Compositionality, Decompositionality and Refinement in Input/Output Conformance Testing – Technical Report

Lars Luthmann
Real-Time Systems Lab
TU Darmstadt, Germany
lars.luthmann@es.tu-darmstadt.de

Stephan Mennicke
Institute for Programming and Reactive Systems
TU Braunschweig, Germany
mennicke@ips.cs.tu-bs.de

Malte Lochau
Real-Time Systems Lab
TU Darmstadt, Germany
malte.lochau@es.tu-darmstadt.de

We propose an input/output conformance testing theory utilizing Modal Interface Automata with Input Refusals (IR-MIA) as novel behavioral formalism for both the specification and the implementation under test. A modal refinement relation on IR-MIA allows distinguishing between obligatory and allowed output behaviors, as well as between implicitly underspecified and explicitly forbidden input behaviors. The theory therefore supports positive and negative conformance testing with optimistic and pessimistic environmental assumptions. We further show that the resulting conformance relation on IR-MIA, called modal-irioco, enjoys many desirable properties concerning component-based behaviors. First, modal-irioco is preserved under modal refinement and constitutes a preorder under certain restrictions which can be ensured by a canonical input completion for IR-MIA. Second, under the same restrictions, modal-irioco is compositional with respect to parallel composition of IR-MIA with multi-cast and hiding. Finally, the quotient operator on IR-MIA, as the inverse to parallel composition, facilitates decompositionality in conformance testing to solve the unknown-component problem.

1 Introduction

Formal approaches to model-based testing of component-based systems define notions of behavioral conformance between a specification and a (black-box) implementation (under test), both usually given as (variations of) labeled transition systems (LTS). Existing notions of behavioral conformance may be categorized into two research directions. Extensional approaches define observational equivalences, requiring that no observer process (tester) is ever able to distinguish behaviors shown by the implementation from those allowed by the specification [17]. In contrast, intensional approaches rely on I/O labeled transition systems (IOLTS) from which test cases are derived as sequences of controllable input and observable output actions, to establish an alternating simulation relation on IOLTS [26] [11]. One of the most prominent conformance testing theories, initially introduced by Tretmans in [24], combines both views on formal conformance testing into an input/output conformance (ioco) relation on IOLTS. Although many formal properties of, and extensions to, ioco have been intensively investigated, ioco still suffers several essential weaknesses.

- The ioco relation permits underspecification by means of (1) unspecified input behaviors and (2) non-deterministic input/output behaviors. But, concerning (1), ioco is limited to positive testing (i.e., unspecified inputs may be implemented arbitrarily) thus implicitly relying on optimistic environmental assumptions. Also supporting negative testing in a pessimistic setting, however,
would require a distinction between critical and uncritical unintended input behaviors. Concerning (2), ioco requires the implementation to exhibit at most output behaviors permitted by the specification. In addition, the notion of quiescence (i.e., observable absence of any outputs) enforces implementations to show at least one specified output behavior (if any). Apart from that, no explicit distinction between obligatory and allowed output behaviors is expressible in IOLTS.

- ioco imposes a special kind of alternating simulation between specification and implementation which is, in general, not a preorder, although being a crucial property for testing relations on LTS [18].
- ioco lacks a unified theory for input/output conformance testing in the face of component-based behaviors being compatible with potential solutions for the aforementioned weaknesses.

As all these weaknesses mainly stem from the limited expressiveness of IOLTS as behavioral formalism, we propose Modal Interface Automata with Input Refusals (IR-MIA) as a new model for input/output conformance testing for both the specification and the implementation under test. IR-MIA adopt Modal Interface Automata (MIA) [7], which combine concepts of Interface Automata [11] (i.e., I/O automata permitting underspecified input behaviors) and (I/O-labeled) Modal Transitions Systems [13, 32] (i.e., LTS with distinct mandatory and optional transition relations). In particular, we exploit enhanced versions of MIA supporting both optimistic and pessimistic environmental assumptions [15] and non-deterministic input/output behaviors [7]. For the latter, we have to re-interpret the universal state of MIA, simulating every possible behavior, as failure state to serve as target for those unintended, yet critical input behaviors to be refused by the implementation [21]. Modal refinement of IR-MIA therefore allows distinguishing between obligatory and allowed output behaviors, as well as between implicitly underspecified and explicitly forbidden input behaviors.

The resulting testing theory on IR-MIA unifies positive and negative conformance testing with optimistic and pessimistic environmental assumptions. We further prove that the corresponding modal I/O conformance relation on IR-MIA, called modal-irioco, exhibits essential properties, especially with respect to component-based systems testing.

- modal-irioco is preserved under modal refinement and constitutes a preorder under certain restrictions which can be obtained by a canonical input completion [23].
- modal-irioco is compositional with respect to parallel composition of IR-MIA with multi-cast and hiding [7].
- modal-irioco allows for decomposition of conformance testing, thus supporting environmental synthesis for component-based testing in contexts [20, 11], also known as the unknown-component problem [27]. To this end, we adapt the MIA quotient operator to IR-MIA, serving as the inverse to parallel composition.

The remainder of this paper is organized as follows. In Sect. 2, we revisit the foundations of ioco testing. In Sect. 3, we introduce IR-MIA and modal refinement on IR-MIA and, thereupon, define modal-irioco, provide a correctness proof and discuss necessary restrictions to obtain a preorder. Our main results concerning compositionality and decompositionality of modal-irioco are presented in Sect. 4 and Sect. 5 respectively. In Sect. 6, we discuss related work and in Sect. 7, we conclude the paper. Please note that all proofs may be found in Appendix A.
2 Preliminaries

The ioco testing theory relies on I/O-labeled transition systems (IOLTS) as behavioral formalism [24]. An IOLTS \((Q, I, O, \rightarrow)\) specifies the externally visible behaviors of a system or component by means of a transition relation \(\rightarrow \subseteq Q \times (I \cup O \cup \{\tau\}) \times Q\) on a set of states \(Q\). The set of transition labels \(A = I \cup O\) consists of two disjoint subsets: set \(I\) of externally controllable/internally observable input actions, and set \(O\) of internally controllable/externally observable output actions. In figures, we use prefix \(\tau\) to mark input actions and prefix \(!\) for output actions, respectively. In addition, transitions labeled with internal actions \(\tau \notin (I \cup O)\) denote silent moves, neither being externally controllable, nor observable. We write \(A^* = A \cup \{\tau\}\), and by \(q \xrightarrow{\alpha} q'\) we denote that \((q, \alpha, q') \in \rightarrow\) holds, where \(\alpha \in A^*\), and we write \(q \xrightarrow{\alpha}\) as a short hand for \(\exists q' \in Q : q \xrightarrow{\alpha} q'\) and \(q \xrightarrow{\cdot\alpha}\), else. Furthermore, we write \(q \xrightarrow{\alpha_1 \cdots \alpha_n} q'\) to express that \(\exists q_0, \ldots, q_n \in Q : q = q_0 \xrightarrow{\alpha_0} q_1 \xrightarrow{\alpha_1} \cdots q_n = q'\) holds, and write \(q \xrightarrow{\epsilon} q'\) whenever \(q = q'\) or \(q \tau \rightarrow q'\). Additionally, by \(q \xrightarrow{\epsilon}\), we denote that \(\exists q_1, q_2 : q \xrightarrow{\epsilon} q_1 \xrightarrow{\alpha} q_2 \xrightarrow{\epsilon} q'\). We further use the notations \(q \xrightarrow{a_1 \cdots a_n} q'\) and \(q \xrightarrow{a} (a, a_1, \ldots, a_n \in A^*)\) analogously to \(q \xrightarrow{\alpha_1 \cdots \alpha_n} q'\) and \(q \xrightarrow{\alpha}\). Finally, by \(q_0 \xrightarrow{\alpha_1} q_1 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_n} q_n\) we denote a path, where \(\alpha = \alpha_1 \alpha_2 \cdots \alpha_n \in A^*\) is called a trace (note: \(\tau\) equals \(\epsilon\)). We identify an IOLTS with its initial state (i.e., \(q \in Q\) is the initial state of \(q = (Q, I, O, \rightarrow)\)). We only consider strongly convergent IOLTS (i.e., no infinite \(\tau\)-sequences exist).

In the ioco testing theory, both specification \(s\) as well as a (black-box) implementation under test \(i\) are assumed to be (explicitly or implicitly) given as IOLTS. In particular, ioco does not necessarily require specification \(s\) to be input-enabled, whereas implementation \(i\) is assumed to never reject any input \(a \in I\) from the environment (or tester). More precisely, ioco requires implementations to be weak input-enabled (i.e., \(\forall q \in Q : \exists a \in I : q \xrightarrow{a}\)) thus yielding the subclass of I/O transition systems (IOTS). Intuitively, the IOTS of implementation \(i\) I/O-conforms to the IOLTS of specification \(s\) if all output behaviors of \(i\) observed after any possible sequence \(\sigma = \alpha_1 \cdots \alpha_n\) in \(s\) are permitted by \(s\). In case of nondeterminism, more than one state may be reachable in \(i\) as well as in \(s\) after sequence \(\sigma\) and therefore all possible outputs of any state in the set

\[
\mathit{p_{after}}(\sigma) := \{q \in Q \mid p \xrightarrow{\sigma} q\}
\]

have to be taken into account. Formally, set \(\mathit{Out}(Q') \subseteq O\) denotes all output actions being enabled in any possible state \(q \in Q' = \mathit{p_{after}}(\sigma)\). To further reject trivial implementations never showing any outputs, the notion of quiescence has been introduced by means of a special observable action \(\delta\) explicitly denoting the permission of the absence (suspension) of any output in a state \(p\), thus requiring an input to proceed. In particular, \(p\) is quiescent, denoted \(\mathit{\delta}(p)\), iff

\[
\mathit{init}(p) := \{\alpha \in (I \cup O \cup \{\tau\}) \mid p \xrightarrow{\alpha}\} \subseteq I
\]

holds. Thereupon, we denote

\[
\mathit{Out}(P) := \{\alpha \in O \mid \exists p \in P : p \xrightarrow{\alpha}\} \cup \{\delta \mid \exists p \in P : \mathit{\delta}(p)\},
\]

where symbol \(\delta\) is used both as action as well as a state predicate. Based on these notions, I/O conformance is defined with respect to the set of suspension traces

\[
\mathit{Straces}(s) := \{\sigma \in (I \cup O \cup \{\delta\})^* \mid p \xrightarrow{\sigma}\}
\]

of specification \(s\), where \(q \xrightarrow{\delta} q\) iff \(\delta(q)\).

**Definition (ioco [24]).** Let \(s\) be an IOLTS and \(i\) an IOTS with identical sets \(I\) and \(O\).

\[
i\mathit{ioco} s \iff \forall \sigma \in \mathit{Straces}(s) : \mathit{Out}(i \mathit{after} \sigma) \subseteq \mathit{Out}(s \mathit{after} \sigma).
\]
3 Modal Input/Output Conformance with Input Refusals

IOLTS permit specifications $s$ to be underspecified by means of unspecified input behaviors and non-deterministic input/output behaviors. In particular, if $q \xrightarrow{a} q'$, then no proper reaction on occurrences of input $a \in I$ is specified while residing in state $q$. Moreover, if $q \xrightarrow{a} q'$ and $q \xrightarrow{a^\prime} q''$, $a \in A^\tau$, it does not necessarily follow that $q' = q''$ and if $q \xrightarrow{a} q'$ and $q \xrightarrow{a''} q''$ with $a', a'' \in O$, it does not necessarily follow that $a' = a''$ (i.e., IOLTS are neither input-, nor output-deterministic). In this way, Ioco permits, at least up to a certain degree, implementation freedom in two ways. First, in case of input behaviors being unspecified in $s$, Ioco solely relies on positive testing principles, i.e., reactions to unspecified input behaviors are never tested and may therefore show arbitrary output behaviors if ever applied to $i$. Second, in case of non-deterministic specifications, implementation $i$ is allowed to show any, but at least one of those output behaviors being permitted by $s$ (if any), or it must be quiescent, else. These limitations of Ioco in handling underspecified behaviors essentially stem from the limited expressive power of IOLTS. To overcome these limitations, we propose to adopt richer specification concepts from interface theories [22] to serve as novel formal foundation for I/O conformance testing. In particular, we replace IOLTS by a modified version of (I/O-labeled) Modal Interface Automata (MIA) with universal state [7]. Similar to IOLTS, MIA also support both kinds of underspecification but allow for explicit distinctions (1) between obligatory and allowed behaviors in case of non-deterministic input/output behaviors, and (2) between critical and non-critical unspecified input behaviors.

