Logical Methods in Computer Science
Vol. 9(3:4)2013, pp. 1–28
www.lmcs-online.org

Submitted Nov. 5, 2012
Published Aug. 14, 2013

MODAL INTERFACE AUTOMATA *

GERALD LÜTTGEN [a] AND WALTER VOGLER [b]

[a] Software Technologies Research Group, University of Bamberg, 96045 Bamberg, Germany
e-mail address: gerald.luettgen@swt-bamberg.de

[b] Institute for Computer Science, University of Augsburg, 86135 Augsburg, Germany
e-mail address: vogler@informatik.uni-augsburg.de

Abstract. De Alfaro and Henzinger's Interface Automata (IA) and Nyman et al.'s recent combination IOMTS of IA and Larsen's Modal Transition Systems (MTS) are established frameworks for specifying interfaces of system components. However, neither IA nor IOMTS consider conjunction that is needed in practice when a component shall satisfy multiple interfaces, while Larsen's MTS-conjunction is not closed and Beneš et al.'s conjunction on disjunctive MTS does not treat internal transitions. In addition, IOMTS-parallel composition exhibits a compositionality defect.

This article defines conjunction (and also disjunction) on IA and disjunctive MTS and proves the operators to be 'correct', i.e., the greatest lower bounds (least upper bounds) wrt. IA- and resp. MTS-refinement. As its main contribution, a novel interface theory called Modal Interface Automata (MIA) is introduced: MIA is a rich subset of IOMTS featuring explicit output-must-transitions while input-transitions are always allowed implicitly, is equipped with compositional parallel, conjunction and disjunction operators, and allows a simpler embedding of IA than Nyman's. Thus, it fixes the shortcomings of related work, without restricting designers to deterministic interfaces as Raclet et al.'s modal interface theory does.

1. Introduction

Interfaces play an important role when designing complex software and hardware systems so as to be able to check interoperability of system components already at design stage. Early interface theories deal with types of data and operations only and have been successfully deployed in compilers. Over the past two decades, research has focused on more advanced interface theories for *sequential* and object-oriented software systems, where interfaces also comprise behavioural types. Such types are often referred to as *contracts* [Mey92] and can express pre- and post-conditions and invariants of methods and classes. Much progress has

2012 ACM CCS: [**Theory of computation**]: Models of computation—Concurrency; Logic—Logic and Verification; Semantics and reasoning; [**Software and its engineering**]: Context specific languages—Interface definition languages; Software system models—State systems.

Key words and phrases: interface theories, interface automata, modal transition systems, disjunctive modal transition systems, modal interface automata, conjunction, disjunction.

* An extended abstract of this article appeared in 7th IFIP Intl. Conf. on *Theoretical Computer Science* (TCS 2012), vol. 7604 of Lecture Notes in Computer Science, pp. 265–279, Springer, 2012.

LOGICAL METHODS IN COMPUTER SCIENCE
DOI:10.2168/LMCS-9(3:4)2013

been made on the design of contract languages and on automated verification techniques that can decide whether a system component meets its contract (cf. [HLL$^+$12] for a survey).

More recently, *behavioural interfaces* have also been proposed and are being investigated for the use in *concurrent* systems, with prominent application examples being embedded systems (e.g., [MG05]) and web services (e.g., [BCHS07, MB03]). In this context, behavioural interfaces are intended to capture protocol aspects of component interaction. One prominent example of such an interface theory is de Alfaro and Henzinger's *Interface Automata* (IA) [dH01, dH05], which is based on labelled transition systems (LTS) but distinguishes a component's input and output actions. The theory comes with an asymmetric parallel composition operator, where a component may wait on inputs but never on outputs. Thus, a component's output must be consumed immediately, or an error occurs. In case no potential system environment may restrict the system components' behaviour so that all errors are avoided, the components are deemed to be incompatible.

