$$\begin{aligned} s \models \top &\Leftrightarrow \text{always} & s \models \varphi_1 \wedge \varphi_2 &\Leftrightarrow s \models \varphi_1 \text{ and } s \models \varphi_2 \\ s \models \neg\varphi &\Leftrightarrow s \not\models \varphi & s \models \varphi_1\langle\alpha\rangle\varphi_2 &\Leftrightarrow \exists s', s''. s \rightarrow_\tau s' \rightarrow_{(\alpha)} s'' \wedge s' \models \varphi_1 \wedge s'' \models \varphi_2. \end{aligned}$$

Proof. By induction on the formula, using Lemma 2.9. $\square$

In other words, for left-positive formulas the satisfaction relations of the *until* modality and the *just-before* modality coincide. The just-before modality has been studied amongst others by van Glabbeek [vG93], who remarks that both until and just-before give logics equivalent to branching bisimulation. The standard proof, given for example in [vGLT09]⁴, is similar to the one we take here: one defines a class of upwards and downwards formulas and show that formulas of logic with until can be expressed as combinations of those.

We take this approach a step further by defining a logic of positive formulas, giving them a syntactic characterisation. We call this logic Positive Hennessy-Milner Logic with Until, and ensure by construction that all formulas are both positive and left-positive.

Definition 2.12. Formulas of Positive Hennessy-Milner Logic with Until (PHMLU) are given by the following inductive definition

$$\varphi := \top \mid \perp \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1\langle\alpha\rangle(\varphi_2 \wedge \neg\varphi_3).$$

We again abbreviate $\top\langle\alpha\rangle\varphi$ to $\langle\alpha\rangle\varphi$ and define $\text{Th}_P(s) := \{\varphi \in \text{PHMLU} \mid s \models \varphi\}$. We regard formulas of PHMLU as also being formulas of HMLU, allowing us to use the same semantics. By induction, we see that all PHMLU formulas are positive and left-positive; in particular, this means that the transfer property of Lemma 2.9 and the semantics of Lemma 2.11 hold for PHMLU formulas.

Like with PHML (see Notation 1.8), there are two syntactic classes related to PHMLU that will be useful for our proofs, as well as some convenient shorthands.

Notation 2.13.

- $\pm\text{PHMLU}$, the set of PHMLU formulas and their negations.
- $\bigwedge \pm\text{PHMLU}$, the set of conjunctions of $\pm\text{PHMLU}$ formulas.
- Given a formula $\varphi \in \bigwedge \pm\text{PHMLU}$, we again use φ^+ and φ^- to denote the PHMLU formulas such that φ is equivalent to $\varphi^+ \wedge \neg\varphi^-$.
- Given formulas $\delta \in \text{PHMLU}$ and $\varphi \in \bigwedge \pm\text{PHMLU}$, we use $\delta\langle\alpha\rangle\varphi$ to denote $\delta\langle\alpha\rangle\varphi^+ \wedge \neg\varphi^-$.

³Recall our convention that $\langle\alpha\rangle\varphi$ is a shorthand for $\top\langle\alpha\rangle\varphi$, and that modality binds strongly on the left and weakly on the right, so $\delta\langle\alpha\rangle\varphi\langle\beta\rangle\psi$ should be read as $\delta\langle\alpha\rangle(\varphi\langle\beta\rangle\psi)$.

⁴This presentation features an explicit divergence operator, but this does not change the key argument.

At this point, it may be concerning that we have both ‘positive HMLU formulas’ and ‘PHMLU formulas’, where the former is a superset of the latter; for example $\langle a \rangle (\neg(\langle b \rangle \top) \vee \langle a \rangle \top)$ is positive but is not a PHMLU formula. The following theorem shows that when reasoning up to equivalence, this is not a problem, as every positive HMLU formula is equivalent to some PHMLU formula.

Theorem 2.14. *Every HMLU formula φ is equivalent to a disjunction of $\bigwedge \pm$ PHMLU formulas. Moreover, if φ is positive then it is equivalent to a disjunction of PHMLU formulas.*

The full proof can be found in Appendix A. Here, we restrict ourselves to a proof sketch.

Proof sketch. The proof is similar to that of Theorem 1.10. We again show, by induction, that every formula φ is equivalent to a disjunction $\bigvee_{i \in I} \Phi_i$, and only the modality case is of interest.

Let $\delta \langle \alpha \rangle \psi$ be an HMLU formula. By induction, δ is equivalent to $\check{\Delta} = \bigvee_i \Delta_i$ and ψ is equivalent to $\bigvee_j \Psi_j$. Since the modality distributes over disjunction on the right, we can take the same approach as in Theorem 1.10 to show that $\delta \langle \alpha \rangle \psi$ is equivalent to $\bigvee_j (\check{\Delta} \langle \alpha \rangle \Psi_j)$. It remains to show that for every j , the disjunct $\check{\Delta} \langle \alpha \rangle \Psi_j$ is equivalent to a disjunction of $\bigwedge \pm$ PHMLU formulas.

For any state s , if $s \models \check{\Delta} \langle \alpha \rangle \Psi_j$ then there is some path $s \twoheadrightarrow_{\tau}^{\check{\Delta}} s' \rightarrow_{(\alpha)} s''$ with $s'' \models \Psi_j$. By taking the path $s = s_1 \rightarrow_{\tau} s_2 \rightarrow_{\tau} \dots \rightarrow_{\tau} s_n$ and finding the disjunct Δ_{i_k} that holds in state s_k , we can find a formula

$$\Delta_{i_1} \langle \tau \rangle \Delta_{i_2} \langle \tau \rangle \dots \langle \tau \rangle \Delta_{i_n} \langle \alpha \rangle \Psi_j. \quad (2.2)$$

This formula holds in s , and conversely, if any formula of this shape holds in s , then so does $\check{\Delta} \langle \alpha \rangle \Psi_j$. However, we cannot take the disjunction of all such formulas as our solution: these formulas are not necessarily $\bigwedge \pm$ PHMLU, and there may be infinitely many of them.

To remedy this, note that for any $\bigwedge \pm$ PHMLU formula θ , the formula $\Delta_{i_k} \langle \tau \rangle \theta$ is equivalent to $\neg \Delta_{i_k}^- \wedge \Delta_{i_k}^+ \langle \tau \rangle \theta$. This is because by Lemma 2.9, satisfaction of negative formulas is closed under silent steps, so if $\neg \Delta_{i_k}^-$ holds in some state r , it will hold in all states reachable from r by silent steps. The formula (2.2) is thus equivalent to

$$\neg \Delta_{i_1}^- \wedge \Delta_{i_1}^+ \langle \tau \rangle \neg \Delta_{i_2}^- \wedge \Delta_{i_2}^+ \langle \tau \rangle \dots \langle \tau \rangle \neg \Delta_{i_n}^- \wedge \Delta_{i_n}^+ \langle \alpha \rangle \Psi_j.$$

Recall that by our convention, the modality binds tightly on the left and loosely on the right, so this should be read as $\neg \Delta_{i_1}^- \wedge (\Delta_{i_1}^+ \langle \tau \rangle (\dots))$.

Moreover, suppose that for some l, k , $i_l = i_k$. Then in fact $s_{k'} \models \Delta_{i_k}$ for all k' between k and l . Namely, by Lemma 2.9, all the positive formulas of Δ_{i_l} will hold for $s_{k'}$ if $k' \leq l$, while all the negative formulas of Δ_{i_k} will hold for $s_{k'}$ if $k' \geq k$. It follows that a single modality can be used for all the steps from l to k , meaning that the number of extra τ modalities we use in the construction of our formula can be bounded by the number of disjuncts in Δ (in fact, minus one).