Concerning (1), MIA separate mandatory from optional behaviors in terms of may/must transition modality. For every must-transition $q \xrightarrow{a,\square} q'$, a corresponding may-transition $q \xrightarrow{a,\Diamond} q'$ exists, as mandatory behaviors must also be allowed (so-called syntactic consistency). Conversely, may-transitions $q \xrightarrow{a,\Diamond} q'$ for which $q \xrightarrow{a,\Diamond} q'$ holds constitute optional behaviors. Accordingly, we call may-transitions without corresponding must-transitions optional, else mandatory.

Concerning (2), MIA make explicit input actions $a \in I$ being unspecified, yet uncritical in a certain state $q$ by introducing may-transitions $q \xrightarrow{a,\Diamond} u$ leading to a special universal state $u$ (permitting any possible behavior following that input). In contrast, unintended input actions to be rejected in a certain state are implicitly forbidden if $q \xrightarrow{a,\Diamond} q$. We alter the interpretation of unspecified input behaviors of MIA by introducing a distinct failure state $q_\Phi$, replacing $u$. As a consequence, an unspecified input $a \in I$ being uncritical if residing in a certain state $q$ is (similar to IOLTS) implicitly denoted as $q \xrightarrow{a,\Diamond} q$, whereas inputs $a' \in I$ being critical while residing in state $q$ are explicitly forbidden by $q \xrightarrow{a',\Box} q_\Phi$. We therefore enrich I/O conformance testing by the notion of input refusals in the spirit of refusal testing, initially proposed by Phillips for testing preorders on LTS with undirected actions [21]. Analogous to quiescence, denoting the observable absence of any output in a certain state, refusals therefore denote the observable rejection of a particular input in a certain state during testing. In this way, we unify positive testing (i.e., unspecified behaviors are ignored) and negative testing (i.e., unspecified behaviors must be rejected) with optimistic and pessimistic environmental assumptions known from interface theories [22]. In particular, we are now able to explicitly reject certain input behavior, which is not supported by Ioco. We refer to the resulting model as Modal Interface Automata with Input Refusals (IR-MIA).

**Definition 2** (IR-MIA). A Modal Interface Automaton with Input-Refusal (IR-MIA or MIA$_\Phi$) is a tuple $(\mathcal{Q}, I_\mathcal{Q}, O_\mathcal{Q}, \xrightarrow{\square}, \xrightarrow{\Diamond}: \xrightarrow{\Diamond}, \xrightarrow{\Box}, q_\Phi)$, where $\mathcal{Q}$ is a finite set of states with failure state $q_\Phi \in \mathcal{Q}$, $A_\mathcal{Q} = I_\mathcal{Q} \cup O_\mathcal{Q}$ is a finite set of actions with $\tau \notin A_\mathcal{Q}$ and $I_\mathcal{Q} \cap O_\mathcal{Q} = \emptyset$ and for all $a \in A_\mathcal{Q} \cup \{\tau\}, i \in I_\mathcal{Q}$,

1. $\xrightarrow{\square} \subseteq ((\mathcal{Q} \setminus \{q_\Phi\}) \times I_\mathcal{Q} \times \mathcal{Q}) \cup ((\mathcal{Q} \setminus \{q_\Phi\}) \times (O_\mathcal{Q} \cup \{\tau\}) \times ((\mathcal{Q} \setminus \{q_\Phi\}))$,

2. $\xrightarrow{\Diamond} \subseteq ((\mathcal{Q} \setminus \{q_\Phi\}) \times I_\mathcal{Q} \times \mathcal{Q}) \cup ((\mathcal{Q} \setminus \{q_\Phi\}) \times (O_\mathcal{Q} \cup \{\tau\}) \times ((\mathcal{Q} \setminus \{q_\Phi\}))$,
is an
Definition 3
P
Adapted to IR-MIA, input behaviors being unspecified in
Q
and even
Q
fines MIA
by state
mandatory behaviors. Additionally, the distinct state
further requires consistency of refusals of specified input actions in every state (i.e., each input is either
forbidden or not, but not both in a state q).

Figure 1 shows a sample IR-MIA. Dashed lines denote optional behaviors and solid lines denote
mandatory behaviors. This example also exhibits input non-determinism (state q0 defines two possible
reactions to input a), as well as output non-determinism (state q3 defines two possible outputs after input d).

Modal refinement provides a semantic implementation relation on MIA [7]. Intuitively, MIA P refine
MIA Q if mandatory behaviors of Q are preserved in P and optional behaviors in P are permitted by Q.
Adapting to IR-MIA, input behaviors being unspecified in Q, may be either implemented arbitrarily in
P, or become forbidden after refinement. In particular, if \( q \xrightarrow{a} \square q' \) holds in Q, then either \( q \xrightarrow{a} \diamond q' \) (and even \( q \xrightarrow{a} \square q' \)), or \( q \xrightarrow{a} \square q_\Phi \) holds in P, respectively.

**Definition 3 (IR-MIA Refinement).** Let \( P, Q \) be MIA\_\( P \) with \( I_P = I_Q \) and \( O_P = O_Q \). A relation \( \mathcal{R} \subseteq P \times Q \) is an IR-MIA Refinement Relation if for all \( (p, q) \in \mathcal{R} \) and \( \omega \in (O \cup \{ \tau \}) \), with \( p \neq p_\Phi \) and \( \gamma \in \{ \diamond, \square \} \), it holds that

1. \( q \neq q_\Phi \).
2. \( q \xrightarrow{i_\square} q' \neq q_\Phi \) implies \( \exists p'. p \xrightarrow{i_\square} p' \neq p_\Phi \) and \( (p', q') \in \mathcal{R} \),
3. \( q \xrightarrow{\omega_\square} q' \) implies \( \exists p'. p \xrightarrow{\omega_\square} p' \) and \( (p', q') \in \mathcal{R} \),
4. \( p \xrightarrow{i_\diamond} p' \land q \xrightarrow{i_\diamond} q' \) implies \( \exists q'. q \xrightarrow{i_\diamond} q' \) and \( (p', q') \in \mathcal{R} \),
5. \( q \xrightarrow{i_\diamond} q' \) implies \( \exists p'. p \xrightarrow{i_\diamond} p' \) and \( (p', q') \in \mathcal{R} \), and
6. \( p \xrightarrow{\omega_\diamond} p' \) implies \( \exists q'. q \xrightarrow{\omega_\diamond} q' \) and \( (p', q') \in \mathcal{R} \).

State p refines state q if there exists \( \mathcal{R} \) such that \( (p, q) \in \mathcal{R} \). (Note: \( q \xrightarrow{\epsilon} q' \) equals \( q \xrightarrow{\epsilon} p \xrightarrow{\gamma} q' \).)

**Figure 1:** Sample IR-MIA
Clause 1 ensures that the failure state $q_φ$ can only be refined by $p_φ$, since both suspend any subsequent behavior. Clauses 2 and 3 guarantee that mandatory behavior of $Q$ is preserved by $P$. All other clauses handle optional behavior, where inputs are either refined to forbidden or mandatory inputs, and outputs are either refined to mandatory or unspecified outputs. By $P \sqsubseteq_φ Q$, we denote the existence of an IR-MIA refinement relation between $P$ and $Q$.

As an example, consider IR-MIA $Q$ and $Q'$ in Fig. 1. $Q' \sqsubseteq_φ Q$ does not hold as the mandatory output $e$ of $q_4$ in $Q$ is not mandatory anymore in $Q'$. However, the other modifications in $Q'$ are valid refinements of $Q$ as output $g$ of $q_3$ has become mandatory, and optional input $f$ of $q_1$ is now refused (i.e., the transition is redirected to $q_φ$). Additionally, inputs being unspecified in $Q$ may be added to $Q'$ (e.g., $q_4$ of $Q'$ now accepts input $d$). Furthermore, internal steps may be added after inputs as well as before and after outputs under refinement (e.g., $Q'$ has a $τ$ step after output $e$ in $q_3$). The former ensures that a refined IR-MIA can be controlled by the environment in the same way as the unrefined IR-MIA. Considering IR-MIA $Q''$ in Fig. 1 instead, $Q'' \sqsubseteq_φ Q$ holds. The removal of mandatory output $e$ from $q_4$ is valid as $q_4$ is not reachable anymore after refinement.

In the context of modal I/O conformance testing, modal refinement offers a controlled way to resolve underspecification within specifications $s$. In addition, we also assume $s$ to be represented as IR-MIA in order to support (partially) underspecified implementations under test as apparent in earlier phases of continuous systems and component development.

We next define an adapted version of $\text{ioco}$ to operate on IR-MIA. Intuitively, a modal implementation $i$ I/O-conforms to a modal specification $s$ if all observable mandatory behaviors of $s$ are also observable as mandatory behaviors of $i$ and none of the observable optional behaviors of $i$ exceed the observable optional behaviors of $s$. If established between implementation $i$ and specification $s$, modal I/O conformance ensures for all implementations $i' \sqsubseteq_φ i$, derivable from $i$ via modal refinement, the existence of an accompanying specification refinement $s' \sqsubseteq_φ s$ of $s$ such that $i'$ is I/O conforming to $s'$.

Similar to $δ$ denoting observable quiescence, we introduce a state predicate $φ$ to denote may-failure/must-failure states (i.e., states having may/must input-transitions leading to $q_φ$). We therefore use $φ$ as a special symbol to observe refusals of particular inputs in certain states of the implementation during testing. To this end, we first lift the auxiliary notations of $\text{ioco}$ from IOLTS to IR-MIA, where we write $γ \in \{\Diamond, \Box\}$ for short in the following.

**Definition 4.** Let $Q$ be a $\text{MA}_φ$ over $I$ and $O$, $p \in Q$ and $σ \in (I \cup O \cup \{δ, φ\})^*$.

- $\text{init}_γ(p) := \{μ \in (I \cup O) \mid p \xrightarrow{μ} γ\} \cup \{φ \mid p = p_φ\}$.
- $p$ is may-quiescent, denoted by $δ_φ(p)$, iff $\text{init}_γ(p) \subseteq I$ and $p \neq p_φ$.
- $p$ is must-quiescent, denoted by $δ_φ(p)$, iff $\text{init}_γ(p) \subseteq I$ and $p \neq p_φ$.
- $p$ is may-failure, denoted by $φ_φ(p)$, iff $p = p_φ$ or $∃p' \in Q : (p'' \xrightarrow{i} φ_φ(p) \wedge p'' \xrightarrow{φ_φ(p)} p')$.
- $p$ is must-failure, denoted by $φ_φ(p)$, iff $p = p_φ$.
- $p_{after}_γ σ := \{p' \mid p \xrightarrow{σ}_γ p'\}$.
- $\text{Out}_γ(p) := \{μ \in O \mid p \xrightarrow{μ} γ\} \cup \{δ \mid δ_φ(p) \cup \{φ \mid φ_φ(p)\}\} \cup \{φ \mid φ_φ(p)\}$, and
- $\text{Straces}_γ(p) := \{σ \in (I \cup O \cup \{δ, φ\})^* \mid p \xrightarrow{σ}_γ\}$, where $p \xrightarrow{δ_φ(p)} p$ if $δ_φ(p)$, and $p \xrightarrow{φ_φ(p)} p$ if $φ_φ(p)$.

Hence, quiescence as well as failure behaviors may occur with both may- and must-modality. Intuitively, a state is may-quiescent if all enabled output transitions are optional, i.e., such a state may become quiescent under refinement. Likewise, a state $p$ is a may-failure if there is an optional input leading to $p$, since this optional input may be refused under refinement.
According to \textit{ioco}, MIA$_\Phi$ $i$ constituting a modal implementation under test is assumed to be input-enabled. In particular, modal input-enabledness of IR-MIA comes in four flavors by combining weak/strong input-enabledness with may-/must-modality. Note, that $q \xrightarrow{i} q'$ implies $q \xRightarrow{\gamma} q'$, $q \xRightarrow{\square} q'$ implies $q \xRightarrow{\Diamond} q'$, and $q \xrightarrow{i} q'$ implies $q \xRightarrow{\Diamond} q'$.

**Definition 5** (Input-Enabled IR-MIA). MIA$_\Phi$ $Q$ is weak/strong $\gamma$-input-enabled, respectively, iff for each $q \in Q \setminus \{q_\Phi\}$ it holds that $\forall i \in I : \exists q' \in Q : q \xRightarrow{\gamma} q'$, or $\forall i \in I : \exists q' \in Q : q \xrightarrow{i} q'$.

May-input-enabledness is preserved under modal refinement as optional input behaviors either remain optional, become mandatory, or are redirected to the failure state (and finally become must-input-enabled under complete refinement).

**Lemma 1.** If MIA$_\Phi$ $i$ is strong may-input-enabled then $i' \sqsubseteq_\Phi i$ is strong may-input-enabled.

We now define a modal version of \textit{ioco} on IR-MIA (called \textit{modal-ioco} or \textit{mioco$_\Phi$}), by means of alternating suspension-trace inclusion.