Semantically, IA employs a refinement notion based on an alternating simulation, such that a component satisfies an interface if (a) it implements all input behaviour prescribed by the interface and (b) the interface permits all output behaviour executed by the implementing component. Accordingly and surprisingly, an output in a specification can always be ignored in an implementation. In particular, a component that consumes all inputs but never produces any output satisfies any interface. Since a specifier certainly wants to be able to prescribe at least some outputs, Larsen, Nyman and Wasowski have built their interface theory on Modal Transition Systems (MTS) [Lar90] rather than LTS, which enables one to distinguish between may- and must-transitions and thus to express mandatory outputs. The resulting *IOMTS* interface theory [LNW07], into which IA can be embedded, is equipped with an IA-style parallel composition and an MTS-style modal refinement. Unfortunately, IOMTS-modal refinement is not a precongruence (i.e., not compositional) for parallel composition; a related result in [LNW07] has already been shown incorrect by Raclet et al. in [RBB$^+$11].

The present article starts from the observation that the above interface theories are missing one important operator, namely conjunction on interfaces. Conjunction is needed in practice since components are often designed to satisfy multiple interfaces simultaneously, each of which specifies a particular aspect of component interaction. Indeed, conjunction is a key operator when specifying and developing systems from different viewpoints as is common in modern software engineering. We thus start off by recalling the IA-setting and defining a conjunction operator $\wedge$ for IA; we prove that $\wedge$ is indeed conjunction, i.e., the greatest lower bound wrt. alternating simulation (cf. Sec. 2). Essentially the same operator has recently and independently been defined in [CCJK12], where it is shown that it gives the greatest lower bound wrt. a *trace-based* refinement relation. As an aside, we also develop and investigate the dual disjunction operator $\vee$ for IA. This is a natural operator for describing alternatives in loose specifications, thus leaving implementation decisions to implementors.

Similarly, we define conjunction and disjunction operators for a slight extension of MTS (a subset of *Disjunctive MTS* [LX90], cf. Sec. 3), which paves us the way for our main contribution outlined below. Although Larsen has already studied conjunction and disjunction for MTS, his operators do, in contrast to ours, not preserve the MTS-property of syntactic consistency, i.e., a conjunction or disjunction almost always has some required transitions (must-transitions) that are not allowed (missing may-transitions). An additional difficulty when compared to the IA-setting is that two MTS-interfaces may not have a common implementation; indeed, inconsistencies may arise when composing MTSs conjunctively. We

handle inconsistencies in a two-stage definition of conjunction, adapting ideas from our prior work on conjunction in a CSP-style process algebra [LV10] that uses, however, a very different parallel operator and refinement preorder. In [BCK11], a conjunction for Disjunctive MTS (DMTS) is introduced in a two-stage style, too. Our construction and results for conjunction significantly extend the ones of [BCK11] in that we also treat internal transitions that, e.g., result from communication.

Note also that our setting employs event-based communication via handshake and thus differs substantially from the one of shared-memory communication studied by Abadi and Lamport in their paper on conjoining specifications [AL95]. The same comment applies to Doyen et al. [DHJP08], who have studied a conjunction operator for an interface theory involving shared-variable communication.

Our article's main contribution is a novel interface theory, called *Modal Interface Automata* (MIA), which is essentially a rich subset of IOMTS that still allows one to express output-must-transitions. In contrast to IOMTS, must-transitions can also be disjunctive, and input-transitions are either required (i.e., must-transitions) or allowed implicitly. MIA is equipped with an MTS-style conjunction $\wedge$, disjunction $\vee$ and an IOMTS-style parallel composition operator, as well as with a slight adaptation of IOMTS-refinement. We show that (i) MIA-refinement is a precongruence for all three operators; (ii) $\wedge$ ($\vee$) is indeed conjunction (disjunction) for this preorder; and (iii) IA can be embedded into MIA in a much cleaner, homomorphic fashion than into IOMTS [LNW07] (cf. Sec. 4). Thereby, we remedy the shortcomings of related work while, unlike the language-based modal interface theory of [RBB+11], still permitting nondeterminism in specifications.

2. Conjunction and Disjunction for Interface Automata

Interface Automata (IA) were introduced by de Alfaro and Henzinger [dH01, dH05] as a *reactive type* theory that abstractly describes the communication behaviour of software or hardware components in terms of their inputs and outputs. IAs are labelled transition systems where visible actions are partitioned into inputs and outputs. The idea is that interfaces interact with their environment according to the following rules. An interface cannot block an incoming input in any state but, if an input arrives unexpectedly, it is treated as a catastrophic system failure. This means that, if a state does not enable an input, this is a requirement on the environment not to produce this input. Vice versa, an interface guarantees not to produce any unspecified outputs, which are in turn inputs to the environment.

