Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science — Postprint (2018) 1–21
TCS

Interleaving Isotactics – An Equivalence Notion on Behaviour Abstractions

Artem Polyvyanyy[a,1,*], Jan Sürmeli[b], Matthias Weidlich[c]

[a] *The University of Melbourne, Australia*
[b] *Technische Universität Berlin, Germany*
[c] *Humboldt-Universität zu Berlin, Germany*

Abstract

We study the equivalence of models that capture the behaviour of systems, such as process-oriented information systems. We focus on models that are not related by a bijection over their actions, but by an alignment between sets of their actions. For this setting, we propose *interleaving isotactics* as an equivalence notion based on abstractions that are induced by the alignment. We demonstrate that this notion is grounded in trace equivalence, provide a temporal logic characterisation of the properties it preserves, prove decidability of the respective verification problems, and present an implementation of a decision procedure for the equivalence notion.

Keywords: Behavioural equivalence, bisimulation, process semantics, behavioural abstraction, model matching

1. Introduction

The behaviour of a system is often described by a model that defines a set of actions and causal dependencies for their execution. Such models, also referred to as process models, have been widely adopted in the design and implementation of software systems in general [13] and process-oriented information systems in particular [22]. Various applications require an assessment of the equivalence of two models that describe a system's behaviour. As examples, we consider applications from the domain of process-oriented systems:

Variation Management: In large organisations, a single process, and thus a single type of information system, exists in many variations due to country-specific legal requirements, deviations in the IT infrastructure, or organisational differences [32]. Models that describe these variations specify the same functionality, yet rely on different sets of actions as basic building blocks. Ensuring that the models are free of behavioural contradictions is a major concern when managing process variations, see [28].

Implementation of Reference Models: For several domains, best-practices and well-established standard procedures have been published as reference models [10, 23]. In order to make effective use of reference models, their equivalence needs to be assessed with respect to models that have been tailored and customised

*Corresponding author

Email addresses: artem.polyvyanyy@unimelb.edu.au (Artem Polyvyanyy), jan.suermeli@tu-berlin.de (Jan Sürmeli), matthias.weidlich@hu-berlin.de (Matthias Weidlich)

[1] This work was initiated at the Queensland University of Technology, Brisbane, Australia.

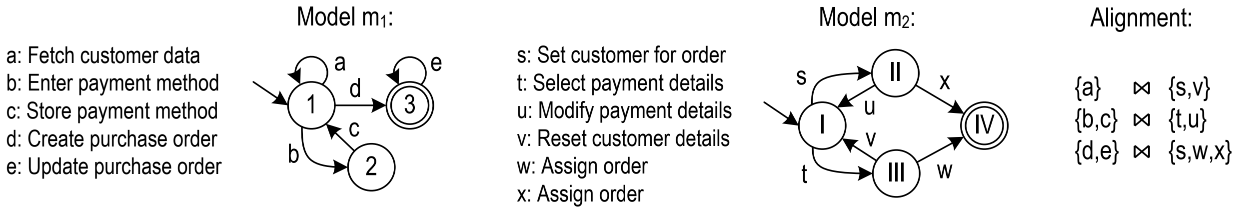

Figure 1. Two finite state machines modelling an excerpt of a business process and an alignment.

to a particular implementation environment. However, not all actions defined in a reference model may have a direct counterpart in the implementation.

Process Querying: Process querying studies methods for managing, e.g., retrieving or manipulating, repositories of models that describe observed and/or envisioned processes [17]. It involves correlation models that relate actions of different abstractions of a system to each other (e.g., actions of a reference model and their corresponding implementations). Behavioural equivalences can be used to induce correlation models.

The above applications have in common that the models, for which equivalence shall be assessed, typically assume different levels of abstraction. In that case, the semantic correspondence between the actions of such models cannot be captured by a bijection. Rather, actions are *grouped* in either model and the groups of one model are *related* to the groups of another model by means of a binary relation, called *alignment*.[2] Alignments are typically constructed using techniques for automated model matching [4, 29]. Note that the groups that are part of the alignment may overlap, i.e., a certain action can be interpreted in terms of different groups of actions in the other model. Hence, we do not narrow the scope to alignments that relate occurrences of actions to each other, see [20], since those are not discovered by matching techniques.

For illustration, we consider two finite state machines that depict a process to issue purchase orders, as implemented in two different systems, see Figure 1. Models m_1 and m_2 both contain actions related to the handling of customer data, the modification of payment details, as well as the actual order setup. Yet, when comparing the models, neither the actions nor their occurrences can be related by a bijection. For instance, the action *'a: Fetch customer data'* in model m_1 relates to two actions, *'s: Set customer for order'* and *'v: Reset customer details'* in model m_2. This relation cannot be traced back to established notions of hierarchical action refinements [5, 27]. Action s in model m_2 comprises functionality that does not only relate to the handling of customer data, but also includes the setup of the order. As such, it also relates to actions *'d: Create purchase order'* and *'e: Update purchase order'* in model m_1. These complex relations between the actions are captured in the alignment in Figure 1. The proposed alignment relates three sets of actions of either model to each other.

In this paper, we address the problem of assessing behavioural equivalence of two models in the context of a given alignment. We solve this problem by verifying equivalence based on abstractions, i.e., on the groups of actions induced by the alignment, rather than the individual actions.

Taking up the example in Figure 1, the alignment defines the abstractions for the verification of equivalence. The intuition of our approach is illustrated in Figure 2, where r_i is an exemplary run of machine m_i, for $i \in \{1, 2\}$. These runs induce traces w_1 and w_2 that contain sets of groups of actions, which can be seen as the interpretation of the runs under the given alignment. For instance, the occurrence of action s in run r_2 is assigned all the groups of actions containing s in the alignment definition, namely $\{s, v\}$ and $\{s, w, x\}$.

Based thereon, traces are compared by resolving each choice between groups of actions that have been assigned to action occurrences, while considering the maximal sequences of these groups. In the example, the first occurrence of action s in run r_2 can be resolved as $\{s, v\}$ (setting and resetting customer data), which, under $\{a\} \bowtie \{s, v\}$, is in line with the occurrence of a in r_1 (fetching customer data). Further, due to $\{b, c\} \bowtie \{t, u\}$, both occurrences of b and c in r_1 (entering and storing the payment method) correspond

[2]The term *alignment* used in this work is not to be confused with its use in process mining, where *business alignment* [24] refers to the practice of comparing the real behaviour of an information system or its users with the intended or expected behaviour and *alignment* refers to a concrete technique for quantifying differences between the observed and expected behaviour [25].

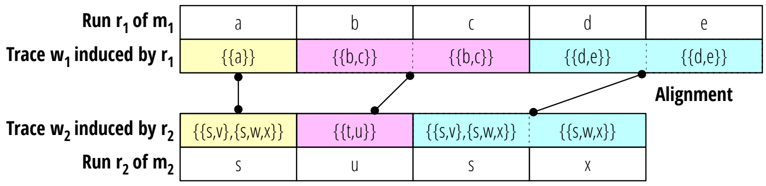

Figure 2. Two example runs of the state machines from Figure 1 and their traces induced by the alignment.

to the occurrence of u in r_2 (modifying payment details). Finally, $\{d, e\} \bowtie \{s, w, x\}$ enables the abstraction of occurrences of d and e in r_1 (creating and updating a purchase order) and relating them to the second occurrence of s followed by x in r_2 (setting the customer and assigning an order). Here, unlike the first occurrence of s that was resolved as $\{s, v\}$, the second occurrence is resolved as $\{s, w, x\}$.

This example illustrates that both runs r_1 and r_2 'mirror' the behaviour of each other under the given alignment: After handling the customer data, the payment details are set, before the purchase order is managed. The equivalence verification relies on the explicit resolution of action occurrences when the action is part of more than one group in the alignment. All the runs of a system must be mirrored by runs of the other system to yield a behavioural equivalence of the systems.

Related Notions of Equivalence. Behavioural equivalences have been studied for decades, yet, established notions, see [3, 26], are not applicable for the setting outlined above as they impose the assumption of a bijection between the actions of two models. Thus, the traditional setting of equivalence verification can be seen as a special case of the setting addressed here, requiring that the alignment is of a particular structure.

As illustrated by the example, we target a setting where differences in the number of occurrences are abstracted. Such situations have been addressed by stutter behavioural equivalences, which are variants of standard behavioural equivalences that allow actions to be mimicked by sequences of actions (rather than by single actions) [6]. In this way, an action in an abstract model can be modelled as a sequence of actions in a concrete model [1]. Stuttering equivalences are not applicable once the alignment relation defines overlapping groups of actions. As such, they are limited to alignments that are grounded in hierarchical refinements.

Recently, there have been several attempts to define behavioural equivalence for general alignments. Specific notions have been proposed for different semantics: interleaving, linear time semantics [30]; interleaving, branching time semantics [31]; and concurrent, linear time semantics [18]. However, all these existing notions are ad-hoc, in the sense that:

(1) It is not known whether these notions are proper generalisations of the well-established behavioural equivalences. That is, if an alignment collapses to a bijection between the actions of two models, it is not known whether the proposed notions coincide with some known equivalences.
(2) It is not known what kind of system properties these notions preserve. That is, if two models show one of the proposed equivalences, it is not known whether there is a class of properties that is preserved, e.g., in terms of logic formulae as presented in [14] for standard equivalences and in [12] for stutter equivalences.
(3) It is not known, if and under which constraints any of the proposed notions is decidable.

Contributions. To assess behavioural equivalence of aligned models, we propose the notion of *interleaving isotactics.* This notion is inspired by earlier ideas on concurrent, linear time isotactics [18] that, however, at this stage suffers from the aforementioned three open issues. The intuition of isotactics (from the Greek: ἴσος [isos] "equal", and τακτική [taktikí] "tactics", "policy", "behaviour") can be described as follows: Every occurrence of an action can be mimicked by an arbitrary number of occurrences of any subset of actions of some aligned group of actions, regardless of the structural relations of these actions in the model. Indeed, it seems more natural to reason on the level of occurrences of actions rather than their structural relations in models when reasoning about behavioural equivalence.

Concretely, this paper contributes:

C1: A grounding of isotactics in an established behavioural equivalence. Trace equivalence and isotactics coincide for simple alignments (bijections over singleton sets of actions) and repetition-free sets of runs.

C2: A Linear Temporal Logic (LTL) characterisation of the system properties preserved by isotactics. Given an alignment, we show that tactic-invariant LTL formulae are preserved, regardless of the actual systems.
C3: Decidability of the logic characterisation. Tactic-invariance of LTL-formulae is decidable.
C4: Decidability of the equivalence. Isotactics is decidable for finite state machines.
C5: An open-source implementation of a decision procedure for isotactics for finite state machines. This implementation is based on the arguments of the decidability proof and is publicly available.[3]

This paper proceeds with Section 2 that lists formal notions used to support the subsequent discussions. In Section 3, we formalise the alignment of (models of) systems and propose the notion of isotactics. The main results of this paper are summarised in Section 4. Sections 5 to 7 prove and discuss the main results. The paper closes with concluding remarks in Section 8. Detailed proofs of all the results of this paper can be found in Appendix A. Appendix B exemplifies the constructions used in one of the proofs of this paper.

2. Formal Framework

This section introduces formal notions that are used to support subsequent discussions.

Basic Notations. $\mathbb{N}$ denotes the set of all natural numbers including zero. Let $i, j \in \mathbb{N}$. By $\min(i, j)$ and $\max(i, j)$, we denote the minimum and maximum of i and j, respectively. Let A be a set. Then, by $\wp(A)$, $\wp_{>0}(A)$, and $\wp_{=1}(A)$, we denote the power set of A, $\wp(A) \setminus \{\varnothing\}$, and $\{x \in \wp_{>0}(A) \mid \exists\, a \in A : x = \{a\}\}$, respectively. Let B be a set, and $f : A \to B$ be a function. If $A' \subseteq A$, then by $f(A')$ we denote the set $\{b \in B \mid \exists\, a \in A' : f(a) = b\}$. Let R be a binary relation between A and B. Then, $R^{-1} := \{(b, a) \in B \times A \mid (a, b) \in R\}$ denotes the *inverse* of R. Let $\equiv$ be an equivalence relation on A. Then, by $\langle a \rangle_{\equiv}$, where $a \in A$, we denote the equivalence class of A by $\equiv$ that contains a. Moreover, by $A/\!\equiv$ we denote the set of all equivalence classes of A by $\equiv$, i.e., $A/\!\equiv := \{x \in \wp_{>0}(A) \mid \exists\, a \in A : \langle a \rangle_{\equiv} = x\}$. We write A^* to denote the set of all (finite) words over A, including the empty word ε. Let $w := a_1 \ldots a_n \in A^*$ be a word. Then, $w(k) := a_k$, where $k \in \{1, \ldots, n\}$, denotes the k-th character of w, and $|w| := n$ denotes the length of w. Let $k \in \{1, \ldots, |w|\}$, then $w[k\rangle$ denotes the suffix $w(k) \ldots w(n)$ of w starting from the k-th character of w. We call w *repetition-free* iff for all $k \in \{1, \ldots, n-1\}$: $w(k) \neq w(k+1)$. We call a set W of words repetition-free, if each of its members $w \in W$ is repetition-free. The concatenation of two words $w = a_1 \ldots a_n$ and $w' = a'_1 \ldots a'_m$ is defined as $ww' := a_1 \ldots a_n a'_1 \ldots a'_m$. For example, $\hat{w} = ababahalamaha$ is a repetition-free word that results from the concatenation of $w = ababa$ and $w' = halamaha$, i.e., $\hat{w} = ww'$.

Traces and Linear Temporal Logic. Let κ be a finite set. A κ*-trace* is an element of $\wp(\kappa)^*$, i.e., a finite sequence of subsets of κ. For $i \in \{1, 2\}$, let κ_i be a finite set, and let W_i be a set of κ_i-traces. W_1 and W_2 are *trace equivalent* up to a bijection b from κ_1 to κ_2, iff b induces an isomorphism between traces in W_1 and W_2.

Definition 2.1 (Trace Equivalence)**.** For $i \in \{1, 2\}$, let κ_i be a finite set, and let W_i be a set of κ_i-traces. Let b be a bijection from κ_1 to κ_2. Then, W_1 and W_2 are *trace equivalent* up to b iff there exists a bijection R from W_1 to W_2 such that for all $w_1 \in W_1$ it holds that (i) $|w_1| = |R(w_1)|$ and (ii) for all $i \in \{1, \ldots, |w_1|\}$ it holds that $b(w_1(i)) = R(w_1)(i)$. ⌟

By $\mathsf{LTL}[\kappa]$, we denote the set of all *LTL-Formulae* [1, 16] (without the *next* operator) over a set κ that are given by the following expression:

$$\varphi ::= \mathit{true} \mid K \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \,\mathsf{U}\, \psi_2, \quad \textit{where} \;\; K \in \kappa \;\; \textit{and} \;\; \psi, \psi_1, \psi_2 \in \mathsf{LTL}[\kappa].$$

Let $\varphi \in \mathsf{LTL}[\kappa]$ and w be a κ-trace. Then, w *satisfies* φ, written $w \vDash \varphi$, iff $\varphi = \mathit{true}$ or:

- $\varphi = K$, $K \in \kappa$, and $K \in w(1)$,
- $\varphi = \neg\psi$, and $w \nvDash \psi$,
- $\varphi = \psi_1 \vee \psi_2$, and there exists $i \in \{1, 2\}$: $w \vDash \psi_i$, or
- $\varphi = \psi_1 \,\mathsf{U}\, \psi_2$, and there exists $k \in \{1, \ldots, |w|\}$ with $w[k\rangle \vDash \psi_2$ and $w[i\rangle \vDash \psi_1$ for all $i \in \{1, \ldots, k-1\}$.

[3] https://github.com/Isotactics/deciding-isotactics

Other Boolean connectives, e.g., $\wedge$, $\Rightarrow$, and $\Leftrightarrow$, can be obtained from $\neg$ and $\vee$ in the standard way. As usual, the until operator (U) can be used to define temporal modalities F (*eventually*) and G (*globally*). In particular, $\mathsf{F}\ \psi \coloneqq \mathit{true}\ \mathsf{U}\ \psi$ and $\mathsf{G}\ \psi \coloneqq \neg\mathsf{F}\ \neg\psi$.