**Definition 6** (modal-ioco). Let $s$ and $i$ be MIA$_\Phi$ over $I$ and $O$ with $i$ being weak may-input-enabled. $\textit{mioco$_\Phi$}(s) :=$

1. $\forall \sigma \in \text{Straces}_\Phi^c(s) : \text{Out}_\Phi^c(i\text{after}_\Phi^c \sigma) \subseteq \text{Out}_\Phi^c(s\text{after}_\Phi^c \sigma)$, and
2. $\forall \sigma \in \text{Straces}_\Phi^c(i) : \text{Out}_\Phi^c(s\text{after}_\Phi^c \sigma) \subseteq \text{Out}_\Phi^c(i\text{after}_\Phi^c \sigma)$.

We illustrate the intuition of \textit{modal-ioco} by providing a concrete example. Let IR-MIA in Fig. 1b constitute implementation $i$ and IR-MIA in Fig. 1c constitute specification $s$. Similar to \textit{ioco}, Property 1 of \textit{modal-ioco} requires all possible output behaviors of $i$ to be permitted by $s$ which is satisfied in this example. Property 2 of \textit{modal-ioco} requires all mandatory outputs of $s$ to be actually implemented as mandatory outputs in $i$. This property does not hold in the example as mandatory output $e$ of $q_4$ in $s$ is not mandatory in $i$. As a consequence, $\textit{mioco$_\Phi$}(s)$ does not hold. The example in Fig. 1 also explains why we consider $\text{Straces}_\Phi^c$ and $\text{after}_\Phi^c$ in property 2 (unlike \textit{modal-ioco} in [14]). Otherwise, output $e$ of $q_4$ in $i$ would not be considered as mandatory output behavior because $q_4$ is not reachable via must-transitions. In contrast, when considering the IR-MIA in Fig. 1a as $i$ and the IR-MIA in Fig. 1c as $s$, we have $\textit{mioco$_\Phi$}(s)$ as the mandatory output $e$ of $q_4$ in $i$ is not reachable in $i$.

Finally, Fig. 2a, 2b and 2c illustrate the necessity for re-interpreting universal state $u$ of MIA [7] as failure state $q_\Phi$ in IR-MIA. The IR-MIA in Fig. 2a serves as implementation $i$, the MIA in Fig. 2b serves as specification $s_u$ with universal state, and the IR-MIA in Fig. 2c depicts the same specification with failure state $s_\Phi$ instead of $u$. Hence, $i$ would be (erroneously) considered to be non-conforming to $s_u$ as state $u$ does not specify any outputs (i.e., $u$ is quiescent). In contrast, we have $\textit{mioco$_\Phi$}(s_\Phi)$ as the reaction of $i$ to input $a$ is never tested, because this input is unspecified in $s_\Phi$.

An I/O conformance testing theory is correct if it is sound (i.e., every implementation $i$ conforming to specification $s$ does indeed only show specified behaviors), and complete (i.e., every erroneous implementation $i$ is rejected) [24]. For lifting these notions to IR-MIA, we relate \textit{modal-ioco} to \textit{ioco}. This way, we show compatibility of \textit{modal-ioco} and the original \textit{ioco} as follows.

- **modal-ioco** is sound if $\textit{mioco$_\Phi$}(s)$ implies that every refinement of $i$ conforms to a refinement of $s$ with respect to \textit{ioco}.

- **modal-ioco** is complete if the correctness of all refinements of $i$ regarding $s$ with respect to \textit{ioco} implies $\textit{mioco$_\Phi$}(s)$, and if at least one refinement of $i$ is non-conforming to any refinement of $s$, then $\textit{mioco$_\Phi$}(s)$ does not hold.
Compositionality, Decompositionality and Refinement in I/O Conformance Testing

Figure 2: Problem of modal-irioco regarding MIA with universal state.

Figure 3: Figures 3a and 3b are a counterexample to \( i_{\text{mioco}} s \Rightarrow i \sqsubseteq s \) because \( i_1 \sqsubseteq s_1 \) but not \( i_1 \sqsubseteq s_1 \). Figures 3c and 3d are a counterexample to \( i \sqsubseteq s \Rightarrow i_{\text{mioco}} s \) because \( i_2 \sqsubseteq s_2 \) but not \( i_2 \mioco s_2 \).

We first have to show that modal-irioco is preserved under modal refinement. Although, intuitions behind both relations are quite similar, they are incomparable. Figure 3 gives examples showing that both \( i_{\text{mioco}} s \Rightarrow i \sqsubseteq s \) and \( i \sqsubseteq s \Rightarrow i_{\text{mioco}} s \) do not hold. Firstly, we take a look at Figs. 3a and 3b. Here \( i_1 \mioco s_1 \) holds. Note, that \( \delta \in \text{Out}_\square (s_1 \text{ after } \square !o \cdot ?i) \) because the state on the right-hand side after the input \( i \) is may-quiescent. Though, \( i_1 \sqsubseteq s_1 \) does not hold \( i_1 \) has the output \( o' \) on the left-hand side, and it does not have \( o' \) on the right side. Therefore, \( i_{\text{mioco}} s \Rightarrow i \sqsubseteq s \). Secondly, consider Figs. 3c and 3d. In this case, \( i_2 \sqsubseteq s_2 \) holds because in \( s_2 \) the input \( i \) on the right-hand side is underspecified and may be implemented arbitrarily. Here, \( i_2 \mioco s_2 \) does not hold because \( \text{Out}_\square (i_2 \text{ after } \square !o \cdot ?i) \notin \text{Out}_\square (s_2 \text{ after } \square !o \cdot ?i) \). Therefore, \( i \sqsubseteq s \Rightarrow i_{\text{mioco}} s \) is also not true. To conclude MIA\(_\Phi\) refinement and mioco\(_\text{MIA}\) are incomparable. Instead, we obtain a weaker correspondence.

Theorem 1. Let \( i, s \) be MIA\(_\Phi\), \( i \) being weak may-input-enabled and \( i_{\text{mioco}} s \). Then for each \( i' \sqsubseteq s \) there exists \( s' \sqsubseteq s \) such that \( i' \mioco s' \) holds.

Note, that we refer to the \textit{ioco}-relation in our modal-irioco in Clause 1 in Def. 6. Hence, in order to relate modal-irioco and \textit{ioco}, we define applications of \textit{ioco} to IR-MIA by considering the may-transition relation as the actual transition relation.

\textbf{Definition 7 (ioco on MIA\(_\Phi\)).} Let \( i, s \) be MIA\(_\Phi\), \( i \) be weak may-input-enabled. Then, \( i_{\text{ioco}} s : \forall \sigma \in \text{Straces}_\square (s) : \text{Out}_\square (i \text{ after } \square \sigma) \subseteq \text{Out}_\square (s \text{ after } \square \sigma) \).

Based on this definition, we are able to prove correctness of modal-irioco.
Theorem 2 (modal-irioco is correct). Let \( i, s \) be MIA\(_\Phi\), \( i \) being weak may-input-enabled.

1. If \( i \text{mioco} \Phi s \), then for all \( i' \sqsubseteq \Phi i \), there exists \( s' \sqsubseteq \Phi s \) such that \( i' \text{ioco} s' \).

2. If there exists \( i' \sqsubseteq \Phi i \) such that \( i' \text{ioco} s' \) does not hold for any \( s' \sqsubseteq \Phi s \), then \( i \text{mioco} \Phi s \) does not hold.

Property 1 states soundness of modal-irioco. However, the immediate inverse does not hold as \( \text{ioco} \) does not guarantee mandatory behaviors of \( s \) to be actually implemented by \( i \) (cf. Fig. 4 for a counterexample where \( i \text{ioco} \Phi s \) but not \( i \text{mioco} \Phi s \)). Instead, Property 2 states completeness of modal-irioco in the sense that modal implementations \( i \) are rejected if at least one refinement of \( i \) exists not conforming to any refinement of specification \( s \). Finally, we conclude that \( \text{mioco} \Phi \) becomes a preorder if being restricted to input-enabled IR-MIA specifications.

Theorem 3. \( \text{mioco} \Phi \) is a preorder on the set of weak may-input-enabled MIA\(_\Phi\).

Must-input-enabledness (and therefore may-input-enabledness) of a specification \( s \) may be achieved for any given IR-MIA by applying a behavior-preserving canonical input completion, while still allowing arbitrary refinements of previously unspecified inputs (instead of ignoring inputs as, e.g., achieved by angelic completion [25]). This construction essentially adapts the notion of demonic completion [18] from IOLTS to IR-MIA as follows.

Definition 8 (Demonic Completion of IR-MIA). The demonic completion of MIA\(_\Phi\) \((Q, I, O, \rightarrow\Box, \rightarrow\Diamond, q_\Phi)\) with \( \forall q \in Q : q \xrightarrow{\tau} \Box q \Rightarrow q \xrightarrow{\tau} \Diamond \) is a MIA\(_\Phi\) \((Q', I, O, \rightarrow\Box', \rightarrow\Diamond', q_\Phi)\), where

- \( Q' = Q \cup \{q_X, q_\Omega\} \) with \( q_X, q_\Omega \notin Q \), and
- \( \rightarrow\Box' = \rightarrow\Box \cup \{(q, i, q_X) \mid q \in Q, i \in I, q \xrightarrow{i} \Box, q \xrightarrow{\tau} \Diamond\} \cup \{(q_X, \tau, q_\Omega)\} \cup \{(q_X, \lambda, q_X) \mid \lambda \in I\} \).
• \( \overset{\lambda}{\to}^c = \overset{\lambda}{\to} \cup \{(q,i,q_x) \mid q \in Q, i \in I, q \overset{i}{\to} \square, q \not\overset{\tau}{\to} \square\} \cup \{(q_x,\tau, q_\Omega)\} \cup \{(q_\Omega, \lambda, q_x) \mid \lambda \in (I \cup O)\} \cup \{(q_x, \lambda, q_x), (q_\Omega, \lambda, q_\Omega) \mid \lambda \in I\} \).

The restriction imposed by \( q \in Q : q \overset{\tau}{\to} \square \Rightarrow q \overset{\tau}{\to} \square \) is due to weak input-enabled states not being input-enabled anymore if an optional \( \tau \)-transition is removed. We refer to the demonic completion of MIA\( _\Phi \) s as \( \Xi(s) \).

Figure 5 illustrates demonic completion. As state \( q_1 \) of \( s \) is not must-input-enabled, a must-transition for action \( a \) is added from \( q_1 \) to \( q_2 \). The fresh states \( q_x \) and \( q_\Omega \) have outgoing must-transitions for each \( i \in I \), thus being (strong) must-input-enabled. Additionally, \( q_x \) in combination with \( q_\Omega \) allow (but do not require) every output \( o \in O \) (in \( q_x \) via one silent move), such that demonic completion preserves underspecification. We conclude that this construction preserves modal-irioco.

**Theorem 4.** Let \( i, s \) be MIA\( _\Phi \) with \( i \) being weak must-input-enabled. Then \( \text{mioco}_\Phi \Xi(s) \) if \( \text{mioco}_\Phi s \).

### 4 Compositionality

Interface theories are equipped with a (binary) interleaving parallel operator \( \| \) on interface specifications to define interaction behaviors in systems composed of multiple concurrently running components \[1]\). Intuitively, transition \( p \overset{\alpha}{\to} p', a \in O_P, \) of component \( P \) synchronizes with transition \( q \overset{\alpha}{\to} q', a \in I_Q, \) of component \( Q \), where the resulting synchronized action \( (p,q) \overset{\tau}{\to} (p',q') \) becomes a silent move. Modal interface theories generalize parallel composition to multicast communication (i.e., one output action synchronizes with all concurrently running components having this action as input) and explicit hiding of synchronized output actions \[22]\). According to MIA, we define parallel composition on IR-MIA in two steps: (1) standard parallel product \( P_1 \otimes P_2 \) on MIA\( P_1, P_2, \) followed by (2) parallel composition \( P_1 \parallel_P P_2, \) removing erroneous states \( (p_{11}, p_{21}) \) from \( P_1 \otimes P_2, \) where for an output action of \( p_{11}, \) no corresponding input is provided by \( p_{21} \) (and vice versa). In addition, all states \( (p_{11}', p_{21}') \) from which erroneous states are reachable are also removed (pruned) from \( P_1 \parallel_P P_2. \)

Concerning (1), we first require composability of \( P_1 \) and \( P_2 \) (i.e., disjoint output actions). In \( P_1 \otimes_P P_2, \) a fresh state \( p_{12\Phi} \) serves as unified failure state. The input alphabet of \( P_1 \otimes_P P_2 \) contains all those inputs of \( P_1 \) and \( P_2 \) not being contained in one of their output sets, whereas the output alphabet of \( P_1 \otimes_P P_2 \) is the union of both output sets. The modality \( \gamma \) of composed transitions \( (p_{11}, p_{21}) \overset{\alpha}{\to} (p_{11}', p_{21}') \) depends on the modality of the individual transitions.