Putting this together, we get the following disjunction, where $i_1, \dots, i_n$ ranges over all non-empty sequences of indices in I without doubles.

$$\Phi := \bigvee_{\vec{i}, j} \neg \Delta_{i_1}^- \wedge \Delta_{i_1}^+ \langle \tau \rangle \dots \langle \tau \rangle \neg \Delta_{i_n}^- \wedge \Delta_{i_n}^+ \langle \alpha \rangle \Psi_j.$$

This concludes our sketch of the construction of the equivalent formula. A more explicit construction, including its verification, can be found in Appendix A.

The second part of the theorem, showing that this construction can be done without negations of PHMLU formulas when φ is positive, again goes by induction: in this case $\Delta_{i_1}^- = \perp$, from which it is immediate that each disjunct of Φ is, up to equivalence, a PHMLU formula. $\square$

Corollary 2.15. *Every positive formula has an equivalent PHMLU formula.*

Proof. Immediate: a disjunction of PHMLU formulas is itself a PHMLU formula. $\square$

Corollary 2.16. *Every formula has an equivalent left-positive formula.*

Proof. Immediate: every Boolean combination of PHMLU formulas is left-positive. $\square$

Corollary 2.17. *For two states s, t , $\text{Th}_P(s) = \text{Th}_P(t)$ if and only if $\text{Th}(s) = \text{Th}(t)$.*

Proof. This proof is essentially the same as the proof of Corollary 1.11. $\square$

2.2. Directed branching bisimulation. While branching bisimulation takes α -steps to eventual α -steps, we let directed branching bisimulation take eventual α -steps to eventual α -steps. We drop the symmetry requirement on the relation itself, instead requiring the final states in these steps to be related in both directions. This is what makes our notion a directed bisimulation, rather than simply a simulation.

Definition 2.18. A relation R is a *directed branching bisimulation* if whenever $R(s, t)$ and $s \twoheadrightarrow_\tau s' \rightarrow_{(\alpha)} s''$, there exist t', t'' such that $t \twoheadrightarrow_\tau t' \rightarrow_{(\alpha)} t''$, $R(s', t')$, $R(s'', t'')$, and $R(t'', s'')$. See the right side of Figure 1.

Two states are *directed branching bisimilar* (notation $s \twoheadrightarrow_{db} t$) if there exists a directed branching bisimulation relating them. We write $s \leftrightarrow_{db} t$ as a shorthand for “ $s \twoheadrightarrow_{db} t$ and $t \twoheadrightarrow_{db} s$ ”.

A number of properties that are known for branching bisimulation have simple proofs in the directed setting. In particular, the stuttering lemma can be stated in the following, more specific, way.

Lemma 2.19 (Stuttering lemma). *See Figure 2 on the left. Given states $s \twoheadrightarrow_{db} t$, we have for all q, r ,*

- *if $s \twoheadrightarrow_\tau q$ then $q \twoheadrightarrow_{db} t$, and*
- *if $r \twoheadrightarrow_\tau t$ then $s \twoheadrightarrow_{db} r$.*

This can be proven using a stuttering closure construction like the one used by van Glabbeek et al. [vGLT09], but we will instead prove its dual, Lemma 2.23.

Theorem 2.20. *Two states are branching bisimilar precisely when they are directed branching bisimilar in both directions:*

$$s \leftrightarrow_b t \quad \Longleftrightarrow \quad s \twoheadrightarrow_{db} t \wedge t \twoheadrightarrow_{db} s$$

Proof. Every branching bisimilarity is a directed branching bisimilarity, giving the left to right direction.

For the other direction, it suffices to show that the relation $\leftrightarrow_{db}$ is a branching bisimilarity. This is easily verified: it is symmetric, and given s, s', t such that $s \rightarrow_\alpha s'$ and $s \leftrightarrow_{db} t$, there exist some t', t'' such that $t \twoheadrightarrow_\tau t' \rightarrow_{(\alpha)} t''$ with $s \twoheadrightarrow_{db} t'$ and $s' \leftrightarrow_{db} t''$. It remains to show that $t' \twoheadrightarrow_{db} s$: this follows from the Stuttering Lemma, since $t \twoheadrightarrow_{db} s$ and $t \twoheadrightarrow_\tau t'$. $\square$

2.3. Directed branching apartness. In the previous two subsections, we have introduced Positive Hennessy-Milner Logic with Until and directed branching bisimulation, which together are enough to formulate a directed Hennessy-Milner theorem. In this subsection, we introduce directed branching apartness, the dual relation of directed branching bisimulation. Using apartness makes the proof of our directed Hennessy-Milner theorem a straightforward induction in both directions.

Definition 2.21. A relation Q is a *directed branching apartness* if the following derivation rule holds:

$$\frac{s \twoheadrightarrow_{\tau} s' \rightarrow_{(\alpha)} s'' \quad \forall t', t''. t \twoheadrightarrow_{\tau} t' \rightarrow_{(\alpha)} t'' \implies Q(s', t') \vee Q(s'', t'') \vee Q(t'', s'')}{Q(s, t)} \text{IN}_{db}$$

Two states s, t are *directed branching apart*, notation $s \not\#_{db} t$, if for every directed branching apartness Q , it holds that $Q(s, t)$. We use $s \#_{db} t$ as a shorthand for “ $s \not\#_{db} t$ or $t \not\#_{db} s$ ”.

Just as for directed strong apartness (Definition 1.15), directed branching apartness is an *inductive* notion, so it can be characterized as a derivation system. For directed branching apartness, we have a lemma similar to Lemma 1.16. The proof is straightforward.

Lemma 2.22. We have that $s \#_{db} t$ if and only this can be derived using the following rule.

$$\frac{s \twoheadrightarrow_{\tau} s' \rightarrow_{(\alpha)} s'' \quad \forall t', t''. t \twoheadrightarrow_{\tau} t' \rightarrow_{(\alpha)} t'' \implies s' \not\#_{db} t' \vee s'' \not\#_{db} t''}{s \not\#_{db} t} \text{IN}_{db}$$

Furthermore, $s \#_{db} t \iff \neg(s \twoheadrightarrow_{db} t)$.

We again note, as we did after Lemma 1.16, that the base case of the inductive definition of $s \not\#_{db} t$ is when s can do an eventual α -step while t cannot. Then the hypothesis is vacuously satisfied and we have $s \not\#_{db} t$.

For all s, t we have either $s \twoheadrightarrow_{db} t$ or $s \not\#_{db} t$, depending on whether there is any directed branching bisimulation that relates them. So we can obtain results about $\not\#_{db}$ from $\twoheadrightarrow_{db}$ (and vice versa) by contraposition. The following lemma is the Stuttering Lemma for apartness, the dual of Lemma 2.19.

Lemma 2.23. Given states $s \not\#_{db} t$, we have for all r, q ,

- if $r \twoheadrightarrow_{\tau} s$ then $r \not\#_{db} t$; and
- if $t \twoheadrightarrow_{\tau} q$ then $s \not\#_{db} q$.

See the right side of Figure 2.

Proof. Since $s \not\#_{db} t$, there exists (by Lemma 2.22) some eventual α -step $s \twoheadrightarrow_{\tau} s' \rightarrow_{(\alpha)} s''$ such that for all eventual α -steps $t \twoheadrightarrow_{\tau} t' \rightarrow_{(\alpha)} t''$ we have $s' \not\#_{db} t'$ or $s'' \not\#_{db} t''$.