Finite State Machines. We rely on a common notion of finite state machines [9, 19]:

Definition 2.2 (Finite State Machine). A *finite state machine* (FSM) is a 5-tuple $S \coloneqq (\mathcal{Q}, \Lambda, \rightarrow, q^{ini}, F)$, where $\mathcal{Q}$ is a finite set of *states*, Λ is a set of *labels*, $\mathcal{Q}$ and Λ are disjoint, $\rightarrow \subseteq \mathcal{Q} \times \Lambda \times \mathcal{Q}$ is the *transition relation*, $q^{ini} \in \mathcal{Q}$ is the *initial state*, and $F \subseteq \mathcal{Q}$ is the set of *final states*. ⌟

We say that S is defined over Λ. By $p \xrightarrow{\lambda} q$, where $p, q \in \mathcal{Q}$ and $\lambda \in \Lambda$, we denote the fact that $(p, \lambda, q) \in \rightarrow$. We say that S *accepts* a word $w \in \Lambda^*$, iff (i) w is the empty word, i.e., $w = \varepsilon$, and $q^{ini} \in F$, or (ii) there exists a sequence $\rho \in \mathcal{Q}^*$ of states of length $|w| + 1$, such that $\rho(1) = q^{ini}$, $\rho(|w| + 1) \in F$, and for all $i \in \{1, \ldots, |w|\}$ it holds that $\rho(i) \xrightarrow{w(i)} \rho(i + 1)$; we refer to a word accepted by S as a *run* of S.

By $\mathcal{L}(S)$, we denote the set of all runs of S, i.e., the *language* of S. Two FSMs S_1 and S_2 are *language-equivalent* iff $\mathcal{L}(S_1) = \mathcal{L}(S_2)$. Two FSMs are illustrated in Figure 1, using common notation. For instance, FSM $m_1 \coloneqq (\mathcal{Q}, \Lambda, \rightarrow, q^{ini}, F)$ has three states, $\mathcal{Q} = \{1, 2, 3\}$, with the initial state $q^{ini} = 1$ and the set of final states $F = \{3\}$. Its set of labels is $\Lambda = \{a, b, c, d, e\}$ and its transition relation is defined by $1 \xrightarrow{a} 1$, $1 \xrightarrow{b} 2$, $2 \xrightarrow{c} 1$, $1 \xrightarrow{d} 3$, and $3 \xrightarrow{e} 3$.

3. Equivalence of Aligned Systems

We consider the behaviour of a system in terms of interleaving, linear time semantics, i.e., as a *set of runs*, where a *run* is a finite sequence over a set of *labels*. A run is produced by a system model, such as an FSM, Petri net, or Turing machine. Against the background of describing the behaviour of systems, a label often represents an action. However, other interpretations are possible: A label could represent a state, a state predicate, a configuration, or a similar concept. The basis for aligning system models comprises:

1. *Grouping* of labels. Labels that shall be considered jointly when comparing the behaviour of systems are part of a single group. The groups may overlap, i.e., a label may be a member of more than one group. Intuitively, this corresponds to different interpretations of a single label. For instance, in Figure 1, the action *'s: Set customer for order'* can be interpreted together with action *'v: Reset customer details'* as the sheer retrieval of customer data, whereas with actions *'w,x: Assign order'*, it relates to processing of a purchase order. One can 'eliminate' labels by not assigning them to any group. The respective labels are then not further considered in the verification of equivalence.
2. *Relating* the groups of one model with the groups of another model by means of a binary relation, called *alignment*. The alignment thereby defines a structural relation between groups of labels for which occurrences shall be considered as equivalent. Referring to Figure 1, as detailed above already, the functionality of action *'a: Fetch customer data'* in model m_1 is reflected in the functionality behind two actions, *'s: Set customer for order'* and *'v: Reset customer details'* in model m_2.

Based thereon, one can assess behavioural equivalence of two models by:

1. *Abstracting* each model based on the groups. The idea here is that the behaviour of a system is no longer considered in terms of runs over a set of labels, but in terms of runs over groups. For two runs of our running example, this abstraction is illustrated in Figure 2, in terms of the traces w_1 and w_2 that are induced by the alignment.
2. *Verifying* whether, according to the alignment, the abstract models are behaviourally equivalent. That is, it is checked whether two systems show the same behaviour once the latter is interpreted in terms of runs over groups. For the example in Figure 2, we see that indeed, both runs describe the same behaviour of first handling customer data, which is followed by the setup of payment details, before the management of the purchase order.

Based on the above intuition, this section develops the formalisation of the proposed equivalence notion: We first define groupings and alignments (Section 3.1). Based thereon, we clarify the notions needed to abstract a model based on groups (Section 3.2), which are then used to verify equivalence (Section 3.3).

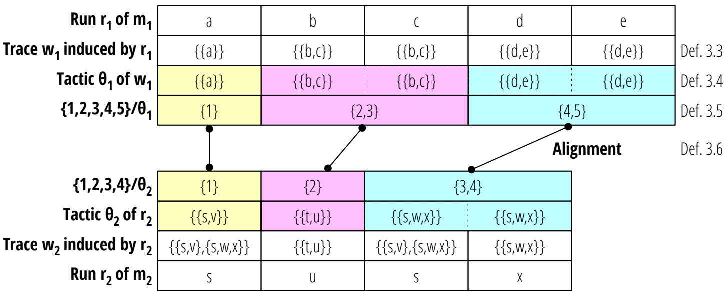

Figure 3. The main concepts illustrated using the runs from Figure 2: For $i \in \{1,2\}$, r_i is a run of m_i from Figure 1, w_i is the trace of r_i, and θ_i is a tactic of w_i. The traces w_1 and w_2 are aligned by tactics θ_1 and θ_2 because $\{a\} \bowtie \{s,v\}$, $\{b,c\} \bowtie \{t,u\}$ and $\{d,e\} \bowtie \{s,w,x\}$. Another tactic of w_2 is $\theta_2' = \{\{s,v\}\}\{\{t,u\}\}\{\{s,v\}\}\{\{s,w,x\}\}$. However, w_1 and w_2 are *not* aligned by θ_1 and θ_2', because the second $\{\{s,v\}\}$ in θ_2' does not have a counterpart in θ_1. In fact, θ_1 and θ_2 is the only combination of tactics aligning w_1 and w_2. Hence, to align w_1 and w_2, the action s has to be resolved once as $\{\{s,v\}\}$ and once as $\{\{s,w,x\}\}$.

Finally, we elaborate on how behavioural properties of systems, defined in LTL, shall be interpreted in the context of an alignment (Section 3.4).

3.1. Groupings and Alignments

This section defines a *grouping* of a set of labels and an *alignment* between sets of labels. Intuitively, a grouping is a collection of sets of labels, in which every set encodes a group of actions of a system that shall be considered together in some behavioural analysis. An alignment specifies correspondences between groups of actions of two systems, thereby establishing which groups of labels relate to the same functionality described by the system models.

Definition 3.1 (Grouping)**.** A *grouping* of a set of labels Λ is a set $\kappa \subseteq \wp_{>0}(\Lambda)$. A set $K \in \kappa$ is a κ-*group* of Λ. ⌟

Referring to the labels of the FSMs in Figure 1, e.g., $\gamma_1 := \{\{a\},\{b,c\},\{d,e\}\}$ and $\gamma_2 := \{\{s,v\},\{t,u\},\{s,w,x\}\}$ are groupings of the sets of labels $\Gamma_1 := \{a,b,c,d,e\}$ and $\Gamma_2 := \{s,t,u,v,w,x\}$, respectively. Given a grouping κ of Λ and a label $\lambda \in \Lambda$, by $\mathcal{G}_\kappa(\lambda)$ we denote the set of all κ-groups that contain λ, i.e., $\mathcal{G}_\kappa(\lambda) := \{K \in \kappa \mid \lambda \in K\}$. For instance, it holds that $\mathcal{G}_{\gamma_2}(s) = \{\{s,v\},\{s,w,x\}\}$, which in the example relates to the different interpretations of action *'s: Set customer for order'*, either in the context of retrieving the customer data or related to the processing of a purchase order.

Definition 3.2 (Alignment)**.** For $i \in \{1,2\}$, let Λ_i be a set of labels, and let $\kappa_i \subseteq \wp_{>0}(\Lambda_i)$ be a grouping of Λ_i. Then, an *alignment* between Λ_1 and Λ_2 is a relation $\bowtie \subseteq \kappa_1 \times \kappa_2$, written $\bowtie : \Lambda_1 \otimes \Lambda_2$, which relates the κ_1-groups of Λ_1 with the κ_2-groups of Λ_2.

Every alignment $\bowtie$ induces groupings $\overset{i}{\bowtie} := \{K_i \in \wp_{>0}(\Lambda_i) \mid K_1 \bowtie K_2\}$ of Λ_i, $i \in \{1,2\}$. ⌟

For example, $\{(\{a\},\{s,v\}),(\{b,c\},\{t,u\}),(\{d,e\},\{s,w,x\})\} \subseteq \gamma_1 \times \gamma_2$ is an alignment between Γ_1 and Γ_2 used in Figure 1; it induces groupings γ_1 and γ_2 proposed above. As mentioned above, this alignment specifies that action a in model m_1 corresponds to actions s and v in model m_2. Furthermore, according to this alignment, actions b and c correspond to actions t and u, and actions d and e correspond to actions s, w, and x.

3.2. Comparing Runs based on Tactics

This section proposes a method for comparing runs of systems based on their tactics induced by alignments. In the remainder of this section, we will illustrate the essential concepts using the initial example of the state machines in Figure 1 and their runs depicted in Figure 2. This example is extended in Figure 3, which also points to the respective definitions in this section.

Let Λ be a set of labels, and let κ be a grouping of Λ. Every run $\sigma \in \Lambda^*$ induces the κ-trace $w := \mathcal{G}_\kappa(\sigma(1))\ldots\mathcal{G}_\kappa(\sigma(|\sigma|))$. Then, the κ-induced trace of σ is a sequence of sets of κ-groups obtained from w by removing all its elements that are equal to $\varnothing$ without changing the order of the remaining elements.

Definition 3.3 (Induced Trace)**.** Let Λ be a set of labels, let κ be a grouping of Λ, and let $\mathcal{T}_\kappa : \Lambda^* \to \wp_{>0}(\kappa)^*$ be given by:

- $\mathcal{T}_\kappa(\varepsilon) := \varepsilon$.
- Let $\sigma \in \Lambda^*$, $\lambda \in \Lambda$. Then, $\mathcal{T}_\kappa(\sigma\lambda) := \begin{cases} \mathcal{T}_\kappa(\sigma)\mathcal{G}_\kappa(\lambda) & \text{if } \mathcal{G}_\kappa(\lambda) \neq \varnothing, \\ \mathcal{T}_\kappa(\sigma) & \text{otherwise.} \end{cases}$

Given a run $\sigma \in \Lambda^*$, $\mathcal{T}_\kappa(\sigma)$ is the *κ-induced trace* of σ. ⌟

As an example, we consider groupings of labels $\gamma_1 := \{\{a\},\{b,c\},\{d,e\}\}$ and $\gamma_2 := \{\{s,v\},\{t,u\},\{s,w,x\}\}$ as defined in Section 3.1. Also, let $r_1 := abcde$ be a run of model m_1, which describes that after fetching some customer data, a payment method is entered and stored, after which a purchase order is created and updated. Run $r_2 := susx$ of model m_2, in turn, describes a situation in which a customer of an order is set initially, after which the payment details are modified, so that the customer data of the order needs to be set again, before the order is eventually assigned. Then, $\{\{a\}\}\{\{b,c\}\}\{\{b,c\}\}\{\{d,e\}\}\{\{d,e\}\}$ is the γ_1-induced trace of r_1, while $\{\{s,v\},\{s,w,x\}\}\{\{t,u\}\}\{\{s,v\},\{s,w,x\}\}\{\{s,w,x\}\}$ is the γ_2-induced trace of r_2, see Figure 3.

We compare traces based on *tactics*. Intuitively, a tactic represents a specific interpretation of the run of a system in terms of specific groups. While an induced trace describes all possible interpretations of a run, a tactic requires to choose among them. Let κ be a grouping of a set of labels Λ, and let $w \in \wp_{>0}(\kappa)^*$ be a κ-trace. A *tactic* θ of w selects $K \in w(i)$ for each position i of w.

Definition 3.4 (Tactic)**.** A *tactic* of a κ-trace $w \in \wp_{>0}(\kappa)^*$, where κ is a finite set, is a κ-trace $\theta \in \wp_{=1}(\kappa)^*$ such that $|\theta| = |w|$ and $\theta(i) \subseteq w(i)$ for all $i \in \{1,\ldots,|w|\}$. ⌟

Continuing with our running example, the γ_1-induced trace $\mathcal{T}_{\gamma_1}(abcde)$ has only a single tactic, which is illustrated in Figure 3. The γ_2-induced trace $\mathcal{T}_{\gamma_2}(susx)$, in turn, has four tactics:

- $\theta_2' := \{\{s,v\}\}\{\{t,u\}\}\{\{s,v\}\}\{\{s,w,x\}\}$,
- $\theta_2 := \{\{s,v\}\}\{\{t,u\}\}\{\{s,w,x\}\}\{\{s,w,x\}\}$,
- $\theta_2'' := \{\{s,w,x\}\}\{\{t,u\}\}\{\{s,v\}\}\{\{s,w,x\}\}$, and
- $\theta_2''' := \{\{s,w,x\}\}\{\{t,u\}\}\{\{s,w,x\}\}\{\{s,w,x\}\}$.

The above tactics essentially stem from the different means to interpret the action *'s: Set customer for order'* in the original run, and their combinations. Tactic $\theta_2 := \{\{s,v\}\}\{\{t,u\}\}\{\{s,w,x\}\}\{\{s,w,x\}\}$ is illustrated in Figure 3.

Each tactic θ induces an equivalence relation $=_\theta$ and a strict partial order $<_\theta$ on the set $\{1,\ldots,|\theta|\}$ of indices of θ. Two indices are equivalent, if θ selects the same $K \in \kappa$ for them and all indices between them. Two indices are ordered if they are ordered inside θ but are not equivalent.

Definition 3.5 (Tactic-induced Relations)**.** Let θ be a tactic of a κ-trace $w \in \wp_{>0}(\kappa)^*$, where κ is a finite set. Then, $=_\theta \subseteq \{1,\ldots,|\theta|\} \times \{1,\ldots,|\theta|\}$ and $<_\theta \subseteq \{1,\ldots,|\theta|\} \times \{1,\ldots,|\theta|\}$ are defined as follows:

1. $i =_\theta j$ iff for all $k,l \in \{1,\ldots,|\theta|\}$ such that $\min(i,j) \leq k,\ell \leq \max(i,j)$ it holds that $\theta(k) = \theta(\ell)$.
2. $i <_\theta j$ iff $i < j$ and $i \neq_\theta j$. ⌟

We note that $=_\theta$ is an equivalence relation. We abbreviate $\langle i\rangle_{=_\theta}$ and $\{1,\ldots,|\theta|\}/{=_\theta}$ as $\langle i\rangle_\theta$ and $\{1,\ldots,|\theta|\}/_\theta$, respectively. Let $i,j \in \{1,\ldots,|\theta|\}$, such that $i <_\theta j$. Then, for all $i' \in \langle i\rangle_\theta$ and $j' \in \langle j\rangle_\theta$, it holds that $i' <_\theta j'$. Hence, one can lift $<_\theta$ from $\{1,\ldots,|\theta|\}$ to $\{1,\ldots,|\theta|\}/_\theta$: For all $i,j \in \{1,\ldots,|\theta|\}$, $\langle i\rangle_\theta <_\theta \langle j\rangle_\theta$ iff $i <_\theta j$.

For example, the tactic $\theta_2 := \{\{s,v\}\}\{\{t,u\}\}\{\{s,w,x\}\}\{\{s,w,x\}\}$ induces the equivalence relation $=_{\theta_2}$ given by $\{(1,1),(2,2),(3,3),(4,4),(3,4),(4,3)\}$; $=_{\theta_2}$ induces three equivalence classes: $\{1\}$, $\{2\}$, and $\{3,4\}$. Moreover, $<_{\theta_2}$ is given by $\{(1,2),(1,3),(1,4),(2,3),(2,4)\}$. Thus, it holds that $\{1\} <_{\theta_2} \{2\} <_{\theta_2} \{3,4\}$.

Let $\bowtie : \Lambda_1 \otimes \Lambda_2$ be an alignment. For $i \in \{1,2\}$, let $w_i \in \wp(\overset{i}{\bowtie})^*$. Intuitively, w_1 and w_2 are *aligned* if there exist tactics θ_1 and θ_2 of w_1 and w_2, respectively, that can be aligned.