**Definition 9** (IR-MIA Parallel Product). MIA\( _\Phi \) \( P_1, P_2 \) are composable if \( O_1 \cap O_2 = \emptyset \). The parallel product is defined as \( P_1 \otimes_P P_2 = ((P_1 \times P_2) \cup \{(q_{12}, \sigma, I, O, \overset{\gamma}{\to} \square, \overset{\gamma}{\to} () , p_{12\Phi}\}) , I, O, \overset{\gamma}{\to} \square, \overset{\gamma}{\to} () , p_{12\Phi}\}) \) \( (O_1 \cup O_2) \) and \( O = \text{def} O_1 \cup O_2 \), and where \( \overset{\gamma}{\to} \square \) and \( \overset{\gamma}{\to} () \) are the least relations satisfying the following conditions:

- **May1/Must1**: \( (p_{11}, p_{21}) \overset{\alpha}{\to} (p_{11}', p_{21}') \) if \( p_{11} \overset{\alpha}{\to} (p_{11}', p_{21}') \) and \( \alpha \notin A_2 \)
- **May2/Must2**: \( (p_{11}, p_{21}) \overset{\alpha}{\to} (p_{11}', p_{21}') \) if \( p_{21} \overset{\alpha}{\to} (p_{11}', p_{21}') \) and \( \alpha \notin A_1 \)
- **May3/Must3**: \( (p_{11}, p_{21}) \overset{\alpha}{\to} (p_{11}', p_{21}') \) if \( p_{11} \overset{\alpha}{\to} (p_{11}', p_{21}') \) and \( p_{21} \overset{\alpha}{\to} (p_{11}', p_{21}') \) for some \( a \)
- **May4/Must4**: \( (p_{11}, p_{21}) \overset{\alpha}{\to} (p_{11}', p_{21}') \) if \( p_{11} \overset{\alpha}{\to} (p_{11}', p_{21}') \) and \( p_{21} \overset{\alpha}{\to} (p_{11}', p_{21}') \) for some \( a \in I_1 \cup A_2 \)
- **May5/Must5**: \( (p_{11}, p_{21}) \overset{\alpha}{\to} (p_{11}', p_{21}') \) if \( p_{21} \overset{\alpha}{\to} (p_{11}', p_{21}') \) and \( p_{11} \overset{\alpha}{\to} (p_{11}', p_{21}') \) for some \( a \in I_2 \cap A_1 \)

Rules (May1/Must1) and (May2/Must2) define interleaving of transitions labeled with actions being exclusive to one of both components; whereas Rule (May3/Must3) synchronizes transitions with common actions, and the Rules (May4/Must4) and (May5/Must5) forbid transitions of a component labeled with inputs being common to both components, but not being supported by the other component. Concerning (2), we define \( E \subseteq P_1 \times P_2 \) to contain illegal state pairs \( (p_{11}, p_{21}) \) in \( P_1 \otimes_P P_2. \)
Definition 10 (Illegal State Pairs). Given a parallel product \( P_1 \otimes_\Phi P_2 \), a state \((p_1, p_2)\) is a new error if there exists \( a \in A_1 \cap A_2 \) such that
\[
\begin{align*}
&\bullet a \in O_1, \ p_1 \xrightarrow{a} q \quad \text{and} \quad p_2 \not\xrightarrow{a} q, \quad \text{or} \\
&\bullet a \in O_2, \ p_2 \xrightarrow{a} q \quad \text{and} \quad p_1 \not\xrightarrow{a} q, \quad \text{or} \\
&\bullet a \in O_1, \ p_1 \xrightarrow{a} q \quad \text{and} \quad p_2 \xrightarrow{a} q \quad \text{and} \quad p_2 \Phi, \quad \text{or} \\
&\bullet a \in O_2, \ p_2 \xrightarrow{a} q \quad \text{and} \quad p_1 \xrightarrow{a} q \quad \text{and} \quad p_1 \Phi.
\end{align*}
\]

The relation \( E \subseteq P_1 \times P_2 \) containing illegal state pairs is the least relation such that \((p_1, p_2) \in E\) if
\[
\begin{align*}
&\bullet (p_1, p_2) \text{ is a new error, or} \\
&\bullet (p_1, p_2) \xrightarrow{\omega} (p_1', p_2') \text{ with } \omega \in (O \cup \{\tau\}) \text{ and } (p_1', p_2') \in E.
\end{align*}
\]

If the initial state of \( P_1 \otimes_\Phi P_2 \) is illegal (i.e., \((p_{01}, p_{02}) \in E\)), it is replaced by a fresh initial state without incoming and outgoing transitions such that \( P_1 \) and \( P_2 \) are considered incompatible.

Definition 11 (IR-MIA Parallel Composition). The parallel composition \( P_1 \parallel_\Phi P_2 \) of \( P_1 \otimes_\Phi P_2 \) is obtained by pruning illegal states as follows.
\[
\begin{align*}
&\bullet \text{transitions leading to a state of the form } (q_1 \Phi, p_2) \text{ or } (p_1, q_2 \Phi) \text{ are redirected to } q_1 \Phi. \\
&\bullet \text{states } (p_1, p_2) \in E \text{ and all unreachable states (except for } q_1 \Phi) \text{ and all their incoming and outgoing transitions are removed.} \\
&\bullet \text{for states } (p_1, p_2) \notin E \text{ and } (p_1, p_2) \xrightarrow{i} (p_1', p_2') \in E, \ i \in I, \text{ all transitions } (p_1, p_2) \xrightarrow{i} (p_1', p_2') \text{ are removed.}
\end{align*}
\]

If \((p_1, p_2) \in P_1 \parallel_\Phi P_2\), we write \( p_1 \parallel_\Phi p_2 \) and call \( p_1 \) and \( p_2 \) compatible.

For example, consider \( P' = Q \parallel_\Phi D \) (cf. Fig. 1c and Fig. 4). Here, \( q_0 \) of both \( Q \) and \( D \) have action \( a \) as common action thus being synchronized to become an output action in \( P' \) (to allow multicast communication). Action \( a \) is mandatory in \( P' \) as \( a \) is mandatory in both \( Q \) and \( D \). In any other case, the resulting transition modality becomes optional. Further common actions (i.e., \( b \) and \( f \)) are treated similarly under composition. In contrast, transitions with actions being exclusive to \( Q \) or \( D \) are preserved under composition. As \( Q \otimes_\Phi D \) contains no illegal states, no pruning is required in \( P' = Q \parallel_\Phi D \). In contrast, assuming, e.g., one of the inputs \( a \) of \( Q \) being optional instead, then the initial state of \( P' \) would become illegal as \( a \in O_D, \ p_D \xrightarrow{a} q \quad \text{and} \quad p_D \not\xrightarrow{a} q, \quad \text{and} \quad Q \) and \( D \) would be incompatible.

We obtain the following compositionality result for modal-irioco with respect to parallel composition with multicast communication.

Theorem 5 (Compositionality of modal-irioco). Let \( s_1, s_2, i_1, \) and \( i_2 \) be MIA\(s_\Phi \) with \( i_1 \) and \( i_2 \) being strong must-input-enabled, and \( s_1 \) and \( s_2 \) being compatible. Then it holds that \((i_1 \text{ mioco}_s \land i_2 \text{ mioco}_s) \Rightarrow i_1 \parallel_\Phi i_2 \parallel_\Phi s_1 \parallel_\Phi s_2\).

Theorem 5 is restricted to must-input-enabled implementations as the input of an input/output pair has to be mandatory (otherwise leading to an illegal state). We further require strong input-enabledness as inputs in an input/output pair have to immediately react to outputs (otherwise, again, leading to an illegal state). Next, we show that IR-MIA parallel composition is associative, thus facilitating multicast communication among multiple IR-MIA components being composed in arbitrary order.

Lemma 2 (Associativity of IR-MIA Parallel Composition). Let \( P, Q, R \) be IR-MIA. It holds that \(( P \parallel_\Phi Q ) \parallel_\Phi R = P \parallel_\Phi ( Q \parallel_\Phi R ).\)
In addition, we show that compositionality of modal-irioco also holds if we combine multicast parallel composition with explicit hiding of outputs, if specification $s$ has no $\tau$-steps. For this, we first define parallel composition with hiding.

**Definition 12** (IR-MIA Parallel Product and Composition with Hiding). Two MIA$_\Phi$ $P_1$, $P_2$ are hiding composable (h-composable) if $O_1 \cap O_2 = \emptyset$ and $I_1 \cap I_2 = \emptyset$. For such MIA$_\Phi$, we define the parallel product $P_1 \otimes^\Phi P_2 = ((P_1 \times P_2) \cup \{q_{12}\}, I, O, \rightarrow_\emptyset, \rightarrow_0, \rightarrow_\top, q_{12})$, where $I = \text{def} (I_1 \cup I_2) \setminus (O_1 \cup O_2)$ and $O = \text{def} O_1 \cup O_2$, and where $\rightarrow_\emptyset$ and $\rightarrow_\top$ are the least relations satisfying the following conditions:

(May1/Must1) $p_1 (p_2) \xrightarrow{a} (p_1') (p_2')$ if $p_1 \xrightarrow{a} p_1'$ and $a \notin A_2$

(May2/Must2) $p_1 (p_2) \xrightarrow{a} (p_1') (p_2')$ if $p_2 \xrightarrow{a} p_2'$ and $a \notin A_1$

(May3/Must3) $p_1 (p_2) \xrightarrow{a} (p_1') (p_2')$ if $p_1 \xrightarrow{a} p_1'$ and $p_2 \xrightarrow{a} p_2'$ for some $a$.

From this parallel product with hiding, we obtain the parallel composition with hiding $P_1 \mid_\Phi P_2$ by the same pruning procedure as in Def. 11.

We obtain the following compositionality result for modal-irioco with respect to parallel composition with hiding.

**Theorem 6** (Compositionality of mioco$_{\text{MIA}}$ Regarding Parallel Composition with Hiding). Let $s_1$, $s_2$, $i_1$, and $i_2$ be strongly must-input-enabled MIA$_\Phi$. Then $(i_1 \mid i_2) \mid_{\text{mioco}_{\text{MIA}}} s_1 \mid \Phi \mid_{\text{mioco}_{\text{MIA}}} s_2 \iff i_1 \mid_\Phi i_2 \mid_{\text{mioco}_{\text{MIA}}} s_1 \mid \Phi s_2$ if $s_1$ and $s_2$ are compatible, $\forall q \in Q_{s_1} : \forall i \in I_{s_1} \cap O_{s_2} : q \xrightarrow{i_{s_1\Phi}} q$, and $\forall q \in Q_{s_2} : \forall i \in I_{s_2} \cap O_{s_1} : q \xrightarrow{i_{s_2\Phi}} q$.

5 Decompositionality

Compositionality of modal-irioco allows for decomposing I/O conformance testing of systems consisting of several interacting components. In particular, given two components $c_1$, $c_2$ being supposed to implement corresponding specifications $s_1$, $s_2$, then Theorem 5 ensures that if $c_1 \mid_{\text{mioco}_{\Phi} s_1}$ and $c_2 \mid_{\text{mioco}_{\Phi} s_2}$ holds, then $c_1 \parallel \Phi c_2 \mid_{\text{mioco}_{\Phi} s_1} \parallel \Phi s_2$ is guaranteed without the need for (re-)testing after composition. However, in order to benefit from this property, a mechanism is required to decompose specifications $s = s_1 \parallel \Phi s_2$ and respective implementations $i = c_1 \parallel \Phi c_2$, accordingly. Interface theories therefore provide quotient operators $\parallel$ serving as the inverse to parallel composition (i.e., if $c_1 \parallel c_2 = c$ then $c \parallel c_1 = c_2$), where $c_2$ is often referred to as unknown component [27] or testing context [20]. We therefore adopt the
quotient operator defined for MIA with universal state \( G \) to IR-MIA. Similar to parallel composition, the quotient operator is defined in two steps.

1. The pseudo-quotient \( P \land D \) is constructed as appropriate communication partner (if exists) for a given divisor \( D \) with respect to the overall specification \( P \).

2. The quotient \( P \parallel_0 D \) is derived from \( P \land D \), again, by pruning erroneous states.

For this, we require \( P \) and \( D \) to be \( \tau \)-free and \( D \) to be may-deterministic (i.e., \( d \xrightarrow{a} \tau d' \) and \( d \xrightarrow{a} \tau d'' \) implies \( d' = d'' \)). In contrast to \( G \), we restrict our considerations to IR-MIA with at least one state and one may-transition. A pair \( P \) and \( D \) satisfying these restrictions is called a quotient pair.