Given $r \twoheadrightarrow_{\tau} s$, we can use the same eventual α -step $r \twoheadrightarrow_{\tau} s \twoheadrightarrow_{\tau} s' \rightarrow_{(\alpha)} s''$ to show that $r \not\#_{db} t$, since the condition on eventual α -steps from t does not mention s itself.

Similarly, given $t \twoheadrightarrow_{\tau} q$, any eventual α -step from q is in fact an eventual α -step from t . Hence for any such step $q \twoheadrightarrow_{\tau} q' \rightarrow_{(\alpha)} q''$ we have $s' \not\#_{db} q'$ or $s'' \not\#_{db} q''$, as desired. $\square$

We now state a theorem similar to Theorem 1.17 for the branching case: $s \not\#_{ds} t$ iff there is some PHMLU formula that distinguishes s and t (from which the well-known

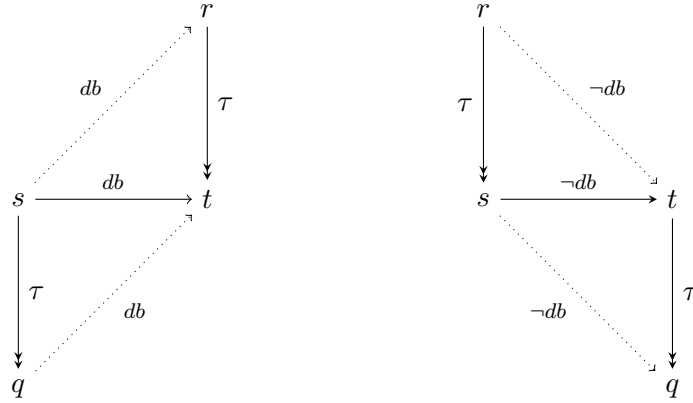


Figure 2: Stuttering using directed branching bisimilarity (left) and directed branching apartness (right)

Hennessey-Milner theorem for branching bisimulation of [dNV95], Theorem 2.6, immediately follows). Moreover we have a direct proof by induction.

Theorem 2.24. *In an image-finite labelled transition system we have, for states s and t ,*

$$s \not\#_{db} t \iff \exists \varphi. \varphi \in \text{Th}_P(s) - \text{Th}_P(t).$$

Equivalently, s and t are directed branching bisimilar, $s \Rightarrow_{db} t$, precisely when $\text{Th}_P(s) \subseteq \text{Th}_P(t)$.

Proof. We only prove the first statement. Both directions of the proof proceed by structural induction.

$\Rightarrow$: Given a derivation of $s \not\#_{db} t$, we know it was obtained by applying derivation rule IN_{db} , and hence there exist s', s'' such that $s \rightarrow_{\tau} s' \rightarrow_{(\alpha)} s''$ such that for all t', t'' , if $t \rightarrow_{\tau} t' \rightarrow_{(\alpha)} t''$ then $s' \not\#_{db} t'$, $s'' \not\#_{db} t''$ or $t'' \not\#_{db} s''$. By induction, for each of these cases there is a distinguishing formula, which we can collect into sets Δ , Ψ^+ , and Ψ^- . Let $\delta = \bigwedge \Delta$, $\psi^+ = \bigwedge \Psi^+$, and $\psi^- = \bigvee \Psi^-$. We have that $s' \models \delta$, $s'' \models \psi^+$, and $s'' \not\models \psi^-$, while for all t', t'' with $t \rightarrow_{\tau} t' \rightarrow_{(\alpha)} t''$, we have $t' \not\models \delta$, $t'' \not\models \psi^+$, or $t'' \models \psi^-$. Let $\varphi = \delta \langle \alpha \rangle \psi^+ \wedge \neg \psi^-$; by construction, $s \models \varphi$ while $t \not\models \varphi$, hence φ is a positive distinguishing formula.

$\Leftarrow$: In this direction, only the modality case is of interest. Suppose $\varphi = \delta \langle \alpha \rangle \psi^+ \wedge \neg \psi^-$ is a positive distinguishing formula for s and t . Then there exist s', s'' such that $s \rightarrow_{\tau} s' \rightarrow_{(\alpha)} s''$, such that $s' \models \delta$, $s'' \models \psi^+$, and $s'' \not\models \psi^-$, while for all t', t'' such that $t \rightarrow_{\tau} t' \rightarrow_{(\alpha)} t''$ one of $t' \not\models \delta$, $t'' \not\models \psi^+$, and $t'' \models \psi^-$ holds. By induction, we can conclude $s' \not\#_{db} t'$, $s'' \not\#_{db} t''$, or $t'' \not\#_{db} s''$, which is exactly what we need to derive the apartness $s \not\#_{db} t$. $\square$

The proof of Theorem 2.24 gives an algorithm to generate a distinguishing PHMLU formula from a proof of directed apartness and vice versa. It extends the algorithms for the strong and weak bisimulation case as described in [Gau22]. We illustrate this using a well-known LTS with states s and r that are not branching bisimilar. We give the proof of $s \not\#_{db} r$ and the corresponding PHMLU formula that distinguishes s and t . In Appendix B we describe the algorithm and give an example of the reverse direction.

Example 2.25. In the following well-known LTS, s and r are not branching bisimilar. We give a proof of $s \not\#_{db} r$ and the distinguishing formula that is computed from that proof.



In the derivation of $s \not\equiv_{db} r$, we indicate above the $\forall r', r'' \dots$ all the (finitely many) possible transitions that we need to prove a property for: just one in the case of the c -step; none in the case of the d -step (which we indicate by $\checkmark$).

$$\begin{array}{c}
 \checkmark \\
 \hline
 s \rightarrow_d s_3 \quad \forall r', r''. r_1 \rightarrow_\tau r' \rightarrow_d r'' \implies s \not\equiv_{db} r' \vee s_3 \not\equiv_{db} r'' \\
 \hline
 s \not\equiv_{db} r_1 \\
 \hline
 s \not\equiv_{db} r_1 \vee s_2 \not\equiv_{db} r_3 \\
 \hline
 s \rightarrow_c s_2 \quad \forall r', r''. r \rightarrow_\tau r' \rightarrow_c r'' \implies s \not\equiv_{db} r' \vee s_2 \not\equiv_{db} r'' \\
 \hline
 s \not\equiv_{db} r
 \end{array}$$

The formula that we compute from this proof is $(\langle d \rangle \top) \langle c \rangle \top$, and indeed $s \models (\langle d \rangle \top) \langle c \rangle \top$ and $r \not\models (\langle d \rangle \top) \langle c \rangle \top$. The formula expresses that there is a τ -path to a state where a c -step is possible, and in all states along that τ -path, a d -step is possible.

Using the Stuttering Lemma 2.19 we can prove the following interesting alternative characterization of directed branching bisimulation.

Theorem 2.26. *For states s, t , we have $s \Rightarrow_{db} t$ precisely when there exists some r such that $t \rightarrow_\tau r$ and $s \Leftarrow_{db} r$.*

Proof. Suppose $s \Rightarrow_{db} t$. Since $s \rightarrow_\tau s \rightarrow_{(\tau)} s$ is an eventual τ -step, there exist t' and t'' such that $t \rightarrow_\tau t' \rightarrow_{(\tau)} t''$, $s \Rightarrow_{db} t'$, and $s \Leftarrow_{db} t''$. For the reverse implication, suppose $t \rightarrow_\tau r$ and $s \Leftarrow_{db} r$. Then also $s \Rightarrow_{db} r$, so by the Stuttering Lemma 2.19, we have $s \Rightarrow_{db} t$. $\square$

This characterization, together with Theorem 2.24, gives us the following corollary.