Definition 3.6 (Alignment of Tactics)**.** Let $\bowtie : \Lambda_1 \otimes \Lambda_2$ be an alignment. For $i \in \{1,2\}$, let $w_i \in \wp(\overset{i}{\bowtie})^*$, and let θ_i be a tactic of w_i. Tactics θ_1 and θ_2 are *aligned* by $\bowtie$, denoted by $\theta_1 \bowtie \theta_2$, iff there exists a bijection b from $\{1,\ldots,|\theta_1|\}/_{\theta_1}$ to $\{1,\ldots,|\theta_2|\}/_{\theta_2}$ such that:

1. for all $i_1 \in \{1, \ldots, |\theta_1|\}$ and $i_2 \in \{1, \ldots, |\theta_2|\}$ it holds that $b(\langle i_1 \rangle_{\theta_1}) = \langle i_2 \rangle_{\theta_2}$ implies $\theta_1(i_1) \bowtie \theta_2(i_2)$, and
2. for all $i, j \in \{1, \ldots, |\theta_1|\}$ it holds that $\langle i \rangle_{\theta_1} <_{\theta_1} \langle j \rangle_{\theta_1}$ implies $b(\langle i \rangle_{\theta_1}) <_{\theta_2} b(\langle j \rangle_{\theta_1})$. ⌟

We say that w_1 and w_2 are *aligned* by θ_1 and θ_2 w.r.t. $\bowtie$, and denote this by $(\theta_1, \theta_2) : w_1 \bowtie w_2$.

To illustrate the alignment of tactics, we again refer to Figure 3. Here, the γ_1-induced trace $\mathcal{T}_{\gamma_1}(abcde)$ and the γ_2-induced trace $\mathcal{T}_{\gamma_2}(susx)$ are aligned by tactics $\theta_1 := \{\{a\}\}\{\{b,c\}\}\{\{b,c\}\}\{\{d,e\}\}\{\{d,e\}\}$ and $\theta_2 := \{\{s,v\}\}\{\{t,u\}\}\{\{s,w,x\}\}\{\{s,w,x\}\}$ w.r.t. the alignment from Figure 1; one can construct bijection $b := \{\{1\} \mapsto \{1\}, \{2,3\} \mapsto \{2\}, \{4,5\} \mapsto \{3,4\}\}$ from $\{1, \ldots, |\theta_1|\}/_{\theta_1}$ to $\{1, \ldots, |\theta_2|\}/_{\theta_2}$. This bijection formally encodes the intuition given earlier: In either system, customer data is handled first, before payment details are set, which is followed by the management of the purchase order.

In contrast to the above example, $\mathcal{T}_{\gamma_1}(ade) = \{\{a\}\}\{\{d,e\}\}\{\{d,e\}\}$ and $\mathcal{T}_{\gamma_2}(susx)$ cannot be aligned. Run ade represents the situation that handling of customer data is immediately followed by the management of the purchase order. Unlike run $susx$, run ade thus does not include any actions related to the setup of the payment details. Finally, illustrating that alignments abstract from cardinalities, for every $n \in \mathbb{N}$ it holds that $\mathcal{T}_{\gamma_1}(\underbrace{bc \ldots bc}_{n \text{ times}} d)$ and $\mathcal{T}_{\gamma_2}(tw)$ can be aligned. Intuitively, this means that entering and storing the payment method repeatedly according to model m_1 is equivalent to a single selection of payment details in model m_2.

3.3. Tactic Coverage and Isotactics

We propose to compare two collections of traces based on *tactic coverage* and *isotactics* relations. The idea is that a collection of traces is "covered" by another collection of traces if for every trace in this collection one can find a trace in the other collection such that the tactics of these two traces are aligned.

Definition 3.7 (Tactic Coverage and Isotactics)**.**
Let $\bowtie : \Lambda_1 \otimes \Lambda_2$ be an alignment. For $i \in \{1, 2\}$, let W_i be a set of $\dot{\bowtie}$-traces.

1. W_1 and W_2 are in the *(interleaving) tactic coverage relation* w.r.t. $\bowtie$, denoted by $W_1 \leqslant_{\bowtie} W_2$, iff for every trace $w_1 \in W_1$ there exists a trace $w_2 \in W_2$ such that there exist tactics θ_1 and θ_2 of w_1 and w_2, respectively, such that $(\theta_1, \theta_2) : w_1 \bowtie w_2$.
2. W_1 and W_2 are in the *(interleaving) isotactics relation* w.r.t. $\bowtie$, denoted by $W_1 \div_{\bowtie} W_2$, iff $W_1 \leqslant_{\bowtie} W_2$ and $W_2 \leqslant_{\bowtie^{-1}} W_1$. ⌟

One can compare runs based on their induced traces, i.e., based on their abstractions induced by the alignment. For $i \in \{1, 2\}$, let Λ_i be a set of labels, let $\Sigma_i \subseteq \Lambda_i^*$ be a set of runs, and let $\bowtie : \Lambda_1 \otimes \Lambda_2$ be an alignment. We write $\Sigma_1 \leqslant_{\bowtie} \Sigma_2$ and $\Sigma_1 \div_{\bowtie} \Sigma_2$ to denote the facts that $\mathcal{T}_{\bowtie}^{1}(\Sigma_1) \leqslant_{\bowtie} \mathcal{T}_{\bowtie}^{2}(\Sigma_2)$ and $\mathcal{T}_{\bowtie}^{1}(\Sigma_1) \div_{\bowtie} \mathcal{T}_{\bowtie}^{2}(\Sigma_2)$, respectively.

For our running example in Figure 1, one can verify that for every γ_1-induced trace w_1 of m_1, there exists some γ_2-induced trace w_2 of m_2 such that some tactics of w_1 and w_2 are aligned, and vice versa; this fact is justified in Section 7. Note that groupings γ_1 and γ_2 are proposed in Section 3.1 and are the groupings used to define alignment $\bowtie$ in Figure 1. Hence, it holds that $\mathcal{L}(m_1) \div_{\bowtie} \mathcal{L}(m_2)$ and we say that m_1 and m_2 are *isotactic* w.r.t. $\bowtie$.

3.4. Alignments and LTL

To later explore the class of properties preserved by isotactics, we first need to review how such properties shall be interpreted in the presence of groupings and alignments. Considering properties formalised in LTL, there is a class of formulae that is of particular interest: Those for which the choice of tactics does not affect its truth value, called tactic-invariant LTL formulae. Below, we provide a formal characterisation of these formulae and how they are interpreted in the context of an alignment. This provides the basis to later show that tactic-invariant LTL formulae are indeed preserved by isotactics.

Let Λ be a set of labels, let $\kappa \subseteq \wp_{>0}(\Lambda)$ be a grouping, let $\Sigma \subseteq \Lambda^*$, let $\sigma \in \Sigma$, and let $\varphi \in \mathsf{LTL}[\kappa]$. Then, σ *satisfies* φ w.r.t. κ, denoted by $(\sigma, \kappa) \models \varphi$, iff $\mathcal{T}_{\kappa}(\sigma) \models \varphi$. For example, given run $\sigma := tvsx$ of model m_2 in Figure 1, it holds that $\{\{t,u\}\}\{\{s,v\}\}\{\{s,v\},\{s,w,x\}\}\{\{s,w,x\}\}$ is the γ_2-induced trace of σ, where γ_2 is defined in Section 3.1. Then, for sample formula $\varphi := (\{s,v\} \vee \{t,u\}) \, \mathsf{U} \, \{s,w,x\}$, we have $(\sigma, \gamma_2) \models \varphi$. Similarly, Σ *satisfies* φ w.r.t. κ, denoted by $(\Sigma, \kappa) \models \varphi$, iff for every $\sigma \in \Sigma$ it holds that $(\sigma, \kappa) \models \varphi$.

Each tactic θ of a κ-trace w is a κ-trace. Thus, every formula $\varphi \in \mathsf{LTL}[\kappa]$ can be evaluated over both w and θ. We call formula $\varphi \in \mathsf{LTL}[\kappa]$ *tactic-invariant*, if for every trace w it holds that the truth value of φ over w and over every tactic θ of w is the same.

Definition 3.8 (Tactic-Invariant LTL-Formula)**.** Let Λ be a set of labels and let $\kappa \subseteq \wp_{>0}(\Lambda)$ be a grouping of Λ. An LTL-formula $\varphi \in \mathsf{LTL}[\kappa]$ is *tactic-invariant* w.r.t. Λ and κ, iff for every $w \in \mathcal{T}_\kappa(\Lambda^*)$ and every tactic θ of w it holds that $w \vDash \varphi$ iff $\theta \vDash \varphi$. ⌟

For instance, consider the alignment $\bowtie$ from Figure 1, and these three $\mathsf{LTL}[\overset{2}{\bowtie}]$-formulae:

$$\varphi_1 := \{t,u\} \,\mathsf{U}\, (\{s,v\} \vee \{s,w,x\}), \quad \varphi_2 := \neg\{s,v\} \vee \neg\{s,w,x\}, \quad \text{and} \quad \varphi_3 := \neg\varphi_2.$$

Clearly, φ_1 is tactic-invariant: The satisfying $\overset{2}{\bowtie}$-induced traces are of the form $w_1w_2w_3$ where w_1 is an element in $\{\{\{t,u\}\}\}^*$, e.g., $w_1 = \varepsilon$ or $w_1 = \{\{t,u\}\}\{\{t,u\}\}$, w_2 is a non-empty subset of $\{\{s,v\},\{s,w,x\}\}$, and w_3 is an arbitrary $\overset{2}{\bowtie}$-induced trace. The tactics of such a trace $w_1w_2w_3$ are of the form $\theta_1\theta_2\theta_3$, where $\theta_1 = w_1$ (because w_1 is a sequence of singleton sets), θ_2 is either $\{\{s,v\}\}$ or $\{\{s,w,x\}\}$, and θ_3 is any tactic of w_3. Because θ_1 is a sequence of $\{\{t,u\}\}$ and θ_2 satisfies $\{s,v\} \vee \{s,w,x\}$, $\theta_1\theta_2\theta_3$ satisfies φ_1. Similarly, we argue that every φ-satisfying tactic of an arbitrary $\overset{2}{\bowtie}$-induced trace w is of the form $\theta_1\theta_2\theta_3$, and we can show that w is then also of the form $w_1w_2w_3$ as above, yielding satisfaction of φ_1. In contrast to that, φ_2 and φ_3 are not tactic-invariant: As proof, we take the $\overset{2}{\bowtie}$-induced trace $w = \{\{s,v\},\{s,w,x\}\}$ and tactic $\theta = \{\{s,v\}\}$ of w. Then, $w \nvDash \varphi_2$ but $\theta \vDash \varphi_2$, and $w \vDash \varphi_3$ and $\theta \nvDash \varphi_3$.

Given an alignment $\bowtie : \Lambda_1 \otimes \Lambda_2$ that relates the κ_1-groups of Λ_1 with the κ_2-groups of Λ_2, every formula $\varphi_1 \in \mathsf{LTL}[\kappa_1]$ is *aligned* to a similar formula $\varphi_2 \in \mathsf{LTL}[\kappa_2]$. Formulae φ_1 and φ_2 are aligned if φ_1 and φ_2 have the same structure, and φ_2 replaces each disjunction of atomic propositions from φ_1 with an aligned disjunction of atomic propositions. Intuitively, two aligned formulae define the same property (modulo the alignment).

Definition 3.9 (Alignment of LTL-Formulae)**.** Let $\bowtie$ be an alignment. LTL-Formulae $\varphi_1 \in \mathsf{LTL}[\overset{1}{\bowtie}]$ and $\varphi_2 \in \mathsf{LTL}[\overset{2}{\bowtie}]$ are *aligned* by $\bowtie$, denoted by $\varphi_1 \bowtie \varphi_2$, iff at least one of the following holds:

- For $i \in \{1,2\}$, $\varphi_i = \bigvee_{K_i \in \mathcal{K}_i} K_i$, where $\mathcal{K}_i \subseteq \overset{i}{\bowtie}$ and $(\mathcal{K}_1 \times \overset{2}{\bowtie}) \cap \bowtie = (\overset{1}{\bowtie} \times \mathcal{K}_2) \cap \bowtie$.
- For $i \in \{1,2\}$, $\varphi_i = \neg\psi_i$ and $\psi_1 \bowtie \psi_2$.
- For $i \in \{1,2\}$, $\varphi_i = \psi_i * \psi_i'$, where $* \in \{\vee, \mathsf{U}\}$, $\psi_1 \bowtie \psi_2$, and $\psi_1' \bowtie \psi_2'$. ⌟

For example, given formulae $\varphi_1 := (\{a\} \vee \{b,c\}) \,\mathsf{U}\, \{d,e\}$ and $\varphi_2 := (\{s,v\} \vee \{t,u\}) \,\mathsf{U}\, \{s,w,x\}$, it holds that $\varphi_1 \bowtie \varphi_2$, where $\bowtie$ is the alignment from Figure 1.

4. Main Results

This section summarizes the main results of this paper. For each proposed formal statement, we refer to the Lemmata that proves the statement.

First, we turn to one of the fundamental questions regarding an isotactics notion: is it a proper generalisation of the well-established behavioural equivalences? That is, once an alignment collapses to a bijection between the labels of two sets of runs, isotactics shall be grounded in a well-established notion of behavioural equivalence. This is indeed the case, as the following theorem demonstrates a relation between isotactics and trace equivalence—a widely established notion of behavioural equivalence—for repetition-free sets of runs.

Theorem 4.1 (Trace equivalence and isotactics for simple alignments and repetition-free runs)**.** *For $i \in \{1,2\}$, let Λ_i be a set of labels and let $\Sigma_i \subseteq \Lambda_i^*$ be repetition-free. Let b be a bijection from Λ_1 to Λ_2, and let $\bowtie := \{(\{\lambda\},\{b(\lambda)\}) \mid \lambda \in \Lambda_1\}$. Then, the following statements are equivalent:*

1. *$\mathcal{T}_{\overset{1}{\bowtie}}(\Sigma_1)$ and $\mathcal{T}_{\overset{2}{\bowtie}}(\Sigma_2)$ are trace equivalent up to b.*
2. *$\Sigma_1 \doteqdot_{\bowtie} \Sigma_2$.* ⌟

Proof. Follows from Lemmata 5.1 and 5.2. ∎

Second, we focus on the question of which system properties are preserved by isotactics. An answer to this question is given for the aforementioned class of tactic-invariant LTL-formulae. Once the choice of a tactic does not affect the truth value of a formula, it is indeed preserved by the proposed notion of equivalence.

Theorem 4.2 (Tactic-invariant LTL-formulae are preserved). *Let* $\bowtie : \Lambda_1 \otimes \Lambda_2$ *be an alignment. For* $i \in \{1, 2\}$, *let* $\Sigma_i \subseteq \Lambda_i^*$ *and let* $\varphi_i \in \mathsf{LTL}[\overset{i}{\bowtie}]$ *be tactic-invariant w.r.t.* Λ_i *and* $\overset{i}{\bowtie}$. *Let* $\varphi_1 \bowtie \varphi_2$ *and let* $\Sigma_1 \doteqdot_{\bowtie} \Sigma_2$. *Then, it holds that* $(\Sigma_1, \overset{1}{\bowtie}) \vDash \varphi_1$ *iff* $(\Sigma_2, \overset{2}{\bowtie}) \vDash \varphi_2$. ⌟

Proof. Follows from Definition 3.7 and Lemma 6.1. ∎

Theorem 4.2 implies that two isotactic (w.r.t. an alignment) FSMs have the same tactic-invariant properties (modulo the alignment). In order to use this result, it must be decidable whether a given LTL-formula is tactic-invariant. The next theorem establishes that this is indeed the case.

Theorem 4.3 (Tactic-invariance of LTL-formula is decidable). *Let* Λ *be a set of labels, let* $\kappa \subseteq \wp_{>0}(\Lambda)$ *be a grouping, and let* $\varphi \in \mathsf{LTL}[\kappa]$. *Then, the following problem is decidable: To decide whether* φ *is tactic-invariant w.r.t.* Λ *and* κ. ⌟

Proof. Follows from Lemmata 6.2 and 6.3. ∎

Finally, given two FSMs, it is decidable whether their languages are in the interleaving isotactics relation, i.e., it is decidable whether the FSMs are isotactic.

Theorem 4.4 (Isotactics is decidable for FSMs). *For* $i \in \{1, 2\}$, *let* Λ_i *be a set of labels and let* S_i *be an FSM over* Λ_i. *Let* $\bowtie : \Lambda_1 \otimes \Lambda_2$ *be an alignment. Then, the following problem is decidable: To decide whether* $\mathcal{L}(S_1)$ *and* $\mathcal{L}(S_2)$ *are in the interleaving isotactics relation w.r.t.* $\bowtie$, *i.e., to decide whether it holds that* $\mathcal{L}(S_1) \doteqdot_{\bowtie} \mathcal{L}(S_2)$. ⌟

Proof. Follows from Lemmata 7.8 and 7.9. ∎

This paper focuses on the definition of isotactics, its preserved properties, and the decidability of the respective verification problem. However, the constructions presented to prove Theorem 4.4 also reveal a decision procedure for isotactics. We implemented this procedure in an open-source tool, which enables a first practical application of isotactics.