This intuition is reflected in the specific refinement relation of *alternating simulation* between IA and in the *parallel composition* on IA, which have been defined in [dH05] and are recalled in this section. Most importantly, however, we introduce and study a *conjunction operator* on IA, which is needed in practice to reason about components that are expected to satisfy multiple interfaces.

Definition 2.1 (Interface Automata [dH05])**.** An *Interface Automaton* (IA) is a tuple $Q = (Q, I, O, \longrightarrow)$, where

(1) Q is a set of states,
(2) I and O are disjoint input and output alphabets, resp., not containing the special, silent action τ,
(3) $\longrightarrow \subseteq Q \times (I \cup O \cup \{\tau\}) \times Q$ is the *transition relation.*

The transition relation is required to be *input-deterministic*, i.e., $a \in I$, $q \xrightarrow{a} q'$ and $q \xrightarrow{a} q''$ implies $q' = q''$. In the remainder, we write $q \xrightarrow{a}$ if $q \xrightarrow{a} q'$ for some q', as well as $q \not\xrightarrow{a}$ for its negation.

In contrast to [dH05] we do not distinguish internal actions and denote them all by τ, as is often done in process algebras. We let A stand for $I \cup O$, let a (α) range over A ($A \cup \{\tau\}$), and introduce the following weak transition relations: $q \stackrel{\varepsilon}{\Longrightarrow} q'$ if $q(\xrightarrow{\tau})^* q'$, and $q \stackrel{o}{\Longrightarrow} q'$ for $o \in O$ if $\exists q''.\, q \stackrel{\varepsilon}{\Longrightarrow} q'' \xrightarrow{o} q'$; note that there are no τ-transitions after the o-transition. Moreover, we define $\hat{\alpha} = \varepsilon$ if $\alpha = \tau$, and $\hat{\alpha} = \alpha$ otherwise.

Definition 2.2 (Alternating Simulation [dH05])**.** Let P and Q be IAs with common input and output alphabets. Relation $\mathcal{R} \subseteq P \times Q$ is an *alternating simulation relation* if for all $(p, q) \in \mathcal{R}$:

(i): $q \xrightarrow{a} q'$ and $a \in I$ implies $\exists p'.\, p \xrightarrow{a} p'$ and $(p', q') \in \mathcal{R}$,

(ii): $p \xrightarrow{\alpha} p'$ and $\alpha \in O \cup \{\tau\}$ implies $\exists q'.\, q \stackrel{\hat{\alpha}}{\Longrightarrow} q'$ and $(p', q') \in \mathcal{R}$.

We write $p \sqsubseteq_{\mathrm{IA}} q$ and say that p *IA-refines* q if there exists an alternating simulation relation $\mathcal{R}$ such that $(p, q) \in \mathcal{R}$.

According to the basic idea of IA, if specification Q in state q allows some input a delivered by the environment, then the related implementation state p of P must allow this input immediately in order to avoid system failure. Conversely, if P in state p produces output a to be consumed by the environment, this output must be expected by the environment even if $q \stackrel{a}{\Longrightarrow}$; this is because Q could have moved unobservedly from state q to some q' that enables a. Since inputs are not treated in Def. 2.2 (ii), they are always allowed for p.

It is easy to see that IA-refinement $\sqsubseteq_{\mathrm{IA}}$ is a preorder on IA and the largest alternating simulation relation. Given input and output alphabets I and O, resp., the IA

$$\mathit{BlackHole}_{I,O} =_{\mathrm{df}} (\{\mathit{blackhole}\}, I, O, \{(\mathit{blackhole}, a, \mathit{blackhole}) \mid a \in I\})$$

IA-refines any other IA over I and O.

2.1. **Conjunction on IA.** Two IAs with common alphabets are always logically consistent in the sense that they have a common implementation, e.g., the respective blackhole IA as noted above. This makes the definition of conjunction on IA relatively straightforward. Here and similarly later, we index a transition by the system's name to make clear from where it originates, in case this is not obvious from the context.