Corollary 2.27. *Given two states s, t in an image-finite LTS_τ , $\text{Th}_P(s) \subseteq \text{Th}_P(t)$ precisely when there exists some t' such that $t \rightarrow_\tau t'$ and $\text{Th}_P(s) = \text{Th}_P(t')$.*

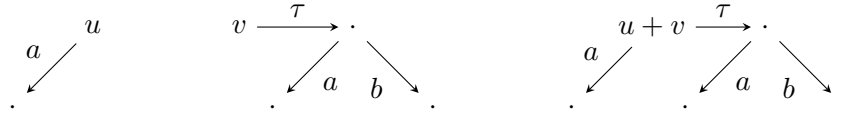
This result is particularly interesting because it does not mention bisimulation or apartness, but it is unclear how one would go about proving it purely within (P)HMLU. While for every formula $\varphi \in \text{Th}_P(s) - \text{Th}_P(t)$ we know $t \models \langle \tau \rangle \neg \varphi$ and hence there is some s' such that $s \rightarrow_\tau s'$ and $s' \not\models \varphi$, we see no way to conclude that a single s' will work independent of φ . The equivalence with directed branching bisimulation, however, gives us exactly this.

2.4. Relation to other notions. In the non-directed case, strong bisimulation implies branching bisimulation, so it may come as a surprise that this does not hold in the directed case. This can be seen as follows: a state with no outgoing transitions is directed strongly bisimilar to every other state, since the condition that every transition can be transferred is satisfied vacuously. In other words, its positive theory contains no formulas containing modalities, and it is thus minimal with respect to inclusion. However, in the branching case a state with no outgoing transitions satisfies formulas such as $\langle \tau \rangle \neg \langle a \rangle \top$, which are not true in every state.

A similar surprise can be found by comparing directed branching bisimulation to the following notion introduced by Bergstra and Klop in [BK85], who are working in the context

of process algebras and have an addition operation on states. In our terms, the state $s + t$ should be seen as the state that can do anything s or t can do; in other words, $s + t \rightarrow_a r$ precisely when $s \rightarrow_a r$ or $t \rightarrow_a r$. Bergstra and Klop define $s \sqsubseteq t$ as follows: $s = t$ or there exists a r such that $s + r = t$. In our context, it makes most sense to interpret this equality as bisimulation.

If we treat the equality as strong bisimulation, we in fact have that $s \rightarrow_{ds} t$ precisely when $s \sqsubseteq t$. However, in the branching case this does not hold: if s is the state with no outgoing transitions, then $s \sqsubseteq t$ for every t , but $s \rightarrow_{db} t$ does not in general hold. A reviewer also pointed out the following example, which shows that there exist states u and v such that $u \rightarrow_{db} v$ but not $u + v \rightarrow_{db} v$, and hence that directed branching bisimulation is not well-behaved under such operation on states:



Both the comparison to directed strong bisimulation and to the $\sqsubseteq$ relation raise the question: why does directed branching bisimulation not relate states s and t even though t appears to be able to do more than s can?

As we will show in a moment, the relation $\rightarrow_{dsb} = \rightarrow_{ds} \cup \rightarrow_{db}$ also satisfies the property that $p \leftrightarrow_b q$ precisely when $p \rightarrow_{dsb} q \wedge q \rightarrow_{dsb} p$.⁵ In this sense, we could weaken the relation $\rightarrow_{db}$ to obtain one that behaves well with respect to $\sqsubseteq$. However, our result is not simply the existence of the $\rightarrow_{db}$ relation: we show that it can be defined coinductively, that it has an inductive complement, and prove a Hennessy-Milner theorem that shows that p and q being directed branching bisimilar is equivalent to the PHMLU theory of p being a subset of that of q , which in turn is equivalent to every positive and left-positive HMLU formula that holds in p also holding in q .

Finding a coinductive presentation and a corresponding logic for the relation $\rightarrow_{dsb}$ (or for some other relation $\rightarrow_{dsb} \subseteq R \subset \leftrightarrow_b$) would be interesting future work. However, such work would not invalidate or supersede ours: the Hennessy-Milner theorem for a (coinductively defined) $\rightarrow_{dsb}$ and some modal logic would simply show a connection between those two; the connection between $\rightarrow_{db}$ and PHMLU is independent of that.

Theorem 2.28. *In any LTS_τ , for all p, q we have $p \leftrightarrow_b q$ iff $p \rightarrow_{dsb} q$ and $q \rightarrow_{dsb} p$.*

Proof. If $p \leftrightarrow_b q$ then by Theorem 2.20 we have $p \rightarrow_{db} q$ and $q \rightarrow_{db} p$, which suffices.

For the other direction, the only interesting case is when (without loss of generality) $p \rightarrow_{ds} q$ and $q \rightarrow_{db} p$, since strong bisimilarity implies branching bisimilarity in the undirected case.

We show that in this case, $p \rightarrow_{db} q$ by showing that $\text{Th}_P(p) \subseteq \text{Th}_P(q)$ by induction on the formula. As in other proofs, only the case when the formula is of the form $\delta\langle\alpha\rangle\psi$ is interesting.

Suppose that $p \models \delta\langle\alpha\rangle\psi$, then there is some path $p \xrightarrow{\delta}_\tau p' \rightarrow_{(\alpha)} p''$ such that $p'' \models \psi$. By induction, $q \models \delta$. If this path has at least one step then we are done: if $p \neq p'$ then by repeated applications of directed strong bisimilarity we can construct a path $q \xrightarrow{\delta}_\tau q'$ such that $p' \leftrightarrow_s q'$ and hence $p' \leftrightarrow_b q'$. By our choice of p' we have $p' \models \delta\langle\alpha\rangle\psi$ and hence by

⁵Strictly speaking, we did not define directed strong bisimulation for an LTS with silent steps. We interpret τ as being no different than any other step for the purposes of directed strong bisimulation.

bisimilarity $q' \models \delta\langle\alpha\rangle\psi$, and by positivity $q \models \delta\langle\alpha\rangle\psi$. Similarly, if $p = p'$ but $p' \neq p''$ then using directed strong bisimilarity we get $p'' \xleftrightarrow{s} q''$ and thus $p'' \xleftrightarrow{b} q''$. Since $p'' \models \psi$ also $q'' \models \psi$, and thus $q \models \delta\langle\alpha\rangle\psi$.

It remains to consider the case when the path is trivial, $p = p' = p''$, and thus $\alpha = \tau$. We use the fact we can write $\psi = \psi^+ \wedge \neg\psi^-$, where both ψ^+ and ψ^- are PHMLU formulas, and we have $p \models \psi^+$ and $p \not\models \psi^-$. Now since ψ^+ is a PHMLU formula, by our induction hypothesis we have $q \models \psi^+$. On the other hand, since $q \xrightarrow{db} p$, every PHMLU formula that holds on q also holds in p , and by the contraposition of this we get $q \not\models \psi^-$. It follows that $q \models \delta\langle\alpha\rangle\psi$ via a trivial path as well, which concludes the proof. $\square$

3. RELATED WORK

Our approach for defining formulas is somewhat similar to that taken by Beohar et al in [BGKM23]. There too, the ‘formulas’ (elements of $\mathbb{L}$ in their terminology) are only closed under negation under the modality. This corresponds to our presentation of PHML in this paper. It would be interesting to see whether our results can be formulated in their setting.