5. Conditional Coincidence of Trace Equivalence and Isotactics

This section lists two statements that correspond to the two directions of the equivalence stated in Theorem 4.1. These statements justify the proposed conditional coincidence of trace equivalence and isotactics. Let $\bowtie : \Lambda_1 \otimes \Lambda_2$ be an alignment such that $\bowtie$ is a bijection from the singletons over Λ_1 to the singletons over Λ_2. In the proofs, we exploit the fact that every trace $w \coloneqq \mathcal{T}_{\overset{i}{\bowtie}}(\sigma)$, where $i \in \{1, 2\}$ and $\sigma \in \Lambda_i^*$, is a sequence of singletons and, thus, has exactly one tactic, namely w.

First, we show that trace equivalence implies isotactics.

Lemma 5.1. *For* $i \in \{1, 2\}$, *let* Λ_i *be a set of labels and let* $\Sigma_i \subseteq \Lambda_i^*$. *Let* b *be a bijection from* Λ_1 *to* Λ_2, *and let* $\bowtie \coloneqq \{(\{\lambda\}, \{b(\lambda)\}) \mid \lambda \in \Lambda_1\}$. *If* $\mathcal{T}_{\overset{1}{\bowtie}}(\Sigma_1)$ *and* $\mathcal{T}_{\overset{2}{\bowtie}}(\Sigma_2)$ *are trace equivalent up to* b, *then* $\Sigma_1 \doteqdot_{\bowtie} \Sigma_2$. ⌟

Second, we demonstrate the converse of Lemma 5.1 for the case when Σ_1 and Σ_2 are repetition-free. If Σ_1 and Σ_2 are repetition-free, then the traces induced by the runs in Σ_1 and Σ_2 are repetition-free as well. Consequently, every equivalence class of an equivalence relation induced by a tactic of any of the induced traces is a singleton. Our proof of the next lemma draws on this observation.

Lemma 5.2. *For* $i \in \{1, 2\}$, *let* Λ_i *be a set of labels and let* $\Sigma_i \subseteq \Lambda_i^*$ *be repetition-free. Let* b *be a bijection from* Λ_1 *to* Λ_2, *and let* $\bowtie \coloneqq \{(\{\lambda\}, \{b(\lambda)\}) \mid \lambda \in \Lambda_1\}$. *If* $\Sigma_1 \doteqdot_{\bowtie} \Sigma_2$, *then* $\mathcal{T}_{\overset{1}{\bowtie}}(\Sigma_1)$ *and* $\mathcal{T}_{\overset{2}{\bowtie}}(\Sigma_2)$ *are trace equivalent up to* b. ⌟

The proofs of Lemmata 5.1 and 5.2 are in Appendix A.

6. Property Preservation

Two isotactic collections of traces enjoy the same tactic-invariant LTL properties, cf. Theorem 4.2, while the problem of checking whether a given LTL-formula is tactic-invariant is decidable, cf. Theorem 4.3. These results are due to the lemmata proposed in this section; the proofs of all the statements proposed below are in Appendix A. Next, we argue that for any two aligned traces w_1 and w_2, it holds that $w_1 \models \varphi_1$ iff $w_2 \models \varphi_2$, where φ_1 and φ_2 are aligned tactic-invariant LTL-formulae.

Lemma 6.1. *Let $\bowtie : \Lambda_1 \otimes \Lambda_2$ be an alignment. For $i \in \{1,2\}$, let $\sigma_i \in \Lambda_i^*$, $w_i := \mathcal{T}_{\overset{i}{\bowtie}}(\sigma_i)$, θ_i be a tactic of w_i, and let $\varphi_i \in \mathsf{LTL}[\overset{i}{\bowtie}]$ be tactic-invariant w.r.t. Λ_i and $\overset{i}{\bowtie}$. Let φ_1 and φ_2 be aligned by $\bowtie$, i.e., $\varphi_1 \bowtie \varphi_2$, and let $(\theta_1, \theta_2) : w_1 \bowtie w_2$. Then, it holds that $w_1 \models \varphi_1$ iff $w_2 \models \varphi_2$.* ⌟

Consider alignment $\bowtie$ from Figure 1, $\mathsf{LTL}[\overset{1}{\bowtie}]$-formulae ψ_1, ψ_2, and ψ_3, and $\mathsf{LTL}[\overset{2}{\bowtie}]$-formulae φ_1, φ_2, and φ_3:

$$\psi_1 := \{b,c\} \,\mathsf{U}\, (\{a\} \vee \{d,e\}), \qquad \psi_2 := \neg\{a\} \vee \neg\{d,e\}, \qquad \psi_3 := \neg\psi_2,$$
$$\varphi_1 := \{t,u\} \,\mathsf{U}\, (\{s,v\} \vee \{s,w,x\}), \qquad \varphi_2 := \neg\{s,v\} \vee \neg\{s,w,x\}, \qquad \varphi_3 := \neg\varphi_2.$$

For all $i \in \{1,2,3\}$, ψ_i and φ_i are aligned by $\bowtie$, i.e., it holds that $\psi_i \bowtie \varphi_i$. It is easy to see that ψ_1, ψ_2, and ψ_3 are tactic-invariant; note that groups in $\overset{1}{\bowtie}$ are pairwise disjoint. Recall from Section 3.4 that φ_1 is tactic-invariant. Therefore, ψ_1 and φ_1 are 'preserved' for the aligned traces. Let w_1 be a $\overset{1}{\bowtie}$-induced trace of the form $\{\{b,c\}\}^*(\{\{a\}\}|\{\{d,e\}\})$. Then, it holds that $w_1 \models \psi_1$. Let w_2 be a $\overset{2}{\bowtie}$-induced trace, such that w_1 and w_2 can be aligned w.r.t. $\bowtie$. Then, w_2 is of the form $\{\{t,u\}\}^*(\{\{s,v\}\}|\{\{s,w,x\}\}|\{\{s,v\},\{s,w,x\}\})^+$, and, thus, it holds that $w_2 \models \varphi_1$. Also, recall from Section 3.4 that φ_2 and φ_3 are not tactic-invariant. Next, consider the $\overset{1}{\bowtie}$-induced trace $w_1' := \{\{a\}\}$ and the $\overset{2}{\bowtie}$-induced trace $w_2' := \{\{s,v\},\{s,w,x\}\}$. Clearly, traces w_1' and w_2' can be aligned w.r.t. $\bowtie$. However, it holds that $w_1' \models \psi_2$, $w_2' \not\models \varphi_2$, $w_1' \not\models \psi_3$, and $w_2' \models \varphi_3$.

We reduce the problem of deciding tactic-invariance of a given LTL-formula to the problem of checking language equivalence. To this end, we exploit the fact that every LTL-formula $\varphi \in \mathsf{LTL}[\kappa]$, where $\kappa \subseteq \wp_{>0}(\Lambda)$ is a grouping, can be translated into a corresponding FSM S_φ. Then, one should intersect the language of S_φ with the sets of traces and tactics over κ, respectively, and compare the results.

For the remainder of this section, we fix a grouping κ and an LTL-formula $\varphi \in \mathsf{LTL}[\kappa]$.

For some κ-trace w, we write Tactics(w) to denote the set of all tactics of w. Moreover, we introduce the helper sets $W := \{w \in \mathcal{T}_\kappa(\Lambda^*) \mid w \models \varphi\}$ and $\Theta := \{\theta \in \text{Tactics}(w) \mid w \in W, \theta \models \varphi\}$, denoting the sets of all φ-satisfying κ-induced traces, and φ-satisfying tactics of κ-induced traces in W, respectively. A tactic does not carry sufficient information about its 'origin'; there can exist two κ-induced traces w, w' with common tactics, i.e., Tactics$(w) \cap$ Tactics$(w') \neq \emptyset$. For instance, the $\overset{2}{\bowtie}$-induced traces $w := \{\{t,u\}\}\{\{s,v\},\{s,w,x\}\}$ and $w' := \{\{t,u\}\}\{\{s,v\}\}$ have the common tactic $\theta = w'$. Therefore, for each tactic $\theta \in$ Tactics(w) of a κ-induced trace w, we define the sequence $\theta_w \in (\wp_{=1}(\kappa) \times \wp_{>0}(\kappa))^*$ of pairs of sets of groups by $\theta_w(i) := (\theta(i), w(i))$, $i \in \{1,\ldots,|w|\}$. That is, each element of θ_w at position i refers to the group chosen by tactic θ and the set of groups this group has been chosen from. For instance, for the $\overset{2}{\bowtie}$-induced trace $w := \{\{t,u\}\}\{\{s,v\},\{s,w,x\}\}$ and its tactic $\theta := \{\{t,u\}\}\{\{s,v\}\}$, it holds that $\theta_w = (\{\{t,u\}\},\{\{t,u\}\})(\{\{s,v\}\},\{\{s,v\},\{s,w,x\}\})$. We refer to θ_w as the enriched version of tactic θ.

Based on the notion of θ_w, we define the sets $\hat{\Theta} := \{\theta_w \mid w \in W, \theta \in \text{Tactics}(w)\}$ and $\hat{\Theta}' := \{\theta_w \mid \theta \in \Theta, w \in \mathcal{T}_\kappa(\Lambda^*)\}$ built from W and Θ, respectively. That is, $\hat{\Theta}$ contains the enriched versions of all the tactics of φ-satisfying traces, and $\hat{\Theta}'$ contains the enriched versions of all the φ-satisfying tactics of arbitrary traces.

Next, we demonstrate that tactic-invariance coincides with equality of $\hat{\Theta}$ and $\hat{\Theta}'$: If φ is tactic-invariant, then for every trace w and tactic $\theta \in$ Tactics(w), it holds that $w \in W$ iff $\theta \in \Theta$. Otherwise, there exist a trace w and a tactic $\theta \in$ Tactics(w) such that $w \in W$ iff $\theta \notin \Theta$.

Lemma 6.2. *$\hat{\Theta} = \hat{\Theta}'$ iff φ is tactic-invariant w.r.t. Λ and κ.* ⌟

For example, consider the $\overset{2}{\bowtie}$-induced trace $w := \{\{s,v\},\{s,w,x\}\}$ and tactic $\theta := \{\{s,v\}\}$ of w, yielding $\theta_w = (\{\{s,v\}\},\{\{s,v\},\{s,w,x\}\})$. As mentioned above, $w \not\models \varphi_2$ and $w \models \varphi_3$; note that $w = w_2'$. But, it holds that $\theta \models \varphi_2$ and $\theta \not\models \varphi_3$. Assuming that $\varphi = \varphi_2$, it holds that $\theta_w \notin \hat{\Theta}$; note that $w \notin W$. However, it holds that

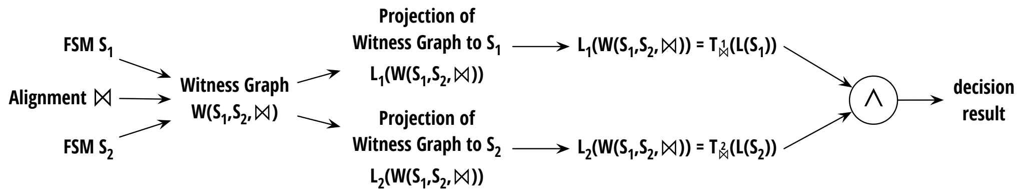

Figure 4. A schematic view of the proposed approach for deciding interleaving isotactics.

$\theta_w \in \hat{\Theta}'$, because $\theta \models \varphi_2$. Similarly, assuming that$\varphi = \varphi_3$, one can check that $\theta_w \in \hat{\Theta}$ and $\theta_w \notin \hat{\Theta}'$. According to Lemma 6.2, these facts confirm that φ_2 and φ_3 are not tactic-invariant.

We reduce the decision of whether $\hat{\Theta} = \hat{\Theta}'$ to the check of language equivalence of FSMs, thus demonstrating decidability of the equality and, consequently, of tactic-invariance. The idea is to construct FSMs $\hat{S}_\Theta$ and $\hat{S}'_\Theta$ that accept $\hat{\Theta}$ and $\hat{\Theta}'$, respectively. To this end, we construct FSMs S_W and S_Θ that accept W and Θ, respectively; here, we exploit the fact that φ can be encoded as FSM S_φ. Then, we 'unfold' the transitions of S_W and S_Θ to obtain $\hat{S}_\Theta$ and $\hat{S}'_\Theta$, respectively. Finally, we decide $\hat{\Theta} = \hat{\Theta}'$ by deciding language equivalence of $\hat{S}_\Theta$ and $\hat{S}'_\Theta$.

Lemma 6.3. *It is decidable whether $\hat{\Theta}$ and $\hat{\Theta}'$ are equal sets.* ⌟

Assuming that $\varphi = \varphi_1$, Appendix B exemplifies the construction of S_φ, S_W, S_Θ, $\hat{S}_\Theta$, and $\hat{S}'_\Theta$.

Deciding tactic-invariance of φ requires at most exponential space w.r.t. to the size of φ and $\bowtie$: The FSM S_φ can be computed in EXPTIME and has exponential size w.r.t. φ and $\bowtie$. The FSMs $\hat{S}_\Theta$ and $\hat{S}'_\Theta$ can be computed from S_φ in polynomial time; the resulting FSMs having exponential size. Finally, the equivalence check requires polynomial space w.r.t. the size of the exponentially-sized FSMs.

7. Deciding Isotactics

In this section, we demonstrate the decidability of interleaving isotactics for languages of FSMs S_1 and S_2 w.r.t. an alignment $\bowtie$. The idea of the proposed approach is shown in Figure 4. It comprises three steps:

1. Compute the '*product*' $\mathcal{W}(S_1, S_2, \bowtie)$ of S_1 and S_2 w.r.t. $\bowtie$, called *witness graph*.
2. For $i \in \{1, 2\}$, compute the '*projection*' $\mathcal{L}_i(\mathcal{W}(S_1, S_2, \bowtie))$ of $\mathcal{W}(S_1, S_2, \bowtie)$, describing the behaviour of S_i that can be mirrored by S_j, $j \in \{1, 2\}, j \neq i$.
3. Reduce deciding interleaving isotactics to a language equivalence check, that is, the equality of the projection $\mathcal{L}_i(\mathcal{W}(S_1, S_2, \bowtie))$ and the trace set $\mathcal{T}_{\overset{i}{\bowtie}}(\mathcal{L}(S_i))$.

Again, the proofs of the main formal statements proposed in this section are in Appendix A.

Determinism. To simplify subsequent discussions, we assume that S_1 and S_2 are *deterministic* w.r.t. $\overset{1}{\bowtie}$ and $\overset{2}{\bowtie}$, respectively. Here, determinism is defined w.r.t. a given grouping, and requires the absence of labels which do not participate in any group.

Definition 7.1 (Deterministic FSM)**.** An FSM $S := (\mathcal{Q}, \Lambda, \rightarrow, q^{ini}, F)$ is *deterministic* w.r.t. a grouping κ of Λ, iff for all states $q \in \mathcal{Q}$ and for all transitions $q \xrightarrow{\lambda_1} q_1$ and $q \xrightarrow{\lambda_2} q_2$ of S: If both $\mathcal{G}_\kappa(\lambda_1) \neq \varnothing$ and $\mathcal{G}_\kappa(\lambda_1) = \mathcal{G}_\kappa(\lambda_2)$, then $q_1 = q_2$. ⌟

The FSMs m_1 and m_2 in Figure 1 are deterministic w.r.t. to $\overset{1}{\bowtie}$ and $\overset{2}{\bowtie}$, respectively. A counter example for determinism would be if in m_1 one adds a fresh state 4 and a fresh transition $1 \xrightarrow{c} 4$. The FSM obtained in this way is nondeterministic w.r.t. $\overset{1}{\bowtie}$.

Given an FSM S and a grouping, one can always construct a deterministic FSM w.r.t. the grouping that describes the same set of induced traces as S. The construction can be accomplished using the powerset construction [21].

Lemma 7.2. *Let $S := (\mathcal{Q}, \Lambda, \rightarrow, q^{ini}, F)$ be an FSM, and let κ be a grouping of Λ. There exists an FSM S' such that: (i) $\mathcal{T}_\kappa(\mathcal{L}(S)) = \mathcal{T}_\kappa(\mathcal{L}(S'))$, and (ii) S' is deterministic w.r.t. κ.* ⌟

For the remainder of this section, we fix an alignment $\bowtie : \Lambda_1 \otimes \Lambda_2$, and two FSMs $S_i := (\mathcal{Q}_i, \Lambda_i, \rightarrow_i, q_i^{ini}, F_i)$, $i \in \{1,2\}$, such that S_i is deterministic w.r.t. $\overset{i}{\bowtie}$.