**Definition 13** (IR-MIA Pseudo-Quotient). Let \( \langle P, I_P, O_P, \xrightarrow{}_\square, \xrightarrow{}_\tau, p_\Phi \rangle \) and \( \langle D, I_D, O_D, \xrightarrow{}_\square, \xrightarrow{}_\tau, D \rangle \) be a MIA\(_\Phi \) quotient pair with \( A_D \subseteq A_P \) and \( O_D \subseteq O_P \). We set \( I = \text{def} I_P \cup O_D \) and \( O = \text{def} O_P \setminus O_D \). \( P \land D = \text{def} \langle P \times D, I, O, \xrightarrow{}_\square, \xrightarrow{}_\tau, \langle (p, d) \rangle \rangle \), where the transition relations are defined by the rules:

- (QMay1/QMust1) \( (p, d) \xrightarrow{a} (p', d) \) if \( p \xrightarrow{a} \gamma p' \neq p_\Phi \) and \( a \notin A_D \)
- (QMay2) \( (p, d) \xrightarrow{}_\tau (p', d') \) if \( p \xrightarrow{a} \tau p' \neq p_\Phi \) and \( d \xrightarrow{a} \square d' \neq d_\Phi \)
- (QMay3) \( (p, d) \xrightarrow{}_\tau (p', d') \) if \( p \xrightarrow{a} \tau p' \neq p_\Phi \) and \( d \xrightarrow{a} \tau d' \neq d_\Phi \) and \( a \notin O_P \cap I_D \)
- (QMust2) \( (p, d) \xrightarrow{a} \square (p', d') \) if \( p \xrightarrow{a} \square p' \neq p_\Phi \) and \( d \xrightarrow{a} \square d' \neq d_\Phi \)
- (QMust3) \( (p, d) \xrightarrow{}_\tau (p', d') \) if \( p \xrightarrow{a} \tau p' \neq p_\Phi \) and \( d \xrightarrow{a} \tau d' \neq d_\Phi \) and \( a \in O_D \)
- (QMay4/QMust4) \( (p, d) \xrightarrow{a} \gamma (p_\Phi, d_\Phi) \) if \( p \xrightarrow{a} \gamma p_\Phi \) and \( d \xrightarrow{a} \square d_\Phi \).

The Rules (QMay1/QMust1) to (QMust3) require \( p \neq p_\Phi \), as the special case \( p = p_\Phi \) is handled by rule (QMay4/QMust4). Rule (QMay1/QMust1) concerns transitions with uncommon actions. Rule (QMay2) requires a mandatory transition with action in \( D \) as composition requires inputs transitions labeled with common actions to be mandatory (the additional requirement of Rule (QMay3) is stated for the same reason). Rule (QMust3) only requires transitions to be optional, because if \( a \in O_D \) holds, then the resulting transition accepts as input a common action (which must be mandatory for the composition).

The quotient \( P \parallel_0 D \) is derived from pseudo-quotient \( P \land D \) by recursively pruning all so-called impossible states \( (p, d) \) (i.e., states leading to erroneous parallel composition).

**Definition 14** (IR-MIA Quotient). The set \( G \subseteq P \times D \) of impossible states of pseudo-quotient \( P \land D \) is defined as the least set satisfying the rules:

- (G1) \( p \xrightarrow{a} \tau p' \neq p_\Phi \) and \( d \xrightarrow{a} \tau \) and \( a \in A_D \) implies \( (p, d) \in G \)
- (G2) \( p \xrightarrow{a} \square p_\Phi \) and \( d \xrightarrow{a} \tau \) and \( a \in O_D \) implies \( (p, d) \in G \)
- (G3) \( (p, d) \xrightarrow{a} \tau r \) and \( r \in G \) implies \( (p, d) \in G \).

The quotient \( P \parallel_0 D \) is obtained from \( P \land D \) by deleting all states \( (p, d) \in G \) (and respective transitions). If \( (p, d) \in P \parallel_0 D \), then we write \( p \parallel_0 d \), and quotient \( P \parallel_0 D \) is defined.

Rule (G1) ensures that for a transition labeled with a common action, there is a corresponding transition in the divisor (otherwise, the state is impossible and therefore removed). Rule (G2) ensures that a forbidden action of the specification is also forbidden in the divisor (otherwise, the state is considered impossible). Finally, Rule (G3) (recursively) removes all states from which impossible states are reachable.

For example, consider the quotient \( Q = P \parallel_0 D \) (cf. Fig. 1c, Fig. 6a and Fig. 6c). A common action becomes input action in \( Q \) if it is an input action in both \( P \) and \( D \) (e.g., \( f \)), and likewise for output actions. If a common action is output action of \( P \) and input action of \( D \), then it becomes output of \( Q \) (e.g., \( b \)). In contrast, a common action must not be input action of \( P \) and output action of \( D \) as composing outputs with
Compositionality, Decompositionality and Refinement in I/O Conformance Testing

inputs always yields outputs. Actions being exclusive to \( P \) are treated similar to parallel composition, whereas \( D \) must not have exclusive actions (cf. Def. [13]).

For decomposability to hold for modal-irioco (i.e., \( i i / / \Phi c i i c o s i i / / \Phi c i s \)), we further require \( i \) to only have mandatory outputs as illustrated in Fig. [7] here, \( i m i o c o \Phi s \) does not hold, although \( c i m i o c o \Phi c s \) and \( i i / / \Phi c i i c o s i i / / \Phi c i s \) holds. This is due to the fact that optional outputs combined with mandatory outputs become mandatory inputs in the quotient (as parallel composition requires inputs of an input/output pair to be mandatory). The following result ensures that the quotient operator on IR-MIA indeed serves (under the aforementioned restrictions) as the inverse to parallel composition with respect to modal-irioco.

**Theorem 7** (Decompositionality of modal-irioco). Let \( i, s, c_i, \) and \( c_s \) be MIA-\( \Phi \) with \( i \) and \( c_i \) being weak must-input-enabled and all output behaviors of \( i \) being mandatory. Then \( i m i o c o \Phi s \) if \( i i / / \Phi c i i c o s i i / / \Phi c i s \) and \( c_i m i o c o \Phi c_s \).

Based on this result, modal-irioco supports synthesis of testing environments for testing through contexts [20, 10], as well as a solution to the unknown-component problem [27].

6 Related Work

We discuss related work on modal conformance relations, testing equivalences, alternative formulations of, and extensions to I/O conformance testing and composition/decomposition results in I/O conformance testing.

Various interfaces theories have been presented defining modal conformance relations by means of different kinds of modal refinement relations [22]. Amongst others, Bauer et al. use interface automata for compositional reasoning [3], whereas Alur et al. characterize modal conformance as alternating simulation relation on interface automata [2], and Larsen et al. have shown that both views on modal conformance coincide [13]. Based on our own previous work on modal I/O conformance testing [14, 15], we present, to the best of our knowledge, the first comprehensive testing theory by means of a modal I/O conformance relation. More recently, Bujtor et al. proposed testing relations on modal transition systems [8] based on (existing) test-suites, rather than being specification-based as our approach.

In contrast to I/O conformance relations, testing equivalences constitute a special class of (observational) equivalence relations [17, 23]. One major difference to ioco-like theories is that actions are usually undirected, thus no distinction between (input) refusals and (output) quiescence is made as in our approach [21, 6].

Concerning alterations of and extensions to I/O conformance testing, Veanes et al. and Gregorio-Rodríguez et al. propose to reformulate I/O conformance from suspension-trace inclusion to an alternating simulation to obtain a more fine-grained conformance notion constituting a preorder [26, 11]. However, these approaches neither distinguish optional from mandatory behaviors, nor underspecified from forbidden inputs as in our approach. Heerink and Tretmans extended ioco by introducing so-called
channels (i.e., subsets of I/O labels) for weakening the requirement of input-enabledness of implementations under test in order to also support refusal testing [12]. However, their notion of input refusals refers to a global property rather than being specific to particular states and they also do not distinguish mandatory from optional behaviors. Beohar and Mousavi extend ioco by replacing IOLTS with so-called Featured Transition Systems (FTS) and thereby enhance ioco to express fine-grained behavioral variability as apparent in software product lines [4]. As in our approach, FTS allow the environment to explicitly influence the presence or absence of particular transitions, whereas compositionality properties are not considered.

Concerning (de-)compositionality in I/O conformance testing, van der Bijl et al. present a compositional version of ioco with respect to synchronous parallel composition on IOTS [5], whereas Noroozi et al. consider asynchronously interacting components [19]. To overcome the inherent limitations of compositional I/O conformance testing, Daca et al. introduce alternative criteria for obtaining compositional specifications [10]. Concerning decomposition in I/O conformance testing, Noroozi et al. describe a framework for decomposition of ioco testing similar to our setting. However, all these related approaches neither distinguish mandatory from optional behaviors, nor support input refusals as in our approach.

7 Conclusion

We proposed a novel foundation for modal I/O-conformance testing theory based on a modified version of Modal Interface Automata with Input Refusals and show correctness and (de-)compositionality properties of the corresponding modal I/O conformance relation called modal-irioco. As a future work, we are interested in properties of modal-irioco regarding compositionality with respect to further operators on IR-MIA, such as interface conjunction [16] and asynchronous parallel composition [19]. Furthermore, we aim at generating test suites exploiting the capabilities of modal-irioco, i.e., test cases distinguishing optional from mandatory behaviors, as well as recognizing refused inputs.

References

[1] L. de Alfaro & T. A. Henzinger (2001): Interface Automata. In: ESEC, ACM, pp. 109–120.
[2] R. Alur, T. A. Henzinger, O. Kupferman & M. Y. Vardi (1998): Alternating Refinement Relations. In: CONCUR’98, LNCS 1466, Springer, pp. 163–178.
[3] S. S. Bauer, P. Mayer, A. Schroeder & R. Hennicker (2010): On Weak Modal Compatibility, Refinement, and the MIO Workbench. In: TACAS’10, LNCS 6015, Springer, pp. 175–189.
[4] H. Beohar & M. R. Mousavi (2014): Input-output Conformance Testing Based on Featured Transition Systems. SAC’14, ACM, NY, US, pp. 1272–1278.
[5] M. van der Bijl. A. Rensink & J. Tretmans (2004): Compositional Testing with ioco. In: FATES’04, LNCS 2931, Springer, pp. 86–100.
[6] I. B. Bourdonov, A. S. Kossatchev & V. V. Kuliamin (2006): Formal Conformance Testing of Systems with Refused Inputs and Forbidden Actions. MBT’06, pp. 83–96.
[7] F. Bujtor, S. Fendrich, G. Lüttgen & W. Vogler (2015): Nondeterministic Modal Interfaces. In: SOFSEM’15, LNCS 8939, Springer, pp. 152–163.
[8] F. Bujtor, L. Sorokin & W. Vogler (2015): Testing Preorders for dMTS: Deadlock- and the New Deadlock/Divergence-Testing. In: ACSD’15, pp. 60–69.
Compositionality, Decompositionality and Refinement in I/O Conformance Testing

[9] Ferenc Bujtor, Sascha Fendrich, Gerald Lüttgen & Walter Vogler (2014): Nondeterministic Modal Interfaces – Technical Report. Available at https://opus.bibliothek.uni-augsburg.de/opus4/frontdoor/index/index/year/2014/docId/2865

[10] P. Daca, T. A. Henzinger, W. Krenn & D. Niˇckovi´c (2014): Compositional Specifications for ioco Testing. In: ICST’14, pp. 373–382.

[11] C. Gregorio-Rodríguez, L. Llana & R. Martínez-Torres (2013): Input-Output Conformance Simulation (iocos) for Model Based Testing. In: FORTE’13, LNCS 7892, Springer, pp. 114–129.

[12] L. Heerink & J. Tretmans (1997): Refusal testing for classes of transition systems with inputs and outputs. In: FORTE’97, Springer, pp. 23–39.

[13] K. G. Larsen, U. Nyman & & W. Wąsowski (2007): Modal I/O Automata for Interface and Product Line Theories. In: ESOP’07, LNCS 4421, Springer, pp. 64–79.

[14] M. Lochau, S. Peldszus, M. Kowal & I. Schaefer (2014): Model-Based Testing. In: SFM’14, LNCS 8483, Springer, pp. 310–342.

[15] L. Luthmann, S. Mennicke & M. Lochau (2015): Towards an I/O Conformance Testing Theory of Software Product Lines based on Modal Interface Automata. In: FMSPLE’15, EPTCS, pp. 1–13.

[16] G. Lüttgen, W. Vogler & S. Fendrich (2014): Richer Interface Automata with Optimistic and Pessimistic Compatibility. Acta Inf., pp. 1–32.

[17] R. de Nicola (1987): Extensional Equivalences for Transition Systems. Acta Inf. 24(2), pp. 211–237.

[18] R. de Nicola & R. Segala (1995): A process algebraic view of input/output automata. Theoretical Computer Science 138(2), pp. 391–423.