Our proof of Theorem 2.14, while discovered independently, is very similar to the proofs from Section 6 of van Glabbeek et al. [vGLT09], where the authors show that an ‘until’ modality (denoted by $\delta\langle\alpha\rangle\psi$, where s satisfies $\delta\langle\alpha\rangle\psi$ when there is a path $s \xrightarrow{\delta}_{\tau} s' \rightarrow_{(\alpha)} s''$ with $s'' \models \psi$) is equally expressive as a ‘just-before’ modality (denoted by $\delta\alpha\psi$, where s satisfies $\delta\alpha\psi$ when there is a path $s \rightarrow_{\tau} s' \rightarrow_{(\alpha)} s''$ with $s' \models \delta$ and $s'' \models \psi$). Note that when δ is positive, this is trivial, since the same paths satisfy both conditions, and in fact this same approach is how van Glabbeek et al. proved their results.

The case of branching bisimulation and a modal logic with a ‘just-before’ modality has been studied by Martens and Groote [MG24], who proved a Hennessy-Milner using an inductive argument on the apartness derivation, precisely of the sort that Geuvers [Geu22] set out as an open question. Given the aforementioned equivalence in power between the ‘until’ and ‘just-before’ modalities, the two results imply each other.

A notion similar to directed bisimulation is the one introduced by Bergstra and Klop in [BK85], who are working in the context of process algebras and have defined an addition operation.

Moreover, even for strong bisimulation we feel that our presentation gives something that is not covered by the process algebra view. Namely, directed bisimulation is built up coinductively, and has an inductive dual. This means that the structure of a proof of $s \xrightarrow{ds} t$ or $s \not\xrightarrow{ds} t$ can be useful when reasoning about a labelled transition system. For the same reason, we do not use the characterization in Theorem 2.26 as our definition of directed branching bisimulation.

For Kripke structures, a notion of “directed modal simulation” has been studied by Kurtonina and de Rijke in [KdR97], where it is also related to theories of positive formulas in modal logic. Bisimulation for Kripke structures is comparable to strong bisimulation for LTSs, so silent steps are not treated. One of the goals of [KdR97] is to capture the class of modal formulas that are preserved by a directed modal simulation, which includes box formulas, but does not include formulas where a negation appears under a modality. As a result, their directed modal simulation is incomparable to ours: it includes a condition for backwards transfer, which ours does not, but does not require that successor states be bisimilar.

Notions of bisimilarity exist for a large variety of systems and calculi (process algebras, π -calculus, ...), for example see [San11]. Desharnais et al [DGJP04] have studied bisimulation relations on Markov processes, and in their work, bisimulation is equivalent to two-way simulation. In particular, we would call their Proposition 2.11 a directed Hennessy-Milner theorem.

4. CONCLUSION

We have formulated and proved a directed Hennessy-Milner theorem, by adapting known Hennessy-Milner logic to a positive variant and by adapting bisimulation to directed bisimulation. From this directed version, the usual Hennessy-Milner theorem follows as a corollary. We also show that the dual of bisimulation, apartness (and its directed variant), enables a direct proof of the Hennessy-Milner theorem by induction.

We have illustrated this using LTSs (labelled transition systems) with strong bisimulation and using LTSs with silent steps with branching bisimulation. The situation for LTSs with silent steps and weak bisimulation is very similar to strong bisimulation. Applications of our approach to other logics and notions of bisimilarity would be of interest. Variations like divergence-preserving branching bisimulation [vGLT09, Lut20] would be interesting to consider from this directed perspective.

In the branching setting, our “directed approach” brings the logic and bisimulation sides of the Hennessy Milner theorem closer together. There is a remarkable similarity between proving that a positive formula holds in a state and proving that a state is on the right of a directed bisimilarity. Namely, to show that $t \models \delta(\alpha)(\psi^+ \wedge \neg\psi^-)$ we show

$$\exists t', t''. t \twoheadrightarrow_{\tau} t' \rightarrow_{(\alpha)} t'' \wedge t' \models \delta \wedge t'' \models \psi^+ \wedge t'' \not\models \psi^-,$$

while to show that $s \rightarrow_{ds} t$, we show that for every step $s \rightarrow_{\alpha} s'$,

$$\exists t', t''. t \twoheadrightarrow_{\tau} t' \rightarrow_{(\alpha)} t'' \wedge s \rightarrow_{ds} t' \wedge s' \rightarrow_{ds} t'' \wedge t'' \rightarrow_{ds} s'.$$

Similarly, showing that a formula fails to hold in a state is akin to showing that the state is on the right of a directed apartness.

This level of similarity suggests that there should be some way of constructing a modal logic from a coinductive notion of bisimilarity, or vice-versa, but this has so far proved elusive. For bisimulation relations, this question is raised in a considerably more general setting by Kupke and Rot [KR21]. The work of Beohar et al [BGKM23] may be relevant to this point.

REFERENCES

- [Bas96] T. Basten. Branching bisimilarity is an equivalence indeed! *Information Processing Letters*, 58(3):141–147, 1996. doi:10.1016/0020-0190(96)00034-8.
- [BGKM23] H. Beohar, S. Gurke, B. König, and K. Messing. Hennessy-Milner theorems via Galois connections. In *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*, volume 252 of *Leibniz International Proceedings in Informatics*, pages 12:1–12:18. Schloss Dagstuhl – Leibniz-Zentrum, February 2023. doi:10.4230/LIPIcs.CSL.2023.12.
- [BK85] J. A. Bergstra and J. W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37:77–121, 1985. doi:10.1016/0304-3975(85)90088-X.
- [DGJP04] Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled Markov processes. *Theoretical Computer Science*, 318(3):323–354, 2004. doi:10.1016/j.tcs.2003.09.013.

- [dNV95] R. de Nicola and F. Vaandrager. Three logics for branching bisimulation. *J. ACM*, 42(2):458–487, March 1995. doi:10.1145/201019.201032.
- [Geu22] H. Geuvers. Apartness and distinguishing formulas in Hennessy-Milner logic. In N. Jansen, M. Stoelinga, and P. van den Bos, editors, *A Journey from Process Algebra via Timed Automata to Model Learning: Essays Dedicated to Frits Vaandrager on the Occasion of His 60th Birthday*, pages 266–282. Springer Nature Switzerland, Cham, 2022. doi:10.1007/978-3-031-15629-8_14.
- [GJ21] Herman Geuvers and Bart Jacobs. Relating apartness and bisimulation. *Logical Methods in Computer Science*, 17(3), July 2021. doi:10.46298/lmcs-17(3:15)2021.
- [GV90] J. F. Groote and F. Vaandrager. An efficient algorithm for branching bisimulation and stuttering equivalence. In M. S. Paterson, editor, *Automata, Languages and Programming*, pages 626–638, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg. doi:10.1007/BFb0032063.
- [HM85] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, January 1985. doi:10.1145/2455.2460.
- [JGKW20] D. N. Jansen, J. F. Groote, J. J. A. Keiren, and A. Wijs. An $O(m \log n)$ algorithm for branching bisimilarity on labelled transition systems. In A. Biere and D. Parker, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 3–20, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-45237-7_1.
- [KdR97] N. Kurtonina and M. de Rijke. Simulating without negation. *Journal of Logic and Computation*, 7(4):501–522, July 1997. doi:10.1093/logcom/7.4.501.
- [KR21] Clemens Kupke and Jurriaan Rot. Expressive logics for coinductive predicates. *Logical Methods in Computer Science*, 17(4), December 2021. doi:10.46298/lmcs-17(4:19)2021.
- [Lut20] B. Luttik. Divergence-preserving branching bisimilarity. In O. Dardha and J. Rot, editors, *Proceedings Combined 27th International Workshop on Expressiveness in Concurrency and 17th Workshop on Structural Operational Semantics*, Online, 31 August 2020, volume 322 of *Electronic Proceedings in Theoretical Computer Science*, pages 3–11. Open Publishing Association, 2020. doi:10.4204/EPTCS.322.2.
- [MG24] J. Martens and J.F. Groote. Minimal depth distinguishing formulas without until for branching bisimulation. In V. Capretta, R. Krebbers, and F. Wiedijk, editors, *Logics and Type Systems in Theory and Practice: Essays Dedicated to Herman Geuvers on The Occasion of His 60th Birthday*, pages 188–202. Springer Nature Switzerland, Cham, 2024. doi:10.1007/978-3-031-61716-4_12.
- [San11] D. Sangiorgi. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, 2011. doi:10.1017/CB09780511777110.
- [vG93] R. J. van Glabbeek. The linear time — branching time spectrum II. In E. Best, editor, *CONCUR'93*, pages 66–81, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg. doi:10.1007/3-540-57208-2_6.
- [vGLT09] R. J. van Glabbeek, B. Luttik, and N. Trčka. Branching bisimilarity with explicit divergence. *Fundamenta Informaticae*, 93(4):319–392, 2009. doi:10.3233/FI-2009-109.
- [vGW89] R. J. van Glabbeek and W. P. Weijland. Branching time and abstraction in bisimulation semantics (extended abstract). In G. X. Ritter, editor, *Information Processing 89, 11th World Computer Congress, San Francisco 1989*, pages 613–618. North-Holland Publishing Company, September 1989. doi:10.1145/233551.233556.
- [vGW96] R. J. van Glabbeek and W. P. Weijland. Branching time and abstraction in bisimulation semantics. *J. ACM*, 43(3):555–600, May 1996. doi:10.1145/233551.233556.