Matches. The idea behind the *witness graph* $\mathcal{W}(S_1, S_2, \bowtie)$ is to construct a finite representation of all possible ways to align the traces of S_1 with the traces of S_2. To this end, we build a product of S_1 and S_2 where each product state (q_1, q_2) is additionally distinguished by a set M of possible *matches*. Here, a *match* is a pair (K_1, K_2) of two aligned groups K_1 and K_2, indicating that there exist traces w_1 and w_2 yielding q_1 and q_2, respectively, which can be aligned by some tactics θ_1 and θ_2 satisfying $\theta_i(|\theta_i|) = \{K_i\}$, for $i \in \{1,2\}$. The edges of $\mathcal{W}(S_1, S_2, \bowtie)$ are labelled with pairs $(\mathcal{K}_1, \mathcal{K}_2)$, where each $\mathcal{K}_i$, $i \in \{1,2\}$, is a set of groups. A non-empty set of groups $\mathcal{K}_i$ indicates an action of S_i which is abstracted by $\overset{i}{\bowtie}$ to $\mathcal{K}_i$. In contrast to that, $\mathcal{K}_i = \varnothing$ indicates that S_i did not 'move'.

To simplify further discussions, we introduce the notation $M + (\mathcal{K}_1, \mathcal{K}_2)$ for a set M of matches and a pair $(\mathcal{K}_1, \mathcal{K}_2)$ of sets of groups. Intuitively, $M + (\mathcal{K}_1, \mathcal{K}_2)$ describes the set of possible result matches if we start from M, and both FSMs act according to $(\mathcal{K}_1, \mathcal{K}_2)$.

Let $M \subseteq \bowtie$ and let $\mathcal{K}_i \subseteq \overset{i}{\bowtie}$, $i \in \{1,2\}$, such that $\mathcal{K}_1 \cup \mathcal{K}_2 \neq \varnothing$. Then, $M + (\mathcal{K}_1, \mathcal{K}_2)$ is defined as follows:

$$M+(\mathcal{K}_1,\mathcal{K}_2) := \begin{cases} (\mathcal{K}_1 \times \mathcal{K}_2) \cap \bowtie & \text{if } M = \varnothing \\ \{(G_1,G_2) \in ((\mathcal{K}_1 \times \mathcal{K}_2) \cap \bowtie) \setminus M \mid \exists (G_1', G_2') \in M : G_1 \neq G_1' \wedge G_2 \neq G_2'\} & \text{if } \mathcal{K}_1 \neq \varnothing \wedge \mathcal{K}_2 \neq \varnothing, \\ \{(G_1,G_2) \in M \mid \exists\, i \in \{1,2\} : G_i \in \mathcal{K}_i\} & \text{otherwise.} \end{cases}$$

Consider m_1, m_2, and $\bowtie$ as defined in Figure 1, and let $M := \{(\{b,c\},\{t,u\}), (\{d,e\},\{s,w,x\})\}$. Then, it holds that $M + (\{\{a\},\{b,c\}\}, \{\{s,v\},\{s,w,x\}\}) = \{(\{a\},\{s,v\})\}$, $M + (\{\{b,c\}\}, \varnothing) = \{(\{b,c\},\{t,u\})\}$, and $M + (\{\{b,c\}\}, \{\{t,u\}\}) = \varnothing$.

The Witness Graph. Next, we define the *witness graph* $\mathcal{W}(S_1, S_2, \bowtie)$ based on the possible transitions in S_1 and S_2. We start from the initial state, and the empty set of matches. Then, we add nodes and edges according to the respective transition relations and the alignment.

Definition 7.3 (Witness Graph)**.** The *witness graph* $\mathcal{W}(S_1, S_2, \bowtie)$ of S_1 and S_2 w.r.t. an alignment $\bowtie$ is the least edge-labelled graph (V, E), where $V \subseteq \mathcal{Q}_1 \times \mathcal{Q}_2 \times \wp(\bowtie)$ and $E \subseteq V \times \wp(\overset{1}{\bowtie}) \times \wp(\overset{2}{\bowtie}) \times V$, such that:

1. $(q_1^{ini}, q_2^{ini}, \varnothing) \in V$.
2. Let $v = (q_1, q_2, M) \in V$.
   (a) For $i \in \{1,2\}$, let $q_i \xrightarrow{\lambda_i}_i q_i'$ be a transition of S_i, $\mathcal{K}_i = \mathcal{G}_{\overset{i}{\bowtie}}(\lambda_i)$, and $M' = M + (\mathcal{K}_1, \mathcal{K}_2)$, such that $M' \neq \varnothing$. Then, it holds that $v' := (q_1', q_2', M') \in V$ and $(v, \mathcal{K}_1, \mathcal{K}_2, v') \in E$.
   (b) Let $i, j \in \{1,2\}$, $i \neq j$. Let $q_i \xrightarrow{\lambda_i}_i q_i'$ be a transition of S_i, $\mathcal{K}_i = \mathcal{G}_{\overset{i}{\bowtie}}(\lambda_i)$, $\mathcal{K}_j = \varnothing$, $q_j' = q_j$, and $M' = M + (\mathcal{K}_1, \mathcal{K}_2)$, such that $M' \neq \varnothing$. Then, it holds that $v' := (q_1', q_2', M') \in V$ and $(v, \mathcal{K}_1, \mathcal{K}_2, v') \in E$.

Let $e = (v, \mathcal{K}_1, \mathcal{K}_2, v') \in E$. For $i \in \{1,2\}$, $e[i] := \mathcal{K}_i$ and $e[i]_{\neq\varnothing} := \begin{cases} e[i] & e[i] \neq \varnothing, \\ \varepsilon & \text{otherwise.} \end{cases}$

⌟

Let $v_0 \dots v_n \in V^*$ such that $v_0 = (q_1^{ini}, q_2^{ini}, \varnothing)$. Let $\pi = e_1 \dots e_n \in E^*$ such that $\pi(i) = (v_{i-1}, \mathcal{K}_1^i, \mathcal{K}_2^i, v_i)$ for all $i \in \{1, \dots, n\}$. Then, π is a *path* of $\mathcal{W}(S_1, S_2, \bowtie)$ resulting in v_n. We define $\pi[i] := \pi(1)[i] \dots \pi(n)[i]$, and $\pi[i]_{\neq\varnothing} := \pi(1)[i]_{\neq\varnothing} \dots \pi(n)[i]_{\neq\varnothing}$, where $i \in \{1,2\}$.

Figure 5 exemplifies the notion of the witness graph $\mathcal{W}(m_1, m_2, \bowtie)$ for the FSMs m_1 and m_2, and the alignment $\bowtie$ given in Figure 1. In the proposed notation, the outmost round brackets in the node labels are omitted. For example, the topmost node in the figure is $(1, \mathrm{I}, \varnothing)$, but we write $1, \mathrm{I}, \varnothing$. Note that edge labels are written in two lines (the first element is written above the second). We also omit the outmost braces when depicting each of the non-empty elements of the label. For example, the edge from $(1, \mathrm{I}, \varnothing)$ to $(1, \mathrm{II}, \{(\{a\},\{s,v\})\})$ has label $(\{\{a\}\}, \{\{s,v\},\{s,w,x\}\})$. However, in the figure, the corresponding edge

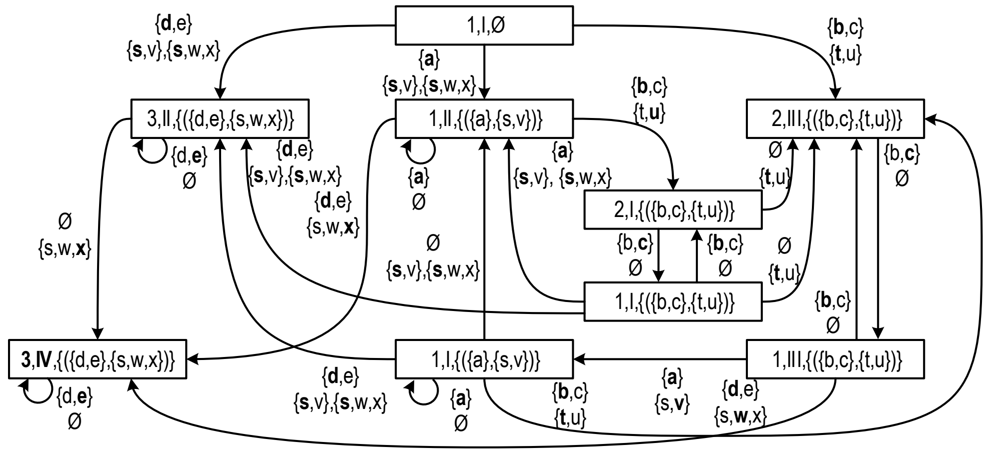

Figure 5. Witness graph for the FSMs and the alignment shown in Figure 1.

is labelled with $\{a\}$ written above $\{s,v\},\{s,w,x\}$. Labels of the respective transitions in m_1 and m_2 are printed in bold in the edge labels of the witness graph. Finally, if both m_1 and m_2 are in a final state, the state names are also put in bold in the graph.

For example, Rule 2(a) in Definition 7.3 produces the edge labelled $(\{\{a\}\},\{\{s,v\},\{s,w,x\}\})$ from node $(1,\mathrm{I},\varnothing)$ to node $(1,\mathrm{II},\{(\{a\},\{s,v\})\}$: $1\xrightarrow{a}_{m_1}1$, $\mathrm{I}\xrightarrow{s}_{m_2}\mathrm{II}$, $\mathcal{G}_{\overset{1}{\bowtie}}(a)=\{\{a\}\}$, $\mathcal{G}_{\overset{2}{\bowtie}}(s)=\{\{s,v\},\{s,w,x\}\}$, and $\varnothing+(\mathcal{G}_{\overset{1}{\bowtie}}(a),\mathcal{G}_{\overset{2}{\bowtie}}(s))=\{(\{a\},\{s,v\})\}$. Rule 2(b) produces the edge labelled $(\{\{b,c\}\},\varnothing)$ from node $(2,\mathrm{III},\{(\{b,c\},\{t,u\})\})$ to $(1,\mathrm{III},\{(\{b,c\},\{t,u\})\})$: $2\xrightarrow{c}_{m_1}1$, $\mathcal{G}_{\overset{1}{\bowtie}}(c)=\{\{b,c\}\}$, and $\{(\{b,c\},\{t,u\})\}+(\mathcal{G}_{\overset{1}{\bowtie}}(c),\varnothing)=\{(\{b,c\},\{t,u\})\}$. The sequence π of edges with respective labels $(\{\{a\}\},\{\{s,v\},\{s,w,x\}\})$, $(\{\{b,c\}\},\{\{t,u\}\})$, $(\{\{b,c\}\},\varnothing)$ and $(\varnothing,\{\{t,u\}\})$ from node $(1,\mathrm{I},\varnothing)$ to node $(2,\mathrm{III},\{(\{b,c\},\{t,u\})\})$ is a path with $\pi[2]=\{\{s,v\},\{s,w,x\}\}\{\{t,u\}\}\varnothing\{\{t,u\}\}$ and $\pi[2]_{\neq\varnothing}=\{\{s,v\},\{s,w,x\}\}\{\{t,u\}\}\{\{t,u\}\}$.

For illustration, we sketch an example producing non-singleton sets of matches: For $i\in\{1,2\}$, let K_i and K_i' be groups with $\lambda_i\in K_i\cap K_i'$. Let $\bowtie'$ be an alignment with $K_1\bowtie' K_2$ and $K_1'\bowtie K_2'$. For $i\in\{1,2\}$, let S_i be an FSM accepting the word λ_i. Then, $\mathcal{W}(S_1,S_2,\bowtie')$ has an edge labelled $(\{K_1,K_1'\},\{K_2,K_2'\})$ from the initial node to a node with matches $\{(K_1,K_2),(K_1',K_2')\}$.

Realisability. A pair $(w_1,w_2)\in\wp(\overset{1}{\bowtie})^*\times\wp(\overset{2}{\bowtie})^*$ is *realisable* iff there exists a path of $\mathcal{W}(S_1,S_2,\bowtie)$ that represents w_1 and w_2, possibly also containing $\varnothing$.

Definition 7.4 (Realisable)**.** For $i\in\{1,2\}$, let $w_i\in\wp(\overset{i}{\bowtie})^*$. Then, (w_1,w_2) is *realisable* in $\mathcal{W}(S_1,S_2,\bowtie)$ resulting in a node v iff there is a path π of $\mathcal{W}(S_1,S_2,\bowtie)$ resulting in v, such that $\pi[i]_{\neq\varnothing}=w_i$, $i\in\{1,2\}$. ⌟

For example, let $w_1:=\mathcal{T}_{\overset{1}{\bowtie}}(bca)=\{\{b,c\}\}\{\{b,c\}\}\{\{a\}\}$ and $w_2:=\mathcal{T}_{\overset{2}{\bowtie}}(tv)=\{\{t,u\}\}\{\{s,v\}\}$. Then, (w_1,w_2) is realisable in the graph in Figure 5, resulting in node $(1,\mathrm{I},\{(\{a\},\{s,v\})\})$. In contrast, the pair of traces $(\mathcal{T}_{\overset{1}{\bowtie}}(bc),w_2)$ is not realisable.

We now show that realisability of (w_1,w_2) implies that w_1 and w_2 can be aligned.

Lemma 7.5. *For $i\in\{1,2\}$, let $w_i\in\wp(\overset{i}{\bowtie})^*$. Let (w_1,w_2) be realisable in $\mathcal{W}(S_1,S_2,\bowtie)$ resulting in a node $v=(q_1,q_2,M)$. For each $(K_1,K_2)\in M$, there exist a tactic θ_1 of w_1 and a tactic θ_2 of w_2, such that $(\theta_1,\theta_2):w_1\bowtie w_2$ and for all $i\in\{1,2\}$: $|w_i|>0\Rightarrow\theta_i(|w_i|)=\{K_i\}$.* ⌟

Because (w_1,w_2) is realizable, where $w_1:=\mathcal{T}_{\overset{1}{\bowtie}}(bca)=\{\{b,c\}\}\{\{b,c\}\}\{\{a\}\}$ and $w_2:=\mathcal{T}_{\overset{2}{\bowtie}}(tv)=\{\{t,u\}\}\{\{s,v\}\}$, w_1 and w_2 can be aligned. To justify this fact, one can use tactics $\theta_1:=w_1$ and $\theta_2:=w_2$, and the bijection $\{\{1,2\}\mapsto\{1\},\{3\}\mapsto\{2\}\}$ from $\{1,\ldots,|\theta_1|\}/_{\theta_1}$ to $\{1,\ldots,|\theta_2|\}/_{\theta_2}$.

We now show the converse, i.e., if w_1 and w_2 can be aligned, then (w_1,w_2) is realisable. The idea of the proof is to construct a path that justifies that (w_1,w_2) is indeed realisable based on alignable tactics θ_1 and

θ_2 in the order of the aligned equivalence classes: Rule 2(a) creates an edge for every fresh pair of aligned equivalence classes, while Rule 2(b) creates the edges for the remaining indices in these equivalence classes.

Lemma 7.6. *For $i \in \{1,2\}$, let σ_i be a prefix of some word in $\mathcal{L}(S_i)$, and $w_i = \mathcal{T}_{\bowtie}^{i}(\sigma_i)$. Let for $i \in \{1,2\}$, θ_i be a tactic of w_i, such that $(\theta_1, \theta_2) : w_1 \bowtie w_2$. Then, (w_1, w_2) is realisable resulting in some node $v = (q_1, q_2, M)$ with $|w_1| > 0 \Rightarrow |w_2| = 0 \wedge (\theta_1(|w_1|), \theta_2(|w_2|)) \in M$.* ⌟

For illustration, we again consider $w_1 := \mathcal{T}_{\bowtie}^{1}(bca) = \{\{b,c\}\}\{\{b,c\}\}\{\{a\}\}$ and $w_2 := \mathcal{T}_{\bowtie}^{2}(tv) = \{\{t,u\}\}\{\{s,v\}\}$, which can be aligned by tactics $\theta_1 := w_1$ and $\theta_2 := w_2$, and bijection $\{\{1,2\} \mapsto \{1\}, \{3\} \mapsto \{2\}\}$. One can construct a path that justifies that (w_1, w_2) is realisable by using the bijection: $\{1,2\} \mapsto \{1\}$ yields that one must start by first taking edge $(\theta_1(1), \theta_2(1))$ followed by edge $(\theta_1(2), \varnothing)$. Then, $\{3\} \mapsto \{2\}$ yields edge $(\theta_1(3), \theta_2(2))$.