[19] N. Noroozi, R. Khosravi, M. R. Mousavi & T. A. C. Willemse (2011): Synchronizing Asynchronous Conformance Testing. In: SEFM’11, Springer, pp. 334–349.

[20] N. Noroozi, M. R. Mousavi & T. A. C. Willemse (2013): Decomposability in Input Output Conformance Testing. In: MBT’13, pp. 51–66.

[21] I. Phillips (1987): Refusal Testing. Theoretical Computer Science 50(3), pp. 241–284.

[22] J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay & R. Passerone (2011): A Modal Interface Theory for Component-based Design. Fund. Informaticae 108, pp. 119–149.

[23] A. Rensink & W. Vogler (2007): Fair Testing. Inform. and Comp. 205(2), pp. 125–198.

[24] J. Tretmans (1996): Test Generation with Inputs, Outputs and Repetitive Quiescence.

[25] F. W. Vaandrager (1991): On the Relationship Between Process Algebra and Input/Output Automata. In: LICS’91, pp. 387–398.

[26] M. Veanes & N. Bjørner (2012): Alternating simulation and IOCO. STTT 14(4), pp. 387–405.

[27] Tiziano Villa, Nina Yevtushenko, Robert K Brayton, Alan Mishchenko, Alexandre Petrenko & Alberto Sangiovanni-Vincentelli (2011): The Unknown Component Problem: Theory and Applications. Springer Science & Business Media.
A Appendix

A.1 Proof of Lemma 1

Proof. We prove that MIAΦ refinement preserves strong may-input-enabledness. Assume two MIAΦ, P and Q, with P ⊑Φ Q and Q being strong may-input-enabled. There are two possible reasons why strong may-input-enabledness may be lost under refinement: (1) An input transition may be removed, and (2) the target of a transition may be changed to a new state not being may-input-enabled. However, under MIAΦ refinement, both cases are not possible.

1. According to property 5 of Def. 3 it must hold that q \xrightarrow{i} q' implies \exists p'. p \xrightarrow{i} q \xrightarrow{g} p' and (p', q') ∈ R. But, it is impossible to remove input transitions under MIAΦ refinement.

2. Now we have to look at the possibility of changing the target of an input transition to a new state not being strongly may-input-enabled. Properties 3 and 6 ensure that output transitions must be preserved if they are mandatory or may be removed if they are optional. However, MIAΦ refinement only allows to change the target to a new state, if the behavior of that new state is equivalent to the old (may-input-enabled) target state, thus also being strong may-input-enabled. Otherwise (p', q') ∈ R would be violated. The same holds for input transitions, but with one exception: input transitions may change their target to the failure state under MIAΦ refinement. By definition, the failure state does not have any (input) behavior, but this does not obstruct input-enabledness as the failure state is excluded from this requirement.

Under MIAΦ refinement, it is impossible to remove input transitions or change the target of a transition to a new state not being strong may-input-enabled. Therefore, strong may-input-enabledness is always preserved under IR-MIA refinement.

A.2 Proof of Theorem 1

Proof. We construct a unifying specification s_u serving as s' for all i' ⊑Φ i, by initially setting s_u = s. As i is may-input-enabled, inputs in i' are either may-failure, must-failure, or implemented as mandatory behavior. Hence, we do not have to modify s_u as optional inputs of i are either also optional in s_u or unspecified thus allowing input behaviors to be may-failure, must-failure, optional as well as mandatory. However, it is possible that δ ∈ Out_Φ(i after_Φ σ) although δ ∉ Out_Φ(s after_Φ σ) if there are states having only optional outputs. In this case, we add a τ-transition to every state in s_u, having only optional output behavior leading to a fresh must-quiescent state without any output transitions, such that δ ∈ Out_Φ(s_u after_Φ σ). From Def. 3 it follows that s_u ⊑Φ s. Thus, ∀i' ⊑Φ i : i' miocoΦ s_u holds and therefore the claim holds.

A.3 Proof of Theorem 2

Proof. We prove both parts separately.

- We make use of the unifying specification s_u from the proof of Theorem 1 (cf. Sect. A.2). Therefore, it holds that ∀i' ⊑Φ i : i' mioco s_u.

- For this part, we rely on the unifying specification s_u. Let i' ⊑Φ i be a MIAΦ such that i' mioco s_u does not hold, i.e., there exists a trace σ ∈ Straces_Φ(s_u) such that Out_Φ(i after_Φ σ) ⊈ Out_Φ(s_u after_Φ σ). Thus, we have Out_Φ(i after_Φ σ) \ Out_Φ(s_u after_Φ σ) \≠ \emptyset. From the construction of s_u, it follows that there is an ω ∈ Out_Φ(i after_Φ σ) \ Out_Φ(s_u after_Φ σ). But, then i miocoΦ s does not hold since Out_Φ(i after_Φ σ) \ ⊈ Out_Φ(s after_Φ σ).
A.4 Proof of Theorem 3

Proof. Let \( p, q, r \) be MIA\(_\Phi\) such that \( p \) and \( q \) are weak may-input-enabled and \( p \text{mioco}_\Phi q \) and \( q \text{mioco}_\Phi r \).

It holds by Def. 6 that \( p \text{mioco}_\Phi p \), i.e., \( \text{mioco}_\Phi \) is reflexive. It remains to be shown that \( p \text{mioco}_\Phi r \), i.e.,

(a) for all \( \sigma \in \text{Straces}_\Phi(r) \), \( \text{Out}_\Phi(p \text{after}_\Phi \sigma) \subseteq \text{Out}_\Phi(r \text{after}_\Phi \sigma) \) and (b) for all \( \text{Straces}_\Phi(p) \), \( \text{Out}_\Phi(r \text{after}_\Phi \sigma) \subseteq \text{Out}_\Phi(p \text{after}_\Phi \sigma) \).

Let \( \sigma \in \text{Straces}_\Phi(r) \). If \( \sigma \in \text{Straces}_\Phi(q) \), then (a) and (b) follow from transitivity of \( \subseteq \).

The case of \( \sigma \notin \text{Straces}_\Phi(q) \) remains.

Suppose (a) fails for a \( \sigma \in \text{Straces}_\Phi(r) \setminus \text{Straces}_\Phi(q) \), i.e., such a \( \sigma \) exists. Trace \( \sigma \) decomposes into \( \sigma_1 \cdot a \cdot \sigma_2 \) where \( \sigma_1 \in \text{Straces}_\Phi(q) \) but \( \sigma_1 \cdot a \notin \text{Straces}_\Phi(q) \). Since \( \text{Out}_\Phi(p \text{after}_\Phi \sigma_1) \subseteq \text{Out}_\Phi(q \text{after}_\Phi \sigma_1) \), \( a \notin O \cup \{\delta\} \). Otherwise, \( a \in I \) contradicts weak may-input-enabledness of \( q \). Thus, \( \sigma \in \text{Straces}_\Phi(q) \).

Case (b) remains for \( \sigma \in \text{Straces}_\Phi(p) \setminus \text{Straces}_\Phi(q) \). We show that such a \( \sigma \) again contradicts the assumptions of the theorem. As \( \sigma \notin \text{Straces}_\Phi(q) \), \( \sigma \) decomposes into a prefix \( \sigma_1 \in \text{Straces}_\Phi(q) \) and a postfix \( a \cdot \sigma_2 \) such that \( \sigma_1 \cdot a \notin \text{Straces}_\Phi(q) \). Since \( \sigma_1 \cdot a \in \text{Straces}_\Phi(p) \) and \( p \text{mioco}_\Phi q \), \( a \notin O \cup \{\delta\} \).

Hence \( a \in I \), but as stated above, this contradicts the assumption that \( q \) is weak input-enabled.

From reflexivity and transitivity of \( \text{mioco}_\Phi \) it follows that \( \text{mioco}_\Phi \) is indeed a preorder on weak may-input-enabled MIA\(_\Phi\).

A.5 Proof of Theorem 4

Proof. Let \( i, s \) be MIA\(_\Phi\) with \( i \) being weak must-input-enabled. We prove that it holds that (1) \( \forall \sigma \in \text{Straces}_\Phi(s) : \text{Out}_\Phi(i \text{after}_\Phi \sigma) \subseteq \text{Out}_\Phi(s \text{after}_\Phi \sigma) \Rightarrow \forall \sigma \in \text{Straces}_\Phi(\Xi(s)) : \text{Out}_\Phi(i \text{after}_\Phi \sigma) \subseteq \text{Out}_\Phi(\Xi(s) \text{after}_\Phi \sigma) \) and (2) \( \forall \sigma \in \text{Straces}_\Phi(i) : \text{Out}_\Phi(s \text{after}_\Phi \sigma) \subseteq \text{Out}_\Phi(i \text{after}_\Phi \sigma) \Rightarrow \forall \sigma \in \text{Straces}_\Phi(i) : \text{Out}_\Phi(\Xi(s) \text{after}_\Phi \sigma) \subseteq \text{Out}_\Phi(i \text{after}_\Phi \sigma) \).

1. Because of \( i \text{mioco}_\Phi s \), the subset relation holds for all \( \text{Straces}_\Phi \) specified by \( s \), i.e., \( \forall \sigma \in \text{Straces}_\Phi(s) : \text{Out}_\Phi(i \text{after}_\Phi \sigma) \subseteq \text{Out}_\Phi(\Xi(s) \text{after}_\Phi \sigma) \). Therefore, we have to prove the assumption for all \( \text{Straces}_\Phi(\Xi(s)) \setminus \text{Straces}_\Phi(s) \), i.e., all traces not specified by \( s \).

Let \( \sigma = \sigma' \cdot i \cdot \sigma'' \) with \( \sigma' \in \text{Straces}_\Phi(s) \), \( i \in I \) being an unspecified input such that \( \sigma \notin \text{Straces}_\Phi(s) \), and \( \sigma \in \text{Straces}_\Phi(\Xi(s)) \). We prove that for every \( \text{MIA}_{DC} \) it holds by definition that \( \{(q_\Omega, \lambda, q_\chi) | \lambda \in O\} \subseteq \text{Out} \) and \( \{(q_\chi, \tau, q_\Omega) \subseteq \text{Out} \) because \( \text{Out}_\Phi(i \text{after}_\Phi \sigma) \subseteq \text{Out} \) is always true.

2. Because of \( i \text{mioco}_\Phi s \), the subset relation holds for all \( \text{Straces}_\Phi \) specified by \( s \), i.e., \( \forall \sigma \in \text{Straces}_\Phi(i) : \text{Out}_\Phi(\Xi(s) \text{after}_\Phi \sigma) \subseteq \text{Out}_\Phi(i \text{after}_\Phi \sigma) \). Therefore, the assumption remains to be proven for all \( \sigma \in \text{Straces}_\Phi(\Xi(s)) \setminus \text{Straces}_\Phi(s) \), i.e., all traces not specified by \( s \). Let \( \sigma = \sigma' \cdot i \cdot \sigma'' \) with \( \sigma' \in \text{Straces}_\Phi(s) \), \( i \in I \) being an unspecified input such that \( \sigma \notin \text{Straces}_\Phi(s) \), and \( \sigma \in \text{Straces}_\Phi(\Xi(s)) \).

Therefore, both assumptions hold and \( i \text{mioco}_\Phi s \Rightarrow i \text{mioco}_\Phi \Xi(s) \).
A.6 Proof of Theorem [5]

Proof. Let \( s_1, s_2, i_1, \) and \( i_2 \) be MIA\( _\Phi \) with \( i_1 \) and \( i_2 \) being must-input-enabled, and \( s_1 \) and \( s_2 \) being compatible. Additionally, \( i_1 \ mioco_{\Phi, s_1} \) and \( i_2 \ mioco_{\Phi, s_2} \) hold. In order to prove \( i_1 \parallel i_2 \ mioco_{\Phi, s_1} \parallel \Phi \ s_2 \), we prove that (1) \( \forall \sigma \in Straces_\Phi(s_1 \parallel \Phi \ s_2) : Out_\Phi(i_1 \parallel \Phi \ i_2 \after_\Phi \sigma) \subseteq Out_\Phi(s_1 \parallel \Phi \ s_2 \after_\Phi \sigma) \) and (2) \( \forall \sigma \in Straces_\Phi(s_1 \parallel \Phi \ s_2) : Out_\Phi(s_1 \parallel \Phi \ s_2 \after_\Phi \sigma) \subseteq Out_\Phi(i_1 \parallel \Phi \ i_2 \after_\Phi \sigma) \). 