APPENDIX A. PROOFS

In this appendix we present the full proof of Theorem 2.14.

Let us begin by recalling some conventions. The modality binds tightly on the left and loosely on the right; $\delta \wedge \gamma \langle \alpha \rangle \varphi \wedge \psi$ should be read as $\delta \wedge (\gamma \langle \alpha \rangle (\varphi \wedge \psi))$. The set $\pm\text{PHMLU}$ is the set of PHMLU formulas and their negations and $\bigwedge \pm\text{PHMLU}$ is the set of conjunctions of $\pm\text{PHMLU}$ formulas.

Given a $\bigwedge \pm\text{PHMLU}$ formula φ , we use φ^+ and φ^- to denote the decomposition of φ into PHMLU formulas such that φ is equivalent to $\varphi^+ \wedge \neg \varphi^-$. Concretely, φ^+ is the

conjunction of the conjuncts of φ that are PHMLU formulas, while φ^- is the disjunction of those ψ for which $\neg\psi$ is a conjunct of φ .

Given formulas $\delta \in \text{PHMLU}$ and $\varphi \in \bigwedge \pm \text{PHMLU}$, we write $\delta\langle\alpha\rangle\varphi$ to denote the PHMLU formula $\delta\langle\alpha\rangle\varphi^+ \wedge \neg\varphi^-$.

Definition A.1. We say that a formula φ *entails* a formula ψ if in every state s of every LTS, if $s \models \varphi$ then $s \models \psi$. We say φ and ψ are *equivalent* if each entails the other.

To aid in our proof, we define two functions

$$\Gamma_\alpha : \text{HMLU}^+ \times \text{HMLU} \rightarrow \text{HMLU}$$

$$\Gamma_\alpha^P : \left(\bigwedge \pm \text{PHMLU} \right)^+ \times \bigwedge \pm \text{PHMLU} \rightarrow \bigwedge \pm \text{PHMLU}$$

where X^+ denotes the set of finite non-empty sequences with elements from X . We define both functions recursively:

$$\Gamma_\alpha(\delta; \psi) := \delta\langle\alpha\rangle\psi$$

$$\Gamma_\alpha(\delta_1, \delta_2, \dots, \delta_n; \psi) := \delta_1\langle\tau\rangle\Gamma_\alpha(\delta_2, \dots, \delta_n; \psi)$$

$$\Gamma_\alpha^P(\delta; \psi) := \neg\delta^- \wedge \delta^+\langle\alpha\rangle\psi$$

$$\Gamma_\alpha^P(\delta_1, \delta_2, \dots, \delta_n; \psi) := \neg\delta_1^- \wedge \delta_1^+\langle\tau\rangle\Gamma_\alpha^P(\delta_2, \dots, \delta_n; \psi).$$

We begin by proving a number of lemmas.

Lemma A.2. Given $\vec{\delta}, \vec{\theta} \in \text{HMLU}^+$ and $\psi \in \text{HMLU}$, with $\vec{\delta}$ and $\vec{\theta}$ of equal length and each δ_i entailing θ_i , $\Gamma_\alpha(\vec{\delta}; \psi)$ entails $\Gamma_\alpha(\vec{\theta}; \psi)$.

Proof. For every state s , the path witnessing $s \models \Gamma_\alpha(\vec{\delta}; \psi)$ also witnesses $s \models \Gamma_\alpha(\vec{\theta}; \psi)$. $\square$

Lemma A.3. Given $\delta, \psi \in \bigwedge \pm \text{PHMLU}$, the formulas $\delta\langle\alpha\rangle\psi$ and $\neg\delta^- \wedge \delta^+\langle\alpha\rangle\psi$ are equivalent.

Proof. Note that this does not follow directly from δ and $\delta^+ \wedge \neg\delta^-$ being equivalent: by our convention on parentheses, the formula $\neg\delta^- \wedge \delta^+\langle\alpha\rangle\psi$ should be read as $\neg\delta^- \wedge (\delta^+\langle\alpha\rangle\psi)$.

Let s be an arbitrary state. If $s \models \delta\langle\alpha\rangle\psi$, then in particular $s \models \delta$ and hence $s \models \neg\delta^-$. Moreover, the path $s \xrightarrow{\delta} s' \xrightarrow{(\alpha)} s''$ witnessing $s \models \delta\langle\alpha\rangle\psi$ also gives us $s \models \delta^+\langle\alpha\rangle\psi$.

For the other direction, suppose $s \models \neg\delta^- \wedge (\delta^+\langle\alpha\rangle\psi)$. There is some path $s \xrightarrow{\delta^+} s' \xrightarrow{(\alpha)} s''$ witnessing the second conjunct, with $s'' \models \psi$. By Lemma 2.9, since $s \models \neg\delta^-$ and $\neg\delta^-$ is negative, every state on the path $s \xrightarrow{\delta^+} s'$ satisfies $\neg\delta^-$, and hence $s \xrightarrow{\delta} s'$. Thus $s \models \delta\langle\tau\rangle\psi$, concluding the proof. $\square$

Lemma A.4. Given a sequence $\vec{\delta} \in (\bigwedge \pm \text{PHMLU})^+$ and a $\psi \in \bigwedge \pm \text{PHMLU}$, $\Gamma_\alpha^P(\vec{\delta}; \psi)$ is equivalent to $\Gamma_\alpha(\vec{\delta}; \psi)$.

Proof. The argument proceeds by an induction on the length of $\vec{\delta}$. In both the base case and the inductive case, after writing out the definitions, the result follows from Lemma A.3. $\square$

Given a set I , let $\text{NRS}(I)$ be the set of non-empty non-repeating sequences with elements from I . Note that if I is finite, then so is $\text{NRS}(I)$.