Projecting the Witness Graph. Intuitively, for $i, j \in \{1,2\}$, $i \neq j$, we can conceive $\mathcal{W}(S_1, S_2, \bowtie)$ as an FSM, where for each edge labelled $(\mathcal{K}_1, \mathcal{K}_2)$, we only consider the action $\mathcal{K}_i$ of S_i, i.e., we omit the actions of S_j:

Definition 7.7 (Witness Graph Projection)**.** Let Π be the set of all paths of $\mathcal{W}(S_1, S_2, \bowtie)$ resulting in a node (q_1, q_2, M), where $q_1 \in F_1$ and $q_2 \in F_2$. For $i \in \{1,2\}$, we define $\mathcal{L}_i(\mathcal{W}(S_1, S_2, \bowtie)) := \{\pi[i]_{\neq\varnothing} \mid \pi \in \Pi\}$. ⌟

From Lemma 7.6 and Lemma 7.5, we get that $\mathcal{L}_i(\mathcal{W}(S_1, S_2, \bowtie))$ describes the behavior of S_i which can be aligned to behavior of S_j. This enables us to reduce deciding isotactics to comparing S_i to $\mathcal{L}_i(\mathcal{W}(S_1, S_2, \bowtie))$.

Reducing Decidability of Isotactics to Language Equivalence. We reduce the problem of deciding isotactics of $\mathcal{L}(S_1)$ and $\mathcal{L}(S_2)$ to two language equivalence checks, between the FSMs and the respective projections of the witness graph.

Lemma 7.8. *The following statements are equivalent:*

1. $\mathcal{L}_1(\mathcal{W}(S_1, S_2, \bowtie)) = \mathcal{T}_{\bowtie}^{1}(\mathcal{L}(S_1))$ *and* $\mathcal{L}_2(\mathcal{W}(S_1, S_2, \bowtie)) = \mathcal{T}_{\bowtie}^{2}(\mathcal{L}(S_2))$.
2. $\mathcal{L}(S_1) \doteqdot_{\bowtie} \mathcal{L}(S_2)$. ⌟

Because language equivalence is decidable for FSMs [21], we can also decide the first proposition of Lemma 7.8 by transforming $\mathcal{W}(S_1, S_2, \bowtie)$ into two FSMs: One with language $\mathcal{L}_1(\mathcal{W}(S_1, S_2, \bowtie))$ and the other with language $\mathcal{L}_2(\mathcal{W}(S_1, S_2, \bowtie))$. The transformation basically comprises the projection of the edge labels to the i-th component, $i \in \{1,2\}$, and the subsequent "removal" of $\varnothing$-transitions.

Lemma 7.9. *For $i \in \{1,2\}$, the following problem is decidable:*
To decide whether it holds that $\mathcal{L}_i(\mathcal{W}(S_1, S_2, \bowtie)) = \mathcal{T}_{\bowtie}^{i}(\mathcal{L}(S_i))$. ⌟

We conclude that deciding isotactics is in EXSPACE: If the FSMs are deterministic, $\mathcal{W}(S_1, S_2, \bowtie)$ has at most $|\mathcal{Q}_1| \cdot |\mathcal{Q}_2| \cdot 2^{|\bowtie|}$ nodes; otherwise, determinisation of the FSMs yields a witness graph with at most $2^{|\mathcal{Q}_1||\mathcal{Q}_2||\bowtie|}$ nodes. $\mathcal{W}(S_1, S_2, \bowtie)$ can be computed in EXPTIME. Deciding language equivalence requires polynomial space in the size of the FSMs.

8. Concluding Remarks

This paper proposed interleaving isotactics to assess behavioural equivalence of aligned models, presented results on the properties it preserves, and proved decidability of the respective verification problems. The constructions introduced in Section 7 to show decidability of isotactics, i.e., the witness graph and its projections, give rise to a first decision procedure for isotactics. This procedure has been implemented and is available in an open-source tool.[4] It takes as input two FSMs and the definition of an alignment and returns a Boolean result. It also enables the creation of visualisations of the constructed witness graph and its projections and comes with exemplary input files that encode the running example of this paper.

[4] `https://github.com/Isotactics/deciding-isotactics`

Having introduced a novel equivalence notion, we also reflect on potential causes for non-equivalence. First and foremost, an action that is part solely of one model may lead to non-equivalence. In our model, this situation may manifest when the label of such an action is related to itself in the alignment. As occurrences of this label are limited to runs of one of the system models, they cannot be mirrored by any run of the other model. Assessment of behavioural equivalence of system models often starts by hiding non-matching actions, i.e., by making non-matching actions silent, or invisible. Then, models are compared based on the remaining visible actions, see for example weak trace equivalence [11]. Isotactics is a generalization of (notions like) weak trace equivalence. One can implement action hiding using the isotactics' alignment relation, by excluding the labels of non-matching actions from the alignment. Note that weak trace equivalence cannot capture behavioural correspondence of an action in one model with two distinct mutually exclusive actions in the other model. This correspondence, however, can be captured in an alignment relation and verified using isotactics. Finally, the notion of behaviour inheritance suggests that when verifying an equivalence, in addition to being hidden, an action can be blocked [2]. That is, the blocked action, and all the subsequent actions, are considered to be not reachable. This idea, originally proposed in the context of branching bisimulation, can be lifted to isotactics in a straightforward manner.

For system models that are not behaviourally equivalent, it is often relevant to quantify the discrepancies in their behaviour. To this end, measures for behavioural similarity have been proposed for the different semantics of system models, see [7, 8]. We foresee that the proposed notion of isotactics can be exploited for the definition of similar measures. For instance, such new measures can aim to quantify the ratio of the aligned groups of labels for which the behavioural projections are equivalent.

Acknowledgements

We are grateful for the support by the Universities Australia (UA) and the German Academic Exchange Service (DAAD) as part of the Joint Research Co-operation Scheme. Artem Polyvyanyy was partly supported by the Australian Research Council Discovery Project DP180102839. We further would like to thank Frank Lange and Sebastian Misch for their work on the implementation of the isotactics decision procedure.

References

[1] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking.* MIT Press, 2008.

[2] Twan Basten and Wil M. P. van der Aalst. Inheritance of behavior. *J. Logic. Algebr. Program.*, 47(2):47–145, 2001. doi:10.1016/S1567-8326(00)00004-7.

[3] Eike Best, Raymond R. Devillers, Astrid Kiehn, and Lucia Pomello. Concurrent bisimulations in Petri nets. *Acta Inform.*, 28(3):231–264, 1991. doi:10.1007/BF01178506.

[4] Moisés Castelo Branco, Javier Troya, Krzysztof Czarnecki, Jochen Malte Küster, and Hagen Völzer. Matching business process workflows across abstraction levels. In *MoDELS*, pages 626–641, 2012. doi:10.1007/978-3-642-33666-9_40.

[5] Wilfried Brauer, Robert Gold, and Walter Vogler. A survey of behaviour and equivalence preserving refinements of Petri nets. In *ATPN*, pages 1–46, 1989. doi:10.1007/3-540-53863-1_19.

[6] Michael C. Browne, Edmund M. Clarke, and Orna Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *Theor. Comput. Sci.*, 59:115–131, 1988. doi:10.1016/0304-3975(88)90098-9.

[7] Luca de Alfaro, Marco Faella, and Mariëlle Stoelinga. Linear and branching system metrics. *IEEE Trans. Softw. Eng.*, 35(2):258–273, 2009. doi:10.1109/TSE.2008.106.

[8] Josee Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled Markov processes. *Theor. Comput. Sci.*, 318(3):323–354, 2004. doi:10.1016/j.tcs.2003.09.013.

[9] V. M. Glushkov. The abstract theory of automata. *Russian Mathematical Surveys*, 16(5):1–53, 1961. doi:10.1070/RM1961v016n05ABEH004112.

[10] Volker Guth and Andreas Oberweis. Delta-analysis of Petri net based models for business processes. In *International Conference on Applied Informatics*, pages 23–32, 1997.

[11] Petr Jancar, Javier Esparza, and Faron Moller. Petri nets and regular processes. *J. Comput. Syst. Sci.*, 59(3):476–503, 1999. doi:10.1006/jcss.1999.1643.

[12] Michael Kaminski. Invariance under stuttering in a temporal logic of actions. *Theor. Comput. Sci.*, 368(1–2):50–63, 2006. doi:10.1016/j.tcs.2006.06.020.

[13] Donald E. Knuth. *The Art of Computer Programming, Volume I: Fundamental Algorithms.* Addison-Wesley, 1968.

[14] Martin Lange, Étienne Lozes, and Manuel Vargas Guzmán. Model-checking process equivalences. *Theor. Comput. Sci.*, 560:326–347, 2014. doi:10.1016/j.tcs.2014.08.020.

[15] Doron A. Peled and Thomas Wilke. Stutter-invariant temporal properties are expressible without the next-time operator. *Inf. Process. Lett.*, 63(5):243–246, 1997. doi:10.1016/S0020-0190(97)00133-6.
[16] Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57, 1977. doi:10.1109/SFCS.1977.32.
[17] Artem Polyvyanyy, Chun Ouyang, Alistair Barros, and Wil M. P. van der Aalst. Process querying: Enabling business intelligence through query-based process analytics. *Decis. Support Syst.*, 100:41–56, 2017. doi:10.1016/j.dss.2017.04.011.
[18] Artem Polyvyanyy, Matthias Weidlich, and Mathias Weske. Isotactics as a foundation for alignment and abstraction of behavioral models. In *BPM*, pages 335–351, 2012. doi:10.1007/978-3-642-32885-5_26.
[19] Michael O. Rabin and Dana Scott. Finite automata and their decision problems. *IBM J. Res. Dev.*, 3(2):114–125, 1959. doi:10.1147/rd.32.0114.
[20] Arend Rensink and Roberto Gorrieri. Vertical implementation. *Inf. Comput.*, 170(1):95–133, 2001. doi:10.1006/inco.2001.2967.
[21] Michael Sipser. *Introduction to the Theory of Computation.* Thomson Course Technology, Boston, 2006.
[22] Arthur H. M. ter Hofstede, Wil M. P. van der Aalst, Michael Adams, and Nick Russell. *Modern Business Process Automation: YAWL and Its Support Environment.* Springer Publishing Company, Incorporated, 1st edition, 2009.
[23] Wil M. P. van der Aalst. Inheritance of business processes: A journey visiting four notorious problems. In *Petri Net Technology for Communication-Based Systems*, pages 383–408, 2003. doi:10.1007/978-3-540-40022-6_19.
[24] Wil M. P. van der Aalst. Business alignment: using process mining as a tool for Delta analysis and conformance testing. *Requir. Eng.*, 10(3):198–211, 2005. doi:10.1007/s00766-005-0001-x.
[25] Wil M. P. van der Aalst, Arya Adriansyah, and Boudewijn F. van Dongen. Replaying history on process models for conformance checking and performance analysis. *Wiley Interdiscip. Rev.-Data Mining Knowl. Discov.*, 2(2):182–192, 2012. doi:10.1002/widm.1045.
[26] Rob J. van Glabbeek. The linear time-branching time spectrum (extended abstract). In *CONCUR*, volume 458 of *Lecture Notes in Computer Science*, pages 278–297. Springer, 1990. doi:10.1007/BFb0039066.
[27] Rob J. van Glabbeek and Ursula Goltz. Well-behaved flow event structures for parallel composition and action refinement. *Theor. Comput. Sci.*, 311(1–3):463–478, 2004. doi:10.1016/j.tcs.2003.10.031.
[28] Barbara Weber, Manfred Reichert, and Stefanie Rinderle-Ma. Change patterns and change support features – enhancing flexibility in process-aware information systems. *Data Knowl. Eng.*, 66(3):438–466, 2008. doi:10.1016/j.datak.2008.05.001.
[29] Matthias Weidlich, Remco M. Dijkman, and Jan Mendling. The ICoP framework: Identification of correspondences between process models. In *CAiSE*, pages 483–498, 2010. doi:10.1007/978-3-642-13094-6_37.
[30] Matthias Weidlich, Remco M. Dijkman, and Mathias Weske. Deciding behaviour compatibility of complex correspondences between process models. In *BPM*, pages 78–94, 2010. doi:10.1007/978-3-642-15618-2_8.
[31] Matthias Weidlich, Remco M. Dijkman, and Mathias Weske. Behaviour equivalence and compatibility of business process models with complex correspondences. *Comput. J.*, 55(11):1398–1418, 2012. doi:10.1093/comjnl/bxs014.
[32] F. Wijnhoven, T. Spil, R. Stegwee, and R.T.A. Fa. Post-merger IT integration strategies: An IT alignment perspective. *J. Strateg. Inf. Syst.*, 15(1):5–28, 2006. doi:10.1016/j.jsis.2005.07.002.

Appendix A. Proofs

Lemma 5.1. *For $i \in \{1,2\}$, let Λ_i be a set of labels and let $\Sigma_i \subseteq \Lambda_i^*$. Let b be a bijection from Λ_1 to Λ_2, and let $\bowtie := \{(\{\lambda\},\{b(\lambda)\}) \mid \lambda \in \Lambda_1\}$. If $\mathcal{T}^1_{\bowtie}(\Sigma_1)$ and $\mathcal{T}^2_{\bowtie}(\Sigma_2)$ are trace equivalent up to b, then $\Sigma_1 \doteq_{\bowtie} \Sigma_2$.* ⌟

Proof. According to Definition 2.1, there exists a bijection R from $\mathcal{T}^1_{\bowtie}(\Sigma_1)$ to $\mathcal{T}^2_{\bowtie}(\Sigma_2)$ such that for all $(w_1,w_2) \in R$ it holds that (i) $|w_1| = |w_2|$, and (ii) for all $i \in \{1,\ldots,|w_1|\}$ it holds that $b(w_1(i)) = w_2(i)$. Let $(w_1,w_2) \in R$. Because $\bowtie \subseteq \wp_{=1}(\Lambda_1) \times \wp_{=1}(\Lambda_2)$, for all $i \in \{1,\ldots,|w_1|\}$ it holds that $|w_1(i)| = 1 = |w_2(i)|$. Therefore, for $i \in \{1,2\}$, w_i is the only tactic of w_i. Note that for all $i \in \{1,\ldots,|w_1|\}$, it also holds that $w_1(i) \bowtie w_2(i)$ and, thus, $\{1,\ldots,|w_1|\}/w_1 = \{1,\ldots,|w_2|\}/w_2$. Then, it trivially holds that $(w_1,w_2) : w_1 \bowtie w_2$, cf. Definition 3.6; one can use the identity relation on $\{1,\ldots,|w_1|\}/w_1$ as a bijection from $\{1,\ldots,|w_1|\}/w_1$ to $\{1,\ldots,|w_2|\}/w_2$ to justify this fact. Because R is a bijection, we get $\Sigma_1 \leqslant_{\bowtie} \Sigma_2$. Because R and $\bowtie$ are bijections, we get $\Sigma_2 \leqslant_{\bowtie^{-1}} \Sigma_1$; note that $\bowtie$ is a bijection because b is a bijection. Thus, it holds that $\Sigma_1 \doteq_{\bowtie} \Sigma_2$. ■

Lemma 5.2. *For $i \in \{1,2\}$, let Λ_i be a set of labels and let $\Sigma_i \subseteq \Lambda_i^*$ be repetition-free. Let b be a bijection from Λ_1 to Λ_2, and let $\bowtie := \{(\{\lambda\},\{b(\lambda)\}) \mid \lambda \in \Lambda_1\}$. If $\Sigma_1 \doteq_{\bowtie} \Sigma_2$, then $\mathcal{T}^1_{\bowtie}(\Sigma_1)$ and $\mathcal{T}^2_{\bowtie}(\Sigma_2)$ are trace equivalent up to b.* ⌟

Proof. Because $\bowtie$ is a relation on singletons, for $i \in \{1,2\}$, $w_i \in \mathcal{T}^i_{\bowtie}(\Sigma_i)$ is the only tactic of w_i. Let $R := \{(x,y) \in \mathcal{T}^1_{\bowtie}(\Sigma_1) \times \mathcal{T}^2_{\bowtie}(\Sigma_2) \mid (x,y) : x \bowtie y\}$. Let $(w_1,w_2) \in R$. Because Σ_1 and Σ_2 are repetition-free and because $\bowtie$ is a bijection, it holds that w_1 and w_2 are repetition-free; note that $\bowtie$ is a bijection because b is a bijection. Therefore, for $i \in \{1,2\}$, it holds that $\{1,\ldots,|w_i|\}/w_i = \wp_{=1}(\{1,\ldots,|w_i|\})$. Hence, $|w_1| = |w_2|$ and for all $k \in onetoww_1$ it holds that $w_1(k) \bowtie w_2(k)$ and, thus, $b(w_1(k)) = w_2(k)$. Next, we show that