1. Let \( \omega \in Out_\Phi(i_1 \parallel \Phi \ i_2 \after_\Phi \sigma) \) such that \( \omega \in Out_\Phi(i_1 \after_\Phi \sigma) \). We prove that \( \omega \in Out_\Phi(s_1 \parallel \Phi \ s_2 \after_\Phi \sigma) \) holds. \( Out_\Phi(s_1 \parallel \Phi \ s_2 \after_\Phi \sigma) \neq \emptyset \) because \( \sigma \in Straces_\Phi(s_1 \parallel \Phi \ s_2) \). We now have to distinguish between \( \omega \in O \) and \( \omega = \delta \). If \( Out_\Phi(i_1 \parallel \Phi \ i_2 \after_\Phi \sigma) \subseteq O \), then \( \omega \in Out_\Phi(s_1 \parallel \Phi \ s_2 \after_\Phi \sigma) \) because otherwise \( i_1 \) would have more output behavior than \( s_1 \) such that \( i_1 \ mioco_{\Phi, s_1} \) would not hold. If \( Out_\Phi(i_1 \parallel \Phi \ i_2 \after_\Phi \sigma) = \{ \delta \} \), then \( \omega \notin Out_\Phi(s_1 \parallel \Phi \ s_2 \after_\Phi \sigma) \) because otherwise \( i_1 \) would have less mandatory behavior than \( s_1 \) such that \( i_1 \ mioco_{\Phi, s_1} \) would not hold.

Additionally, we have to consider pruning applied in \( s_1 \parallel \Phi \ s_2 \) but not in \( i_1 \parallel \Phi \ i_2 \). In order for pruning to occur in \( s_1 \parallel \Phi \ s_2 \) but not in \( i_1 \parallel \Phi \ i_2 \), there must be an optional input \( i \in A_1 \cap A_2 \) of \( s_1 \) becoming mandatory in \( i \). Then in \( s_1 \parallel \Phi \ s_2 \), pruning occurs as it is a new error (provided that \( s_2 \) performs a matching output). In \( i_1 \parallel \Phi \ i_2 \), there is no pruning as \( p_1 \uparrow \rightarrow \square \) does not hold. However, due to the definition of illegal states and pruning (cf. Def. [10] and Def. [11], all states being able to reach an illegal state through outputs and all their incoming and outgoing transitions are removed. Therefore, the least removed action of a trace is an input. Hence, that state in \( s_1 \) which had the removed input as an outgoing transition is underspecified and traces including that input are never checked (only traces \( \sigma \in Straces_\Phi(s_1 \parallel \Phi \ s_2) \) are checked). If the optional input \( i \) remains optional or is removed, then the same pruning is applied in \( i_1 \parallel \Phi \ i_2 \) because the state which should have the input is an illegal state. Therefore, \( \forall \sigma \in Straces_\Phi(s_1 \parallel \Phi \ s_2) : Out_\Phi(i_1 \parallel \Phi \ i_2 \after_\Phi \sigma) \subseteq Out_\Phi(s_1 \parallel \Phi \ s_2 \after_\Phi \sigma) \) is always true.

2. Let \( \omega \in Out_\Phi(i_1 \parallel \Phi \ i_2 \after_\Phi \sigma) \) and \( \omega \in Out_\Phi(s_1 \after_\Phi \sigma) \). \( \omega \in Out_\Phi(i_1 \parallel \Phi \ i_2 \after_\Phi \sigma) \) must hold because otherwise \( i_1 \) would have less mandatory output behavior than \( s_1 \) requires such that \( i_1 \ mioco_{\Phi, s_1} \) does not hold. Unlike the first part of this proof, the second part does not need to consider pruning because if there is pruning in \( s_1 \parallel \Phi \ s_2 \), then there is also pruning in \( i_1 \parallel \Phi \ i_2 \). This is because pruning is only performed if the input of a common action is optional, leading to the failure state, or the input is unspecified. If an input is optional in \( i_1 \parallel \Phi \ i_2 \), then it is optional or underspecified in \( s_1 \parallel \Phi \ s_2 \) (meaning, there is pruning on both sides). If an input is leading to the failure state in \( i_1 \parallel \Phi \ i_2 \), then it is optional or leading to failure state in \( s_1 \parallel \Phi \ s_2 \) (again, pruning on both sides). Underspecification is only possible for \( s_1 \) or \( s_2 \) so in this case pruning only takes place in \( s_1 \parallel \Phi \ s_2 \). Therefore, \( \forall \sigma \in Straces_\Phi(s_1 \parallel \Phi \ s_2) : Out_\Phi(i_1 \parallel \Phi \ i_2 \after_\Phi \sigma) \subseteq Out_\Phi(s_1 \parallel \Phi \ s_2 \after_\Phi \sigma) \) is always true.

A.7 Proof of Lemma [2]

To prove associativity of IR-MIA parallel composition, we first define a transformation of MIA according to Bujtor et al. [7], and we prove the transformation to be correct regarding parallel composition. Then, associativity of IR-MIA parallel composition directly follows, because the parallel composition of MIA according to Bujtor et al. is associative.
Definition 15 (Transformation of MIA<sub>A</sub> to MIA<sub>Φ</sub>). The transformation function $\mathcal{T} : \text{MIA}_A \rightarrow \text{MIA}_\Phi$ is defined as $\mathcal{T}(P) := (P', I_P, O_P, \rightarrow_P^A, \rightarrow_P^\Phi, p_\Phi)$ with:

$$P' = (P \setminus \{u_P\}) \cup \{p_\Phi\}$$

- $\rightarrow_P^\Phi = \{(p, i, p_\Phi) \mid p \in P, i \in I, p \not\rightarrow_p\}$
- $\rightarrow_P^A = \rightarrow_P^A \cup \rightarrow_P^\Phi$
- $\rightarrow_P^\Phi = (\rightarrow_P \cap (P' \times A_P^F \times P')) \cup \rightarrow_P^\Phi$

$\cup$ denotes the disjoint union, i.e., it holds that $\langle P \setminus \{u_P\} \rangle \cap \{p_\Phi\} = \emptyset$.

Theorem 8. Let $P$ and $Q$ be MIA<sub>A</sub>. Then it holds that $\mathcal{T}(P) \parallel_\Phi \mathcal{T}(Q) \equiv \mathcal{T}(P \parallel_A Q)$.

Proof. Let $P$ and $Q$ be MIA<sub>A</sub>, $A = \mathcal{T}(P) \parallel_\Phi \mathcal{T}(Q)$, and $B = \mathcal{T}(P \parallel_A Q)$. To prove $A \cong B$, we have to prove that $(S_A, I_A, O_A, \rightarrow_A^A, \rightarrow_A^{a_\Phi}) \cong (S_B, I_B, O_B, \rightarrow_B^B, \rightarrow_B^{b_\Phi})$.

- When constructing the parallel product, the sets of states of both automata is equal because the rules for building the parallel product are similar (cf. Def. 4 of Bujtor et al. [7] and Def. 9). The MIA<sub>Φ</sub> parallel product uses two additional rules to ensure inputs being implicitly forbidden for MIA<sub>A</sub> are explicitly forbidden in MIA<sub>Φ</sub>. Next, the sets of new errors of the parallel composition with multicast are equal because MIA<sub>Φ</sub> share the two rules of MIA<sub>A</sub> (cf. definitions for new errors in Def. 5 of Bujtor et al. [7] and Def. 11). Again, MIA<sub>Φ</sub> parallel composition needs two additional rules to ensure explicitly forbidden inputs (which are implicitly forbidden in MIA<sub>A</sub>) are taken into account. Additionally, MIA<sub>A</sub> parallel composition incorporates inherited errors not being defined for MIA<sub>Φ</sub> parallel composition because they are already explicitly forbidden through MIA<sub>Φ</sub> parallel product rules May4/Must4 and May5/Must5. As a consequence, the sets of illegal states are equal because the new errors are equivalent (and the inherited errors are taken into account for MIA<sub>Φ</sub>). In the last step, an equal set of illegal states is pruned in MIA<sub>A</sub> and MIA<sub>Φ</sub>. Furthermore, the transformations $\mathcal{T}(P)$ and $\mathcal{T}(Q)$ remove the universal states from $P$ and $Q$, respectively, and add the failure state. After using the MIA<sub>Φ</sub> parallel composition with multicast, exactly one failure state is left. When performing $P \parallel_A Q$, exactly one universal state remains. The transformation then removes the universal state and adds a failure state. Therefore, it holds that $S_A = S_B$.

- The transformation does not change the set of inputs and outputs. Therefore, $I_A = (I_P \cup I_Q) \setminus (O_P \cup O_Q) = I_B$ and $O_A = O_P \cup O_Q = O_B$.

- The transformation of $P \parallel_A Q$ adds must-transitions with the failure state as their target, i.e., it adds the set $\{(b, i, b_\Phi) \mid b \in S_B, i \in I, b \not\rightarrow_p\}$ in order to obtain $B$. All these transitions are also contained in $A$ because the transformation of $P$ and $Q$ adds equal transitions to the respective sets of must-transitions. $P \parallel_\Phi Q$ then combines all failure states into one failure state. Additionally, the sets of illegal states of the composition are equal (as described above). Therefore, an equal set of must-transitions is pruned, and it holds that $\rightarrow_A^{a_\Phi} = \rightarrow_B^{b_\Phi}$.

- The transformation of $P \parallel_A Q$ adds may-transitions with the failure state as their target, i.e., it adds the set $\{(b, i, b_\Phi) \mid b \in S_B, i \in I, b \not\rightarrow_p\}$. Furthermore, it removes all transitions with the universal state as their target, i.e., it removes the set $\rightarrow_B^{b_\Phi} \cap (S_B' \times A_B^F \times S_B')$ in order to obtain $B$ (with $S_B' = (S_B \setminus \{u_B\}) \cup \{b_\Phi\}$). All these transitions are also contained in $A$ because the transformation of $P$ and $Q$ adds equal transitions to their sets of may-transitions. $P \parallel_\Phi Q$ then combines all failure states into one failure state. Additionally, the sets of illegal states of the composition are
equal (as described above). Therefore, an equal set of may-transitions is pruned, and it holds that
\[ \gamma_0^A = \gamma_0^B. \]

- The transformation removes the universal state and adds a failure state to \( P \) and \( Q \), respectively. The MIA\(_\Phi\) parallel composition with multicast combines all failure states into one failure state. The MIA\(_A\) parallel composition with multicast combines all universal states into one universal state. The transformation then removes the universal state and adds a failure state. Hence, it holds that \( a_\Phi = b_\Phi \).

Therefore, it holds that \( (S_A, I_A, O_A, \gamma_0^A, a_\Phi) \cong (S_B, I_B, O_B, \gamma_0^B, b_\Phi) \), i.e., \( \mathcal{T}(P) \parallel_{\Phi} \mathcal{T}(Q) \cong \mathcal{T}(P \parallel_A Q) \).

Now, correctness of Lemma 2 directly follows.

**Proof.** Because of Theorem 8 and associativity of parallel composition according to Bujtor et al. [7], it follows that \( (P \parallel_{\Phi} Q) \parallel_{\Phi} R = P \parallel_{\Phi} (Q \parallel_{\Phi} R) \).

### A.8 Proof of Theorem 6

To prove Theorem 6, we first transfer the hiding operator of MIA according to Bujtor et al. [9] to IR-MIA, and prove some intermediate results. The definition of MIA hiding according to Bujtor et al. is directly transferable to MIA\(_\Phi\) because hiding only affects output behavior (which is not changed by the transformation from MIA according to Bujtor et al. to MIA\(_\Phi\)).

**Definition 16 (Hiding for MIA\(_\Phi\)).** Given a MIA\(_\Phi\) \( P = (P, I, O, \gamma, p_\Phi) \) and \( L \subseteq O \), then \( \text{Phiding} \, L \) is a MIA\(_\Phi\) \( P / L = \text{def} (P, I, O \setminus L, \gamma^{PL}, p_\Phi) \), where

\[
\gamma^{PL} = \{ (p_1, o, p_2) \mid p_1, p_2 \in P, o \in L, p_1 \xrightarrow{o} \gamma p_2 \}
\]

\[
\gamma^{PL} = \{ (p_1, \tau, p_2) \mid p_1, p_2 \in P, o \in L, p_1 \xrightarrow{o} \gamma p_2 \}
\]

\[
\gamma^{PL} = \{ \gamma^P \setminus \gamma^P / o \} \cup \gamma^\tau
\]

As described above, the hiding operation can easily be transferred to MIA\(_\Phi\) because it only affects output behavior. In fact, we define hiding in such a way that it commutes with the transformation described in Def. 15. This is due to the transformation (cf. Def. 15) not affecting any output behavior.

**Corollary 1.** Let \( P \) be a MIA\(_A\) with \( O_P \) being the set of output actions of \( P \), and a set of actions \( L \subseteq O_P \). Then \( \mathcal{T}(P \text{hiding} \, L) \cong \mathcal{T}(P) \text{hiding} \, L \).