Lemma A.5. Let $\vec{\delta} = \delta_1, \dots, \delta_n$ and ψ be $\bigwedge \pm \text{PHMLU}$ formulas and suppose $\delta_{l_1} = \delta_{l_2}$ for $l_1 < l_2$. Then $s \models \Gamma_\alpha^P(\vec{\delta}; \psi)$ entails $s \models \Gamma_\alpha^P(\delta_1, \dots, \delta_{l_1}, \delta_{l_2+1}, \dots, \delta_n; \psi)$.

Proof. For this proof, the following notation is convenient:

$$s \rightsquigarrow_{\alpha}^{\delta} t \quad := \quad \exists r. s \rightarrow_{\tau}^{\delta} r \rightarrow_{(\alpha)} t.$$

Suppose $s \models \Gamma_{\alpha}^P(\vec{\delta}; \psi)$. Unfolding the definitions, this means that there is a path

$$s = s_1 \rightsquigarrow_{\tau}^{\delta_1^+} s_2 \rightsquigarrow_{\tau}^{\delta_2^+} \dots \rightsquigarrow_{\alpha}^{\delta_n^+} s_{n+1}$$

with for all $1 \leq k \leq n$, $s_k \models \neg \delta_k^-$, and with $s_{n+1} \models \psi$.

We have $s_{l_1} \rightsquigarrow_{\tau}^{\delta_{l_1}^+} s_{l_2+1}$: since there is some r such that $s_{l_2} \rightarrow_{\tau}^{\delta_{l_2}^+} r \rightarrow_{(\tau)} s_{l_2+1}$, and since $\delta_{l_2}^+$ is positive, all states on this path leading up to r satisfy $\delta_{l_2}^+$, which by assumption is equal to $\delta_{l_1}^+$.

There is thus a path

$$s = s_1 \rightsquigarrow_{\tau}^{\delta_1^+} \dots \rightsquigarrow_{\tau}^{\delta_{l_1-1}^+} s_{l_1} \rightsquigarrow_{\tau}^{\delta_{l_1}^+} s_{l_2+1} \rightsquigarrow_{\tau}^{\delta_{l_2+1}^+} \dots \rightsquigarrow_{\alpha}^{\delta_n^+} s_{n+1},$$

where for all k between 1 and l_1 and between $l_2 + 1$ and n , $s_k \models \neg \delta_k^-$, which is a witness of $s \models \Gamma_{\alpha}^P(\delta_1, \dots, \delta_{l_1}, \delta_{l_2+1}, \dots, \delta_n; \psi)$, as desired. $\square$

Lemma A.6. *Given a finite family $(\delta_i)_{i \in I}$ of $\bigwedge \pm$ PHMLU formulas and a $\bigwedge \pm$ PHMLU formula ψ , the formula $\varphi = (\bigvee_i \delta_i) \langle \alpha \rangle \psi$ is equivalent to*

$$\bigvee_{\vec{i} \in \text{NRS}(I)} \Gamma_{\alpha}^P(\delta_{i_1}, \dots, \delta_{i_n}; \psi).$$

Proof. Let s be a state. We show that φ holds in s precisely when some $\vec{i} \in \text{NRS}(I)$ exists such that $s \models \Gamma_{\alpha}^P(\delta_{i_1}, \dots, \delta_{i_n}; \psi)$. Let $\check{\delta}$ be a shorthand for $\bigvee_i \delta_i$.

For the left-to-right direction, suppose $s \models \varphi$, and hence there exist s', s'' such that $s \rightarrow_{\tau}^{\check{\delta}} s' \rightarrow_{(\alpha)} s''$ and $s'' \models \psi$. In particular, there exists a sequence $s = s_1 \rightarrow_{\tau} \dots \rightarrow_{\tau} s_n = s'$ such that for each $k \in \{1, \dots, n\}$, $s_k \models \check{\delta}$. It follows that for every k there is some disjunct with index i_k such that $s_k \models \delta_{i_k}$. We thus see that $s \models \Gamma_{\alpha}(\delta_{i_1}, \dots, \delta_{i_n}; \psi)$ and by Lemma A.4, we have $s \models \Gamma_{\alpha}^P(\delta_{i_1}, \dots, \delta_{i_n}; \psi)$.

We can now repeatedly use Lemma A.5 to remove any duplicate indices that occur in $\vec{i}$, giving us $\vec{i}' \in \text{NRS}(I)$ such that $s \models \Gamma_{\alpha}^P(\delta_{i'_1}, \dots, \delta_{i'_n}; \psi)$.

Conversely, suppose that there exists a $\vec{i} \in \text{NRS}(I)$ such that $s \models \Gamma_{\alpha}^P(\delta_{i_1}, \dots, \delta_{i_n}; \psi)$, and thus, by Lemma A.4, $s \models \Gamma_{\alpha}(\delta_{i_1}, \dots, \delta_{i_n}; \psi)$. By Lemma A.2, $\Gamma_{\alpha}(\delta_{i_1}, \dots, \delta_{i_n}; \psi)$ entails $\Gamma_{\alpha}(\check{\delta}, \dots, \check{\delta}; \psi)$, which is equivalent to $\Gamma_{\alpha}(\check{\delta}; \psi) = \varphi$. It follows that $s \models \varphi$, which is exactly the desired conclusion. $\square$

Theorem A.7 (Restatement of Theorem 2.14). *For every HMLU formula φ there exists an equivalent disjunction of $\bigwedge \pm$ PHMLU formulas. Moreover, if φ is positive then it is equivalent to a disjunction of PHMLU formulas, and hence to a PHMLU formula.*

Proof. The proof proceeds by induction, with all cases except the modality being straightforward.

Suppose $\varphi = \delta \langle \alpha \rangle \psi$. By induction, δ and ψ are equivalent to $\check{\Delta} = \bigvee_i \Delta_i$ and $\bigvee_j \Psi_j$ respectively. We define a family Φ indexed by $\text{NRS}(I) \times J$ as follows:

$$\Phi_{\vec{i}, j} := \Gamma_{\alpha}^P(\Delta_{i_1}, \dots, \Delta_{i_n}; \Psi_j).$$

It remains to show that $\bigvee_{\vec{i},j} \Phi_{\vec{i},j}$ is equivalent to φ . The chain of reasoning for this is as follows:

$$\begin{aligned} \check{\Delta}\langle\alpha\rangle \bigvee_j \Psi_j &\Leftrightarrow \bigvee_j (\check{\Delta}\langle\alpha\rangle \Psi_j) \\ &\Leftrightarrow \bigvee_j \bigvee_{\vec{i} \in \text{NRS}(I)} \Gamma_{\alpha}^P(\Delta_{i_1}, \dots, \Delta_{i_n}; \Psi_j) \\ &\Leftrightarrow \bigvee_{\vec{i},j} \Gamma_{\alpha}^P(\Delta_{i_1}, \dots, \Delta_{i_n}; \Psi_j). \end{aligned}$$

As in the case of Theorem 1.10, the disjunction distributes over the modality when on the right. The next step holds by Lemma A.6, and finally we merge the two disjunctions.

To show the second claim of the theorem, note that if φ is positive, then so is δ , and hence each Δ_i is a PHMLU formula. We thus take $\Delta_i^+ = \Delta_i$ and $\Delta_i^- = \perp$. It follows that each $\Phi_{\vec{i},j}$ is equivalent to a PHMLU formula, as desired. $\square$

APPENDIX B. ALGORITHM AND EXAMPLE

We now describe the algorithm that computes a distinguishing PHMLU formula from a derivation of a directed apartness and the reverse algorithm that computes a derivation of $s \not\#_{db} t$ from a PHMLU formula φ for which $s \models \varphi$ and $t \not\models \varphi$. This is the algorithm implicit in the proof of Theorem 2.24 that ties together Lemma 2.22, describing derivations of $s \not\#_{db} t$, and Lemma 2.11, describing $s \models \varphi$.