R is a bijection from $\mathcal{T}_{\bowtie}^{1}(\Sigma_1)$ to $\mathcal{T}_{\bowtie}^{2}(\Sigma_2)$. Let $(w_1, w_2), (w_1', w_2') \in R$ and $i, j \in \{1,2\}$, $i \neq j$. Let us assume that $w_i = w_i'$. As shown above, it holds that $|w_i| = |w_j|$ and $|w_i'| = |w_j'|$. Then, $w_i = w_i'$ yields $|w_j| = |w_j'|$. Additionally, for every $k \in \{1, \ldots, |w_i|\}$ it holds that $w_1(k) \bowtie w_2(k)$ and $w_1'(k) \bowtie w_2'(k)$. Because $w_i = w_i'$ and $\bowtie$ is a bijection, $w_j(k) = w_j'(k)$, $k \in \{1, \ldots, |w_j|\}$. Hence, it holds that $w_j = w_j'$. Finally, because $\Sigma_1 \leqslant_{\bowtie} \Sigma_2$ and $\Sigma_2 \leqslant_{\bowtie^{-1}} \Sigma_1$, for $i \in \{1,2\}$, it holds that for all $\sigma \in \Sigma_i$ there exists $(w_1, w_2) \in R$ such that $\mathcal{T}_{\bowtie}^{i}(\sigma) = w_i$. Thus, R is a bijection from $\mathcal{T}_{\bowtie}^{1}(\Sigma_1)$ to $\mathcal{T}_{\bowtie}^{2}(\Sigma_2)$ that justifies the fact that $\mathcal{T}_{\bowtie}^{1}(\Sigma_1)$ and $\mathcal{T}_{\bowtie}^{2}(\Sigma_2)$ are trace equivalent up to b. ∎

Lemma 6.1. *Let $\bowtie : \Lambda_1 \otimes \Lambda_2$ be an alignment. For $i \in \{1,2\}$, let $\sigma_i \in \Lambda_i^*$, $w_i := \mathcal{T}_{\bowtie}^{i}(\sigma_i)$, θ_i be a tactic of w_i, and let $\varphi_i \in \mathsf{LTL}[\overset{i}{\bowtie}]$ be tactic-invariant w.r.t. Λ_i and $\overset{i}{\bowtie}$. Let φ_1 and φ_2 be aligned by $\bowtie$, i.e., $\varphi_1 \bowtie \varphi_2$, and let $(\theta_1, \theta_2) : w_1 \bowtie w_2$. Then, it holds that $w_1 \vDash \varphi_1$ iff $w_2 \vDash \varphi_2$.* ⌟

Proof. For $i \in \{1,2\}$, it holds that $\theta_i \vDash \varphi_i$ iff $w_i \vDash \varphi_i$, because φ_i is tactic-invariant. We show that $\theta_1 \vDash \varphi_1$ iff $\theta_2 \vDash \varphi_2$ by the structural induction on φ_1. Let $\varphi_1 := \bigvee_{K_1 \in \mathcal{K}_1} K_1$ and $\varphi_2 := \bigvee_{K_2 \in \mathcal{K}_2} K_2$, where $\mathcal{K}_1 \subseteq \overset{1}{\bowtie}$ and $\mathcal{K}_2 \subseteq \overset{2}{\bowtie}$ such that $(\mathcal{K}_1 \times \overset{2}{\bowtie}) \cap \bowtie = (\overset{1}{\bowtie} \times \mathcal{K}_2) \cap \bowtie$. Then, for $j \in \{1,2\}$, $\theta_j \vDash \varphi_j$ iff $\theta_j(1) \cap \mathcal{K}_j \neq \varnothing$. Because $\theta_1(1) \bowtie \theta_2(1)$, $\theta_1(1) \cap \mathcal{K}_1 \neq \varnothing$ iff $\theta_2(1) \cap \mathcal{K}_2 \neq \varnothing$ and, thus, it holds that $\theta_1 \vDash \varphi_1$ iff $\theta_2 \vDash \varphi_2$. Let $\varphi_1 := \neg\psi_1$ and $\varphi_2 := \neg\psi_2$ such that $\psi_1 \bowtie \psi_2$. Then, for $k \in \{1,2\}$, it holds that $\theta_k \vDash \psi_k$ iff $\theta_k \nvDash \varphi_k$. Using the inductive assumption, we conclude that $\theta_1 \vDash \varphi_1$ iff $\theta_2 \vDash \varphi_2$. Let $\varphi_1 := \psi_1 \vee \psi_1'$ and $\varphi_2 := \psi_2 \vee \psi_2'$ such that $\psi_1 \bowtie \psi_2$ and $\psi_1' \bowtie \psi_2'$. Then, for $k \in \{1,2\}$, it holds that $\theta_k \vDash \varphi_k$ iff $\theta_k \vDash \psi_k$ or $\theta_k \vDash \psi_k'$. Again, using the inductive assumption, we conclude that $\theta_1 \vDash \varphi_1$ iff $\theta_2 \vDash \varphi_2$. Let $\varphi_1 := \psi_1 \,\mathsf{U}\, \psi_1'$ and $\varphi_2 := \psi_2 \,\mathsf{U}\, \psi_2'$ such that $\psi_1 \bowtie \psi_2$ and $\psi_1' \bowtie \psi_2'$. Then, for $k \in \{1,2\}$, it holds that $\theta_k \vDash \varphi_k$ iff there exists $\iota_k \in \{1, \ldots, |\theta_k|\}$ such that $\theta_k[\iota_k\rangle \vDash \psi_k'$ and $\theta_k[\eta\rangle \vDash \psi_k$ for all $\eta \in \{1, \ldots, \iota_k - 1\}$. By definition, for $k \in \{1,2\}$ and $\iota \in \{1, \ldots, |\theta_k| - 1\}$ it holds that $\iota =_{\theta_k} \iota + 1$ implies $\theta_k(\iota) = \theta_k(\iota + 1)$. Because of the stutter equivalence of LTL-formulae without the next operator [15], $\iota =_{\theta_k} \iota + 1$ also implies that $\theta_k[\iota\rangle$ and $\theta_k[\iota + 1\rangle$ satisfy exactly the same LTL-formulae. Applying the inductive assumption then yields $\theta_1 \vDash \varphi_1$ iff $\theta_2 \vDash \varphi_2$. Finally, because ψ_1 and ψ_2 are tactic-invariant, it holds that $w_1 \vDash \varphi_1$ iff $w_2 \vDash \varphi_2$. ∎

Lemma 6.2. $\hat{\Theta} = \hat{\Theta}'$ *iff φ is tactic-invariant w.r.t. Λ and κ.* ⌟

Proof. We show both directions separately.

1. "1.⇒2.": Let $w \in \mathcal{T}_\kappa(\Lambda^*)$ and $\theta \in \text{Tactics}(w)$. Assume first $w \vDash \varphi$. Then, $w \in W$, and thus $\theta_w \in \hat{\Theta}$. Then, by assumption, $\theta \in \hat{\Theta}'$. Hence, $\theta \in \Theta$. Therefore, $\theta \vDash \varphi$. Symmetrically, if $\theta \vDash \varphi$, then $\theta \in \Theta$, and thus also $\theta_w \in \hat{\Theta}'$. Then, by assumption, $\theta_w \in \hat{\Theta}$. Then, $w \in W$, and therefore $w \vDash \varphi$.
2. "2.⇒1.": Assume first $\theta_w \in \hat{\Theta}$. Then, $w \in \mathcal{T}_\kappa(\Lambda^*)$ with $w \vDash \varphi$, and $\theta \in \text{Tactics}(w)$ of w. Because φ is tactic-invariant, $\theta \vDash \varphi$. Thus, $\theta \in \Theta$, and $\theta_w \in \hat{\Theta}'$. Now, assume $\theta_w \in \hat{\Theta}'$. Then, there exists a tactic $\theta \in \Theta$ of some word w, such that $\theta \vDash \varphi$. Because φ is tactic-invariant, $w \vDash \varphi$. Therefore, $\theta_w \in \hat{\Theta}'$.

∎

Lemma 6.3. *It is decidable whether $\hat{\Theta}$ and $\hat{\Theta}'$ are equal sets.* ⌟

Proof. We construct an FSM S_φ from φ, such that $\mathcal{L}(S_\varphi)$ is the set of all traces satisfying φ. Then, we construct FSMs $\underline{S}$ and $\overline{S}$ accepting all $w \in \mathcal{T}_\kappa(\Lambda^*)$ with $w(i) \neq \varnothing$ for all $i \in \{1, \ldots, |w|\}$, and $\theta \in \text{Tactics}(w)$ and $\overline{S}$ accepts w, respectively. Let S_W be the intersection of S_φ and $\underline{S}$, and let S_Θ be the intersection of S_φ and $\overline{S}$. We construct the FSM $\hat{S}_\Theta$ from S_W by replacing each transition $q \xrightarrow{\mathcal{K}}_{S_W} q'$ by a transition $q \xrightarrow{(K,\mathcal{K})}_{\hat{S}_\Theta} q'$ for each $K \in \mathcal{K}$. Then, we construct the FSM $\hat{S}_\Theta'$ from S_Θ by replacing each transition $q \xrightarrow{\{K\}}_{S_\Theta} q'$ by a transition $q \xrightarrow{(K,\mathcal{K})}_{\hat{S}_\Theta'} q'$ for each $\mathcal{K} \in \mathcal{G}_\kappa(\Lambda)$ with $K \in \mathcal{K}$. Obviously, the languages of $\hat{S}_\Theta$ and $\hat{S}_\Theta'$ are $\hat{\Theta}$ and $\hat{\Theta}'$, respectively. Thus, we decide $\hat{\Theta} = \hat{\Theta}'$ by deciding equivalence of $\hat{S}_\Theta$ and $\hat{S}_\Theta'$. ∎

Lemma 7.5. *For $i \in \{1,2\}$, let $w_i \in \wp(\overset{i}{\bowtie})^*$. Let (w_1, w_2) be realisable in $\mathcal{W}(S_1, S_2, \bowtie)$ resulting in a node $v = (q_1, q_2, M)$. For each $(K_1, K_2) \in M$, there exist a tactic θ_1 of w_1 and a tactic θ_2 of w_2, such that $(\theta_1, \theta_2) : w_1 \bowtie w_2$ and for all $i \in \{1,2\}$: $|w_i| > 0 \Rightarrow \theta_i(|w_i|) = \{K_i\}$.* ⌟

Proof by induction over the length of the realising path. Because (w_1, w_2) is realisable, there exists a path π of $\mathcal{W}(S_1, S_2, \bowtie)$ resulting in v, such that for $i \in \{1,2\}$, the restriction of $\pi[i]$ to non-empty sets yields w_i. Let $|\pi| = 0$. Then, $w_1 = w_2 = \varepsilon$, trivially satisfying the requirements. Let $|\pi| = 1$. Inspecting the rules 2(a) and 2(b) of Definition 7.3, we find that only a) can produce an edge from the node $(q_1^{ini}, q_2^{ini}, \varnothing)$. Hence, for $i \in \{1,2\}$, $\pi[i] = \mathcal{K}_i \neq \varnothing$, and thus $w_i = \mathcal{K}_i$. Additionally, $M = (\mathcal{K}_1 \times \mathcal{K}_2) \cap \bowtie$. Hence, for each $(G_1, G_2) \in M$, there also exists the tactics $\theta_i = \{G_i\}$ and b proving $(\theta_1, \theta_2) : w_1 \bowtie w_2$. Now, let $|\pi| = n \geq 2$, and the proposition hold for all realising paths π' with $|\pi'| < n$. Let the penultimate node of π be $\hat{v} = (\hat{q}_1, \hat{q}_2, \hat{M})$.

1. Assume that the last edge of π is produced by rule 2(a) and labelled $\mathcal{K}_1, \mathcal{K}_2$. For $i \in \{1,2\}$, $w_i(|w_i|) = \mathcal{K}_i$, and, because $\hat{M} \neq \varnothing$, $M = \{(K_1, K_2) \in ((\mathcal{K}_1 \times \mathcal{K}_2) \cap \bowtie) \setminus \hat{M} \mid \exists (\hat{K}_1, \hat{K}_2) \in \hat{M} : K_1 \neq \hat{K}_1 \wedge K_2 \neq \hat{K}_2\} \neq \varnothing$. Let $\hat{w}_i = w_i(1) \dots w_i(|w_i| - 1)$. Let $(K_1, K_2) \in M$. Then, $(\hat{w}_1, \hat{w}_2)$ is realisable resulting in $\hat{v}$ by the prefix of π with length $n - 1 > 0$. Therefore, for each $(\hat{K}_1, \hat{K}_2) \in \hat{M}$, and $i \in \{1,2\}$, there exist a tactic $\hat{\theta}_i$ of $\hat{w}_i$ with $\hat{\theta}_i(|\hat{w}_i|) = \hat{K}_i$, and $\hat{b}$ proving $(\hat{\theta}_1, \hat{\theta}_2) : \hat{w}_1 \bowtie \hat{w}_2$. Then, for $i \in \{1,2\}$, $\hat{\theta}_i(|\hat{w}_i|) \neq \{K_i\}$. Now, set $\theta_i = \hat{\theta}_i\{K_i\}$. Then, $\{1, \dots, |w_i|\}/\theta_i = \{1, \dots, \hat{w}_i\}/\hat{\theta}_i \cup \{\{|w_i|\}\}$, that is, θ_i adds another equivalence class "to the end" of $\hat{\theta}_i$. Let $b = \hat{b} \cup \{(|w_1|, |w_2|)\}$. Because b proves $(\hat{\theta}_1, \hat{\theta}_2) : \hat{w}_1 \bowtie \hat{w}_2$, and $K_1 \bowtie K_2$, $\hat{b}$ proves $(\theta_1, \theta_2) : w_1 \bowtie w_2$.
2. Assume that the last edge of π is produced by rule 2(b) and labelled $(\mathcal{K}_1, \mathcal{K}_2)$. Then, there exist $i, j \in \{1,2\}$, $i \neq j$, $\lambda_i \in \Lambda_i$, $\mathcal{K}_i = \mathcal{G}_{\bowtie}^{i}(\lambda_i)$, such that $\hat{q}_i \xrightarrow{\lambda_i}_i q_i$, and $M = \{(K_1, K_2) \in \hat{M} \mid \exists\, i \in \{1,2\} : K_i \in \mathcal{K}_i\}$. Let $\hat{w}_i = w_i(1) \dots w_i(|w_i| - 1)$. Let $\hat{w}_j = w_j$. Then, $(\hat{w}_1, \hat{w}_2)$ is realisable resulting in $\hat{v}$ by the prefix of π with length $n - 1 > 0$. Therefore, for $k \in \{1,2\}$, and each $(K_1, K_2) \in \hat{M}$ (thus also $(K_1, K_2) \in M$), there exist a tactic $\hat{\theta}_k$ of $\hat{w}_k$ with $\hat{\theta}_k(|\hat{w}_k|) = K_i$, $\hat{\theta}_j(|\hat{w}_j|) = \hat{G}_j$, and $\hat{b}$ proving $(\hat{\theta}_1, \hat{\theta}_2) : \hat{w}_1 \bowtie \hat{w}_2$. Let $\theta_i = \hat{\theta}_i\{K_i\}$, $\hat{X} = \langle|\hat{w}_i|\rangle_{\hat{\theta}_i}$ be the "last" equivalence class in $\hat{\theta}_i$, and $X = \hat{X} \cup \{|w_i|\}$. Then, $\{1, \dots, |w_i|\}/\theta_i = \{1, \dots, \hat{w}_i\}/\hat{\theta}_i \setminus \{\hat{X}\} \cup \{X\}$, that is, the "last" equivalence class of θ_i is the union of the last equivalence class of $\hat{\theta}_i$ and $\{|w_i|\}$. Let $\theta_j = \hat{\theta}_j$, $\hat{Y} = \langle|\hat{w}_j|\rangle_{\hat{\theta}_j}$ and $b = \hat{b} \setminus \{(\hat{X}, \hat{Y})\} \cup \{(X, \hat{Y})\}$. Then, b proves $(\theta_1, \theta_2) : w_1 \bowtie w_2$.