Let \( \parallel_A \) and \( \mid_A \) denote composition with multicast and hiding according to Bujtor et al. [9], respectively. From Bujtor et al. [9], it follows that \( P \mid_A Q = (P \mid_A Q) / S \) with \( S = A_P \cap A_Q \) holds for MIA\(_A\). This means that building the parallel composition with hiding is equal to first building the parallel composition with multicast and, afterward, hide the common actions. We can show that this also holds for MIA\(_\Phi\). For practical purposes, this means that a tool being able to build the parallel composition with multicast only needs an extension which applies hiding (instead of creating an additional tool).

**Lemma 3.** Let \( P \) and \( Q \) be MIA\(_\Phi\) and \( S = A_P \cap A_Q \) the set of common actions of \( P \) and \( Q \). Then, \( P \mid_{\Phi} Q = (P \parallel_{\Phi} Q) / S \).

Next, we look at hiding in the context of mioco\(_\Phi\). In general, hiding does not preserve mioco\(_\Phi\), i.e., \( i \text{mioco}_{\Phi} \circ S \Rightarrow (i \text{hiding} \, L) \text{mioco}_{\Phi} \circ (i \text{hiding} \, L) \) does not hold. However, if we require the specification to be may-input-enabled, then hiding preserves mioco\(_\Phi\).
Theorem 9. Let $i,s$ be MIA$_{\Phi}$ with $i$ being weakly must-input-enabled and $s$ being weakly may-input-enabled, $O$ be the set of outputs, and $L \subseteq O$. Then $\text{imixo}_{{\Phi}_s} \Rightarrow (i\text{hiding}_L)\text{miixo}_{{\Phi}_s}(s\text{hiding}_L)$.

Proof. Let $i$ and $s$ be MIA$_{\Phi}$ with $\text{imixo}_{{\Phi}_s}$, $i$ being weakly must-input-enabled, and $s$ being weakly may-input-enabled, $O$ be the set of outputs of $i$ and $s$, and $L \subseteq O$. Additionally, $s' = s\text{hiding}_L$, $i' = i\text{hiding}_L$, $\sigma \in \text{Straces}_O(s)$, and $\sigma'$ be the corresponding trace $\sigma' \in \text{Straces}_O(s')$ where the hidden actions are removed from $\sigma$. To prove Theorem 9, we prove that (1) $\forall \sigma \in \text{Straces}_O(s): (\text{Out}_o(i\text{after}_o \sigma) \subseteq \text{Out}_o(s\text{after}_o \sigma)) \Rightarrow \forall \sigma' \in \text{Straces}_O(s'): (\text{Out}_o(i'\text{after}_o \sigma) \subseteq \text{Out}_o(s'\text{after}_o \sigma))$ and (2) $\forall \sigma \in \text{Straces}_O(s): (\text{Out}_o(s\text{after}_o \sigma) \subseteq \text{Out}_o(i\text{after}_o \sigma)) \Rightarrow \forall \sigma' \in \text{Straces}_O(s'): (\text{Out}_o(s'\text{after}_o \sigma) \subseteq \text{Out}_o(i\text{after}_o \sigma))$.

1. Assume, (1) does not hold. Then, there exists an $\omega \neq \tau$ such that $\omega \in \text{Out}_o(i\text{after}_o \sigma)$, $\omega \notin \text{Out}_o(s\text{after}_o \sigma)$, and $\omega \notin \text{Out}_o(s'\text{after}_o \sigma')$. However, with $s$ being may-input-enabled, we impose that $i$ implements every input $i \in I$ for every state $q \in Q$ such that $q_s \xrightarrow{i} q_s' \Rightarrow q_i \xrightarrow{i} q_i'$, $q_s \xrightarrow{i} q_s' \Rightarrow q_i \xrightarrow{i} q_i'$, or $q_s \xrightarrow{i} q_s' \Rightarrow q_i \xrightarrow{i} q_i'$. This means, we prescribe how $i$ should behave after $\sigma$. Therefore, $i'$ cannot have any additional output behavior after $\sigma'$ not being in $s'$ after $\sigma'$.

2. Assume, (2) does not hold. Then there exists an $\omega$ such that $\omega \in \text{Out}_o(s\text{after}_o \sigma)$, $\omega \notin \text{Out}_o(i\text{after}_o \sigma)$, $\omega \notin \text{Out}_o(s'\text{after}_o \sigma')$. However, this is not possible because in this case, there would be more output behavior in $i$ than in $s$ such that $\text{imixo}_{{\Phi}_s}$ would not hold.

Now, we prove that parallel composition with hiding preserves strong must-input-enabledness. For this, we first prove preservation for parallel composition with multicast.

Lemma 4. Let $P$ and $Q$ be strongly must-input-enabled MIA$_{\Phi}$. If it holds that $\forall p \in P: \forall i \in I_p \cap O_p: p \xrightarrow{i} p_\Phi$ and $\forall q \in Q: \forall i \in I_q \cap O_q: q \xrightarrow{i} q_\Phi$, then $P \parallel \Phi Q$ is must-input-enabled.

Proof. Let $I_p$ be the set of inputs of $P$, $I_q$ the set of inputs of $Q$, and $I = (I_p \cup I_q) \setminus (O_p \cup O_q)$. MIA$_{\Phi}$ parallel product rule (May1/Must1) ensures that all inputs $I_p \setminus (I_q \cup O_q)$ are accepted in every state of $P \parallel Q$. Rule (May2/Must2) ensures the same for all inputs $I_q \setminus (I_p \cup O_p)$, and rule (May3/Must3) for all inputs $I_p \cap I_q$. Additionally, no inputs are pruned because there are no new errors (and no illegal states) due to the fact that $P$ and $Q$ are strongly must-input-enabled, and there is no reachable input behavior of $I_p \cap O_q$ or $I_q \cap O_p$ with the failure state as its target.

Previously, it has been proven that MIA$_{\Phi}$ parallel composition with multicast preserves strong must-input-enabledness if there are no transitions having a common action with the failure state as their target (cf. Lemma 4). Because of Lemma 4 we can transfer that result to parallel composition with hiding. This is possible as hiding only affects output behavior, whereas input behavior remains unchanged (i.e., no input transition becomes internal behavior).

Corollary 2. Let $P, Q$ be strong must-input-enabled MIA$_{\Phi}$. If it holds that $\forall p \in P: \forall i \in I_p \cap O_p: p \xrightarrow{i} p_\Phi$ and $\forall q \in Q: \forall i \in I_q \cap O_q: q \xrightarrow{i} q_\Phi$, then $P \parallel \Phi Q$ is must-input-enabled.

Now, we can prove Theorem 9.

Proof. From Theorem 5 we know that if $i_1$ and $i_2$ are strongly must-input-enabled, and $s_1$ and $s_2$ are compatible, then $(i_1 \text{miixo}_{{\Phi}_s} s_1 \land i_2 \text{miixo}_{{\Phi}_s} s_2) \Rightarrow i_1 \parallel \text{miixo}_{{\Phi}_s} s_1 \parallel \text{miixo}_{{\Phi}_s} s_2$. Additionally, $P \parallel \Phi Q = (P \parallel \Phi Q)/S$ due to Lemma 3 (with $P$ and $Q$ being MIA$_{\Phi}$ and $S = A_p \cap A_q$). Furthermore, from Theorem 9...
it follows that $imioco_\Phi s \Rightarrow (i\text{ hiding } L)mioco_\Phi (s\text{ hiding } L)$ if $i$ is weakly must-input-enabled and $s$ is weakly may-input-enabled. However, MIA$_\Phi$ parallel composition with hiding does not preserve may-input-enabledness, and strong must-input-enabledness is only preserved if the automata to be composed do not contain any input transitions having a common action with the failure state as their target (cf. Corollary [2]). Therefore, we require $s_1$ and $s_2$ to be strongly must-input-enabled and not containing any input transitions having a common action with the failure state as their target in order for $s_1 \mid_\Phi s_2$ to be weakly may-input-enabled. It follows that $(i_1 mioco_\Phi s_1 \wedge i_2 mioco_\Phi s_2) \Rightarrow i_1 \mid_\Phi i_2 mioco_\Phi s_1 \mid_\Phi s_2$ if $i_1, i_2, s_1$, and $s_2$ are strongly must-input-enabled, and $s_1$ and $s_2$ do not contain any input transitions having a common action with the failure state as their target.

\section*{A.9 Proof of Theorem\textsuperscript{7}}

\textbf{Proof.}\ For this theorem to hold, we first have to consider correctness of the pseudo-quotient and quotient in IR-MIA since the original operators are defined on the MIA model proposed by Bujtor et al.\textsuperscript{7}. In Def. [15], a transformation function $\mathcal{T}$ from MIA$_A$ to MIA$_\Phi$ is given, altering the semantics of the universal state (in MIA$_A$) to become a failure state and adjusting transitions to these states, accordingly. To show correctness of Def. [13] and Def. [14] we prove for a quotient pair $P$ and $D$ that $\mathcal{T}(P)\parallel_\Phi \mathcal{T}(D) \cong \mathcal{T}(P \parallel A D)$ (i.e., isomorphism). The isomorphism used is simply the identity function.

Let $P$ and $D$ be MIA$_A$ (i.e., MIA according to [7]) such that $P$ and $D$ forms a quotient pair. The proof proceeds in two steps, (1) we show that the required property already holds for the pseudo-quotient, i.e., $\mathcal{T}(P)\parallel_\Phi \mathcal{T}(D) \cong \mathcal{T}(P \parallel A D)$ and (2) we show that the same set of states is pruned in order to obtain the quotients.

For (1), we consider states $(p, d)$ of the pseudo quotient. Please note that on both sides, the state identities are preserved. Hence, it suffices to show that $(p, d) \xrightarrow{a} (p', d')$ is covered by both pseudo-quotient operations. For rules (QMay1) to (QMay3) and (QMust1) to (QMust3), this obviously holds. The only difference is the handling of the universal/failure state by the remaining rules, (QMay4) and (QMay5) in [7] and (QMay4) in Def. [13].

If $(p, d) \xrightarrow{a} _\parallel P \parallel A D (e_p, e_D)$ (i.e., $(p, d) \xrightarrow{a} _\parallel p \parallel D (p', d')$) due to (QMay4), $p \xrightarrow{a} P e_p$ and by $\mathcal{T}$, $p \xrightarrow{a} \mathcal{T}(P)$. Thus, $(p, d) \xrightarrow{a} _\parallel \mathcal{T}(P) \parallel \mathcal{T}(D)$. If $(p, d) \xrightarrow{a} _\parallel (P \parallel A D) (e_p, e_D)$ due to (QMay5), $p \neq e_p, d \xrightarrow{a} D$ and $a \in A_D \setminus (O_P \cap I_D)$. Again, neither of the rules of Def. [15] applies. Thus, $(p, d) \xrightarrow{a} _\parallel \mathcal{T}(P) \parallel \mathcal{T}(D)$. Suppose $(p, d) \xrightarrow{a} _\parallel \mathcal{T}(P) \parallel \mathcal{T}(D) (p_q, d_q)$ due to rule (QMay4) of Def. [13]. In this case, $p \xrightarrow{a} _\parallel \mathcal{T}(P)$ $p_{\Phi}$, i.e., $p \xrightarrow{a} _\parallel \mathcal{T}(P)$. Since none of the rules (QMay4) or (QMay5) in [7] applies, $(p, d) \xrightarrow{a} _\parallel P \parallel A D$ thus $(p, d) \xrightarrow{a} _\parallel \mathcal{T}(P) \parallel \mathcal{T}(D)$.

For step (2), we have to consider the rules to identify impossible states. The rules of Def. [14] are adopted from Bujtor et al. [7]. The one rule missing in our set is due to the fact that we do not have a universal state in MIA$_\Phi$.

Therefore, $\mathcal{T}(P)\parallel_\Phi \mathcal{T}(D) \cong \mathcal{T}(P \parallel A D)$. We now proceed the proof of Theorem\textsuperscript{7}.

Let $i' = (i \parallel_\Phi c_1) \parallel_\Phi c_1$ and $s' = (s \parallel_\Phi c_2) \parallel_\Phi c_2$. From [7] and the first part of this proof, we conclude that $i' \subseteq i$ and $s' \subseteq s$. Therefore, we have to show that $i' mioco_\Phi s' \Rightarrow mioco_\Phi s$. MIA$_\Phi$ $i$ is weak must-input-enabled, and, therefore, $i'$ is also weak must-input-enabled, i.e., $i$ and $i'$ have the same input behavior. MIA$_\Phi$ $s$ may only have less than, or equal input behaviors as $s'$ as, under refinement, it is only possible to add inputs but not to remove inputs. Therefore, we only have to consider output behaviors. Hence, $i$ and $i'$ only differ in output behaviors, i.e., mandatory outputs and forbidden outputs of $i'$ may be optional in $i$. However, both cases are not possible as we restrict $i$ to only have mandatory outputs. \hfill \Box