From a derivation of $s \not\#_{db} t$ to a PHMLU formula. A derivation of $s \not\#_{db} t$ has as last rule

$$\frac{s \rightarrow_{\tau} s' \rightarrow_{\alpha} s'' \quad \forall t', t''. t \rightarrow_{\tau} t' \rightarrow_{(\alpha)} t'' \implies s' \not\#_{db} t' \vee s'' \not\#_{db} t''}{s \not\#_{db} t} \text{IN}_{db}$$

We consider the (finite) set of hypotheses of the rule, which are all of the form (1) $s' \not\#_{db} t'$ or (2) $s'' \not\#_{db} t''$ or (3) $t'' \not\#_{db} s''$ (where $t \rightarrow_{\tau} t' \rightarrow_{(\alpha)} t''$). For each hypothesis we have (recursively) a PHMLU formula that distinguishes the states and we collect them together: δ is the conjunction of the formulas that we obtain from hypotheses of the form (1), φ_1 is the conjunction of the formulas that we obtain from hypotheses of the form (2) and φ_2 is the disjunction of the formulas that we obtain from hypotheses of the form (3). Now take $\varphi := \delta\langle\alpha\rangle(\varphi_1 \wedge \neg\varphi_2)$. Then $s \models \varphi$ and $t \not\models \varphi$.

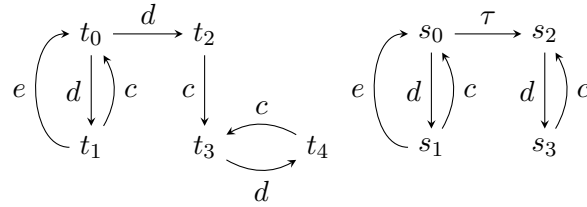
From a PHMLU formula distinguishing s and t to a derivation of $s \not\#_{db} t$. Given the PHMLU formula φ for which $s \models \varphi$ and $t \not\models \varphi$, we compute, recursively, a derivation of $s \not\#_{db} t$. The cases for $\varphi = \perp, \top$ cannot occur, and the cases for $\varphi = \varphi_1 \wedge \varphi_2$ or $\varphi = \varphi_1 \vee \varphi_2$ are dealt with by an immediate recursive call. (For the case of $\varphi = \varphi_1 \wedge \varphi_2$, we have $t \not\models \varphi_1 \wedge \varphi_2$, so $t \not\models \varphi_1$ or $t \not\models \varphi_2$ (possibly both). Say we have $t \not\models \varphi_1$. We also have $s \models \varphi_1$, so we take $\varphi := \varphi_1$.)

For the case of $\varphi := \delta\langle\alpha\rangle(\varphi_1 \wedge \neg\varphi_2)$, from $s \models \varphi$ we find s', s'' with $s \rightarrow_{\tau} s' \rightarrow_{(\alpha)} s''$ and $s' \models \delta$, $s'' \models \varphi_1$ and $s'' \not\models \varphi_2$. From $t \not\models \varphi$ we find that for all t', t'' with $t \rightarrow_{\tau} t' \rightarrow_{(\alpha)} t''$ we have $t' \not\models \delta$ or $t'' \not\models \varphi_1$ or $t'' \models \varphi_2$. We recursively have, for all t', t'' with $t \rightarrow_{\tau} t' \rightarrow_{(\alpha)} t''$, a

derivation of $s' \not\#_{db} t'$ or of $s'' \not\#_{db} t''$ or of $t'' \not\#_{db} s''$, so by rule (In_{db}) we have a derivation of $s \not\#_{db} t$.

We now give another slightly longer example of how a distinguishing formula can be computed from an apartness proof and vice versa. We give an example of the other direction then the one that was shown in Example 2.25, we show how a derivation of $t_0 \not\#_{db} s_0$ is computed from the distinguishing formula.

Example B.1. For the LTS given below we have $t_0 \not\#_{db} s_0$. We give a distinguishing formula and from that produce a derivation of $t_0 \not\#_{db} s_0$, following the proof of Theorem 2.24.



A distinguishing PHMLU-formula for t_0 and s_0 is

$$\varphi := (\langle d \rangle \langle e \rangle \top) \langle d \rangle (\neg \langle e \rangle \top).$$

From φ we can compute a derivation of $t_0 \not\#_{db} s_0$, following the algorithm that we have just described. For purposes of space, we single out a sub-derivation Σ of $t_0 \not\#_{db} s_2$. We indicate above the $\forall s', s'' \dots$ all the (finitely many) possible transitions that we need to prove a property for: two in the case of $s_0 \rightarrow_{\tau} s' \rightarrow_{(d)} s''$, and one in the case of $s_2 \rightarrow_{\tau} s' \rightarrow_{(d)} s''$. Similarly above the $\forall r', r'' \dots$, but there are no possible transitions, so there is nothing to prove, therefore $\checkmark$.

Note that, to compute the derivation from the formula, we need to know the actual reason why $t_0 \models \varphi$ and $s_0 \not\models \varphi$, because we construct the derivation from that information. For $t_0 \models \varphi$ we consider $t_0 \rightarrow_{(d)} t_2$, using which we create a derivation with two subderivations: one basically with conclusion $s_1 \not\#_{db} t_2$ (corresponding with formula $\neg \langle e \rangle \top$) and the other, Σ , with conclusion $t_0 \not\#_{db} s_2$ (corresponding with formula $\langle d \rangle \langle e \rangle \top$).

$$\begin{array}{c}
 \checkmark \\
 \hline
 s_1 \rightarrow_e s_0 \quad \forall r', r''. t_2 \rightarrow_{\tau} r' \rightarrow_{(e)} r'' \implies \dots \\
 \hline
 s_1 \not\#_{db} t_2 \\
 \hline
 t_2 \not\#_{db} s_1 \\
 \hline
 \begin{array}{cc}
 t_0 \not\#_{db} s_0 \vee t_2 \not\#_{db} s_1 & \Sigma \\
 \hline
 t_0 \not\#_{db} s_2 \vee t_2 \not\#_{db} s_3 \\
 \hline
 \forall s', s''. s_0 \rightarrow_{\tau} s' \rightarrow_{(d)} s'' \implies t_0 \not\#_{db} s' \vee t_2 \not\#_{db} s'' \\
 \hline
 t_0 \not\#_{db} s_0
 \end{array}
 \end{array}$$

$\Sigma :=$

$$\begin{array}{c}
 \checkmark \\
 \hline
 t_1 \rightarrow_e t_0 \quad \forall r', r''. s_3 \twoheadrightarrow_\tau r' \rightarrow_{(e)} r'' \implies \dots \\
 \hline
 t_1 \not\#_{db} s_3 \\
 \hline
 t_1 \not\#_{db} s_3 \\
 \hline
 t_0 \not\#_{db} s_2 \vee t_1 \not\#_{db} s_3 \\
 \hline
 t_0 \rightarrow_d t_1 \quad \forall s', s''. s_2 \twoheadrightarrow_\tau s' \rightarrow_{(d)} s'' \implies t_0 \not\#_{db} s' \vee t_1 \not\#_{db} s'' \\
 \hline
 t_0 \not\#_{db} s_2
 \end{array}$$