■

Lemma 7.6. *For $i \in \{1,2\}$, let σ_i be a prefix of some word in $\mathcal{L}(S_i)$, and $w_i = \mathcal{T}_{\bowtie}^{i}(\sigma_i)$. Let for $i \in \{1,2\}$, θ_i be a tactic of w_i, such that $(\theta_1, \theta_2) : w_1 \bowtie w_2$. Then, (w_1, w_2) is realisable resulting in some node $v = (q_1, q_2, M)$ with $|w_1| > 0 \Rightarrow |w_2| = 0 \wedge (\theta_1(|w_1|), \theta_2(|w_2|)) \in M$.* ⌟

Proof by induction over the number of equivalence classes in the tactics. Let w_1, w_2 be traces with respective tactics θ_1, θ_2, such that $(\theta_1, \theta_2) : w_1 \bowtie w_2$. Let $n = |\{1, \dots, |w|\}_1/\theta_1|$. We observe $n = |\{1, \dots, |w_2|\}/\theta_2|$ because there exists a bijection b aligning θ_1 and θ_2.

1. Let $n = 0$. Then, $w_1 = w_2 = \varepsilon$. Then, the empty path proves realisability of (w_1, w_2).

2. Let $n = 1$. Then, for $i \in \{1,2\}$, there exists a label λ_i and a state q_i with $q_i^{ini} \xrightarrow{\lambda_i}_{S_i} q_i$, $w_i = \mathcal{T}_{\bowtie}^{i}(\lambda_i) = \mathcal{G}_{\bowtie}^{i}(\lambda_i)$, and $K_i \in \mathcal{G}_{\bowtie}^{i}(\lambda_i)$. Let $M = \varnothing + (\mathcal{G}_{\bowtie}^{1}(\lambda_1), \mathcal{G}_{\bowtie}^{2}(\lambda_2))$. From $(\theta_1, \theta_2) : w_1 \bowtie w_2$, we get $K_1 \bowtie K_2$ and $(K_1, K_2) \in M$. Hence, $M \neq \varnothing$, and rule 2(a) produces an edge labelled $(\mathcal{G}_{\bowtie}^{1}(\lambda_1), \mathcal{G}_{\bowtie}^{2}(\lambda_2))$ from $(q_1^{ini}, q_2^{ini}, \varnothing)$ to node (q_1, q_2, M) in $\mathcal{W}(S_1, S_2, \bowtie)$.

3. Let $n > 1$ and assume that the lemma holds for all alignable traces and respective tactics with $n - 1$ equivalence classes. We first introduce an auxiliary notation for this part of the proof: Let w be some trace with $|w| > 1$ and θ be a tactic of w. Let $\mathrm{cut}(\theta)$ denote the maximal index in $\{1, \dots, |w|\}$ with $\theta(i) \neq \theta(|w|)$. We observe $1 \leq \mathrm{cut}(\theta) < |w|$. Now, for $i \in \{1,2\}$, let $\hat{w}_i = w_i(1) \dots w_i(\mathrm{cut}(\theta_i))$ and $\hat{\theta}_i = \theta_i(1) \dots \theta_i(\mathrm{cut}(\theta_i))$. Let $\hat{b} = b \setminus \{(\langle|w_1|\rangle_{\theta_1}, \langle|w_2|\rangle_{\theta_2})\}$. Then, $\hat{b}$ proves $(\hat{\theta}_1, \hat{\theta}_2) : \hat{w}_1 \bowtie \hat{w}_2$, and we observe $|\{1, \dots, \hat{w}_i\}/\theta_i| = n - 1$ for $i \in \{1,2\}$. Let for $i \in \{1,2\}$, $\hat{\theta}_i(\mathrm{cut}(\theta_i)) = \{\hat{K}_i\}$. By assumption, $(\hat{w}_1, \hat{w}_2)$ is realisable resulting in a node $(\hat{q}_1, \hat{q}_2, \hat{M})$ with $(\hat{K}_1, \hat{K}_2) \in \hat{M}$. By assumption, for $i \in \{1,2\}$, there exists a label λ_i, and a transition $\hat{q}_i \xrightarrow{\lambda_i} q_i$ with $\mathcal{G}_{\bowtie}^{i}(\lambda_i) = w_i(\mathrm{cut}(\theta_i) + 1)$ and $K_i = \theta_i(\mathrm{cut}(\theta_i) + 1) \in \mathcal{G}_{\bowtie}^{i}(\lambda_i)$. We distinguish the cases $(K_1, K_2) \notin \hat{M}$ and $(K_1, K_2) \in \hat{M}$.

(a) Let $(K_1, K_2) \notin \hat{M}$. Let $M = \hat{M} + (\mathcal{G}^1_{\bowtie}(\lambda_1), \mathcal{G}^2_{\bowtie}(\lambda_2))$. From $K_1 \bowtie K_2$, $(\hat{K}_1, \hat{K}_2) \in \hat{M}$, $\hat{K}_1 \neq K_1$ and $\hat{K}_2 \neq K_2$, we conclude $(K_1, K_2) \in M$ and $M \neq \emptyset$. Therefore, rule 2(a) produces an edge labelled $(\mathcal{G}^1_{\bowtie}(\lambda_1), \mathcal{G}^2_{\bowtie}(\lambda_2))$ from node $(\hat{q}_1, \hat{q}_2, \hat{M})$ to the node (q_1, q_2, M).

(b) Let $(K_1, K_2) \in \hat{M}$. Let $M' = \hat{M} + (\mathcal{G}^1_{\bowtie}(\lambda_1), \emptyset)$ and $M = M' + (\emptyset, \mathcal{G}^2_{\bowtie}(\lambda_2))$. Then, $(K_1, K_2) \in M' \cap M$, and hence $M' \neq \emptyset \neq M$. From $M' \neq \emptyset$ and $M \neq \emptyset$, we conclude that rule 2(b) produces an edge labelled $(\mathcal{G}^1_{\bowtie}(\lambda_1), \emptyset)$ from $(\hat{q}_1, \hat{q}_2, \hat{M})$ to $(q_1, \hat{q}_2, M')$, and an edge labelled $(\emptyset, \mathcal{G}^2_{\bowtie}(\lambda_2))$ from $(q_1, \hat{q}_2, M')$ to (q_1, q_2, M).

If for $i \in \{1, 2\}$, $\mathrm{cut}(\theta_i) = |w_i| - 1$, we are finished. Otherwise, rule 2(b) produces the remaining edges: We first add the edges for the remaining elements of w_1, and then for the remaining elements of w_2, resulting in edges labelled

$$(\, w_1(\mathrm{cut}(\theta_i) + 2), \emptyset \,) \ldots (\, w_1(|w_i|), \emptyset \,) \;\; (\, \emptyset, w_2(\mathrm{cut}(\theta_i) + 2) \,) \ldots (\, \emptyset, w_2(|w_2|) \,)$$

and nodes $(q_1^1, q_2, M_1) \ldots (q_1^\ell, q_2, M_\ell)(q_1^\ell, q_2^1, M_{\ell+1}) \ldots (q_1^\ell, q_2^m, M_{\ell+m})$, where for each $1 \le j \le \ell + m$, we exploit $(K_1, K_2) \in M_j$.

■

Lemma 7.8. *The following statements are equivalent:*

1. $\mathcal{L}_1(\mathcal{W}(S_1, S_2, \bowtie)) = \mathcal{T}^1_{\bowtie}(\mathcal{L}(S_1))$ *and* $\mathcal{L}_2(\mathcal{W}(S_1, S_2, \bowtie)) = \mathcal{T}^2_{\bowtie}(\mathcal{L}(S_2))$.
2. $\mathcal{L}(S_1) \doteqdot_{\bowtie} \mathcal{L}(S_2)$.

⌟

Proof. We show both directions separately.

1. "$1 \Rightarrow 2$": We show: For each $w_1 \in \mathcal{T}^1_{\bowtie}(\mathcal{L}(S_1))$, there exists $w_2 \in \mathcal{T}^2_{\bowtie}(\mathcal{L}(S_2))$, and tactics θ_1 and θ_2 of w_1 and w_2, respectively, such that there is b proving $(\theta_1, \theta_2) : w_1 \bowtie w_2$. From $w_1 \in \mathcal{T}^1_{\bowtie}(\mathcal{L}(S_1))$ and the assumption, we get $w_1 \in \mathcal{L}_1(\mathcal{W}(S_1, S_2, \bowtie))$. Hence, there exists a path π of $\mathcal{W}(S_1, S_2, \bowtie)$, such that $\pi[i]_{\neq\emptyset} = w_1$, resulting in a node (q_1, q_2, M) satisfying $q_1 \in F_1$ and $q_2 \in F_2$. Let $w_2 = \pi[2]_{\neq\emptyset}$. Then, by definition, (w_1, w_2) are realisable. Applying Lemma 7.5 yields the existence of required θ_1, θ_2 and b. It now remains to be shown that $w_2 \in \mathcal{T}^2_{\bowtie}(\mathcal{L}(S_2))$, which follows directly from the assumption.
2. "$2 \Rightarrow 1$":
   - We show $\mathcal{L}_1(\mathcal{W}(S_1, S_2, \bowtie)) \subseteq \mathcal{T}^1_{\bowtie}(\mathcal{L}(S_1))$. Let $w_1 \in \mathcal{L}_1(\mathcal{W}(S_1, S_2, \bowtie))$. Hence, there exists a path π of $\mathcal{W}(S_1, S_2, \bowtie)$, such that $\pi[1]_{\neq\emptyset} = w_1$, resulting in a node (q_1, q_2, M) satisfying $q_1 \in F_1$ and $q_2 \in F_2$. Now, we can show by induction, that there exists an accepting path of S_1 accepting some σ_1 with $\mathcal{T}^1_{\bowtie}(\sigma_1) = w_1$.
   - We show $\mathcal{L}_1(\mathcal{W}(S_1, S_2, \bowtie)) \supseteq \mathcal{T}^1_{\bowtie}(\mathcal{L}(S_1))$ and $\mathcal{L}_2(\mathcal{W}(S_1, S_2, \bowtie)) \supseteq \mathcal{T}^2_{\bowtie}(\mathcal{L}(S_2))$. Let $w_1 \in \mathcal{T}^1_{\bowtie}(\mathcal{L}(S_1))$. Then, by assumption, there exists $w_2 \in \mathcal{T}^2_{\bowtie}(\mathcal{L}(S_2))$, θ_1, θ_2 and b proving $(\theta_1, \theta_2) : w_1 \bowtie w_2$. Hence, by Lemma 7.6, (w_1, w_2) is realisable, yielding some node (q_1, q_2, M). Hence, there exists a path π with $\pi[1]_{\neq\emptyset} = w_1$ and $\pi[2]_{\neq\emptyset} = w_2$. Because for $i \in \{1, 2\}$, $w_i \in \mathcal{T}^i_{\bowtie}(\mathcal{L}(S_i))$, it holds that $q_i \in F_i$. Hence, for $i \in \{1, 2\}$, $w_i \in \mathcal{L}_i(\mathcal{W}(S_1, S_2, \bowtie))$.

■

Lemma 7.9. *For $i \in \{1, 2\}$, the following problem is decidable:*
To decide whether it holds that $\mathcal{L}_i(\mathcal{W}(S_1, S_2, \bowtie)) = \mathcal{T}^i_{\bowtie}(\mathcal{L}(S_i))$.

⌟

Proof. $\mathcal{W}(S_1, S_2, \bowtie)$ is obviously finite and computable. Further, one can construct an FSM S accepting $\mathcal{L}_i(\mathcal{W}(S_1, S_2, \bowtie))$, $i \in \{1, 2\}$, by following this procedure:

1. Each node of $\mathcal{W}(S_1, S_2, \bowtie)$ is a state of S.
2. For each edge $(v, \mathcal{K}_1, \mathcal{K}_2, v')$ of $\mathcal{W}(S_1, S_2, \bowtie)$ introduce a transition $v \xrightarrow{K_i}_S v'$,
3. Set the node $(q_1^{ini}, q_2^{ini}, \emptyset)$ as the initial state.
4. A state (q_1, q_2, M) is a final state of S iff $q_1 \in F_1$ and $q_2 \in F_2$.
5. Finally, remove all $\emptyset$-transitions with a standard ε-removal algorithm [21].

We decide $\mathcal{L}_i(\mathcal{W}(S_1, S_2, \bowtie)) = \mathcal{T}^i_{\bowtie}(\mathcal{L}(S_i))$ by checking language equivalence of S and S_i. ■

Appendix B. Example for Lemma 6.3

This section introduces an example for the decision procedure explained in the proof of Lemma 6.3. We consider the example alignment $\bowtie$ given by Figure 1, and the $\mathsf{LTL}[\overset{2}{\bowtie}]$-formula $\varphi = \{t,u\}\,\mathsf{U}\,(\{s,v\} \vee \{s,w,x\})$. Then, S_φ consists of an initial state q_1, a final state q_2, for each $\mathcal{K} \subseteq \overset{2}{\bowtie}$ a transition $q_2 \xrightarrow{\mathcal{K}} q_2$ (in the following abbreviated as $q_2 \xrightarrow{*} q_2$), and for each $\mathcal{K} \subseteq \overset{2}{\bowtie}$ satisfying $\{s,v\} \in \mathcal{K}$ implies $\{s,w,x\} \notin \mathcal{K}$ a transition $q_1 \xrightarrow{\mathcal{K}} q_2$, namely:

$$q_1 \xrightarrow{\varnothing} q_2, \quad q_1 \xrightarrow{\{\{t,u\}\}} q_2, \quad q_1 \xrightarrow{\{\{s,v\}\}} q_2,$$
$$q_1 \xrightarrow{\{\{s,v\},\{t,u\}\}} q_2, \quad q_1 \xrightarrow{\{\{s,w,x\}\}} q_2, \quad q_1 \xrightarrow{\{\{s,w,x\},\{t,u\}\}} q_2 \ .$$

S_W has the same initial and final states, and the following transitions:

$$q_1 \xrightarrow{\{\{t,u\}\}} q_2, \quad q_1 \xrightarrow{\{\{s,v\}\}} q_2, \quad q_1 \xrightarrow{\{\{s,w,x\}\}} q_2,$$
$$q_2 \xrightarrow{\{\{t,u\}\}} q_2, \quad q_2 \xrightarrow{\{\{s,v\}\}} q_2, \quad q_2 \xrightarrow{\{\{s,w,x\}\}} q_2, \quad q_2 \xrightarrow{\{\{s,v\},\{s,w,x\}\}} q_2 \ .$$

S_Θ has the same initial and final states, and the following transitions:

$$q_1 \xrightarrow{\{\{t,u\}\}} q_2, \quad q_1 \xrightarrow{\{\{s,v\}\}} q_2, \quad q_1 \xrightarrow{\{\{s,w,x\}\}} q_2,$$
$$q_2 \xrightarrow{\{\{t,u\}\}} q_2, \quad q_2 \xrightarrow{\{\{s,v\}\}} q_2, \quad q_2 \xrightarrow{\{\{s,w,x\}\}} q_2 \ .$$

$\hat{S}_\Theta$ has the initial and final states, and the following transitions:

$$q_1 \xrightarrow{(\{\{t,u\}\},\{\{t,u\}\})} q_2, \quad q_1 \xrightarrow{(\{\{s,v\}\},\{\{s,v\}\})} q_2, \quad q_1 \xrightarrow{(\{\{s,w,x\}\},\{\{s,w,x\}\})} q_2,$$
$$q_2 \xrightarrow{(\{\{t,u\}\},\{\{t,u\}\})} q_2, \quad q_2 \xrightarrow{(\{\{s,v\}\},\{\{s,v\}\})} q_2, \quad q_2 \xrightarrow{(\{\{s,v\}\},\{\{s,v\},\{s,w,x\}\})} q_2,$$
$$q_2 \xrightarrow{(\{\{s,w,x\}\},\{\{s,w,x\}\})} q_2, \quad q_2 \xrightarrow{(\{\{s,w,x\}\},\{\{s,v\},\{s,w,x\}\})} q_2 \ .$$

$\hat{S}'_\Theta$ evolves from $\hat{S}_\Theta$ by adding the following transitions:

$$q_1 \xrightarrow{(\{\{s,v\}\},\{\{s,v\},\{s,w,x\}\})} q_2, \quad q_1 \xrightarrow{(\{\{s,w,x\}\},\{\{s,v\},\{s,w,x\}\})} q_2.$$

It is obvious that $\hat{S}'_\Theta$ accepts the language of $\hat{S}_\Theta$, and, in addition, all those sequences that start with $(\{\{s,v\}\},\{\{s,v\},\{s,w,x\}\})$ and $(\{\{s,w,x\}\},\{\{s,v\},\{s,w,x\}\})$, respectively. Therefore, φ is not tactic-invariant.