Definition 2.3 (Conjunction on IA)**.** Let $P = (P, I, O, \longrightarrow_P)$ and $Q = (Q, I, O, \longrightarrow_Q)$ be IAs with common input and output alphabets and disjoint state sets P and Q. The conjunction $P \wedge Q$ is defined by $(\{p \wedge q \mid p \in P, q \in Q\} \cup P \cup Q, I, O, \longrightarrow)$, where $\longrightarrow$ is the least set satisfying $\longrightarrow_P \subseteq \longrightarrow$, $\longrightarrow_Q \subseteq \longrightarrow$, and the following operational rules:

(I1)	$p \wedge q \xrightarrow{a} p'$	if	$p \xrightarrow{a}_P p'$, $q \not\xrightarrow{a}_Q$ and $a \in I$
(I2)	$p \wedge q \xrightarrow{a} q'$	if	$p \not\xrightarrow{a}_P$, $q \xrightarrow{a}_Q q'$ and $a \in I$
(I3)	$p \wedge q \xrightarrow{a} p' \wedge q'$	if	$p \xrightarrow{a}_P p'$, $q \xrightarrow{a}_Q q'$ and $a \in I$
(O)	$p \wedge q \xrightarrow{a} p' \wedge q'$	if	$p \xrightarrow{a}_P p'$, $q \xrightarrow{a}_Q q'$ and $a \in O$
(T1)	$p \wedge q \xrightarrow{\tau} p' \wedge q$	if	$p \xrightarrow{\tau}_P p'$
(T2)	$p \wedge q \xrightarrow{\tau} p \wedge q'$	if	$q \xrightarrow{\tau}_Q q'$

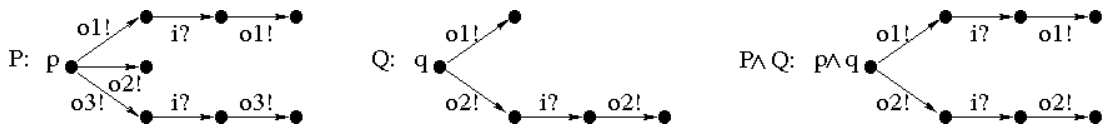

Figure 1: Example illustrating IA-conjunction.

Intuitively, conjunction is the synchronous product over actions (cf. Rules (I3), (O), (T1) and (T2)). Since inputs are always implicitly present, this also explains Rules (I1) and (I2); for example, in Rule (I1), q does not impose any restrictions on the behaviour after input a and is therefore dropped from the target state. Moreover, the conjunction operator is commutative and associative. As an aside, note that the rules with digit 2 in their names are the symmetric cases of the respective rules with digit 1; this convention will hold true throughout this article. Fig. 1 applies the rules above to an illustrating example; here and in the following figures, we write $a?$ for an input a and $a!$ for an output a.

Essentially the same conjunction operator is defined by Chen et al. in [CCJK12], where a non-standard variant of IA is studied that employs *explicit* error states and uses a trace-based semantics and refinement preorder (going back to Dill [Dil89]). The difference between their conjunction and Def. 2.3 is that error states are explicitly used in the clauses that correspond to Rules (I1) and (I2) above, which renders our definition arguably more elegant. In [CCJK12], an analogue theorem to Thm. 2.4 below is shown, but its statement is different as it refers to a different refinement preorder. Also note that, deviating from the IA-literature, error states are called inconsistent in [CCJK12], but this is not related to logic inconsistency as studied by us.

Our first result states that an implementation satisfies the conjunction of interfaces exactly if it satisfies each of them. This is a desired property in system design where each interface describes one aspect (or view) of the overall specification.

Theorem 2.4 ($\wedge$ is And)**.** *Let P, Q, R be IAs with states p, q, r, resp. Then, $r \sqsubseteq_{IA} p$ and $r \sqsubseteq_{IA} q$ if and only if $r \sqsubseteq_{IA} p \wedge q$.*

Proof. "$\Longleftarrow$": It is sufficient to show that $\mathcal{R} =_{\text{df}} \{(r,p) \mid \exists q.\, r \sqsubseteq_{\text{IA}} p \wedge q\} \cup \sqsubseteq_{\text{IA}}$ is an alternating simulation relation. Let $(r,p) \in \mathcal{R}$ due to q; the case $r \sqsubseteq_{\text{IA}} p$ is obvious. We check the conditions of Def. 2.2:

- Let $p \xrightarrow{a}_P p'$ with $a \in I$.
  - $q \not\xrightarrow{a}_Q$*:* Hence, $p \wedge q \xrightarrow{a} p'$ by Rule (I1) and, due to $r \sqsubseteq_{\text{IA}} p \wedge q$, there exists some r' with $r \xrightarrow{a}_R r'$ and $r' \sqsubseteq_{\text{IA}} p'$. Since $(r', p') \in \mathcal{R}$ we are done.
  - $q \xrightarrow{a}_Q q'$*:* Hence, $p \wedge q \xrightarrow{a} p' \wedge q'$ by Rule (I3) and, due to $r \sqsubseteq_{\text{IA}} p \wedge q$, there exists some r' with $r \xrightarrow{a}_R r'$ and $r' \sqsubseteq_{\text{IA}} p' \wedge q'$. Now, $(p', q') \in \mathcal{R}$.
- Let $r \xrightarrow{\alpha}_R r'$ with $\alpha \in O \cup \{\tau\}$.
  - $\alpha \neq \tau$*:* Thus, by Rule (O) and possibly Rules (T1), (T2), $p \wedge q \overset{\alpha}{\Longrightarrow} p' \wedge q'$ with $r' \sqsubseteq_{\text{IA}} p' \wedge q'$. We can project the transition sequence underlying $p \wedge q \overset{\alpha}{\Longrightarrow} p' \wedge q'$ to the P-component and get $p \overset{\alpha}{\Longrightarrow}_P p'$, and we are done since $(r', p') \in \mathcal{R}$.
  - $\alpha = \tau$*:* Hence, $p \wedge q \overset{\varepsilon}{\Longrightarrow} p' \wedge q'$, possibly by Rules (T1) and (T2), with $r' \sqsubseteq_{\text{IA}} p' \wedge q'$. Again, we can project to $p \overset{\varepsilon}{\Longrightarrow}_P p'$ (where possibly $p' = p$) and also have $(r', p') \in \mathcal{R}$.

"$\Longrightarrow$": We show that $\mathcal{R} =_{\text{df}} \{(r, p \wedge q) \mid r \sqsubseteq_{\text{IA}} p \text{ and } r \sqsubseteq_{\text{IA}} q\} \cup \sqsubseteq_{\text{IA}}$ is an alternating simulation relation. Let $(r,p) \in \mathcal{R}$; the case $r \sqsubseteq_{\text{IA}} p$ is obvious, so we consider the following cases:

(1) $p \wedge q \xrightarrow{a}$ with $a \in I$:
- **(I1):** $p \wedge q \xrightarrow{a} p'$ due to $p \xrightarrow{a}_P p'$ and $q \not\xrightarrow{a}_Q$. Then, $r \xrightarrow{a}_R r'$ for some r' with $r' \sqsubseteq_{\text{IA}} p'$ due to $r \sqsubseteq_{\text{IA}} p$, and we are done since $(r', p') \in \mathcal{R}$.
- **(I2):** Analogous to Case (I1).
- **(I3):** $p \wedge q \xrightarrow{a} p' \wedge q'$ due to $p \xrightarrow{a}_P p'$ and $q \xrightarrow{a}_Q q'$. Then, $r \xrightarrow{a}_R r'$ for some r' with $r' \sqsubseteq_{\text{IA}} p'$ due to $r \sqsubseteq_{\text{IA}} p$. By input-determinism and $r \sqsubseteq_{\text{IA}} q$, we also have $r' \sqsubseteq_{\text{IA}} q'$ and are done since $(r', p' \wedge q') \in \mathcal{R}$.

(2) $r \xrightarrow{\alpha}_R r'$ with $\alpha \in O \cup \{\tau\}$:
- $\alpha \in O$*:* Due to $r \sqsubseteq_{\text{IA}} p$ and $r \sqsubseteq_{\text{IA}} q$ we have p', q' such that $p \stackrel{\alpha}{\Longrightarrow}_P p'$, $q \stackrel{\alpha}{\Longrightarrow}_Q q'$, $r' \sqsubseteq_{\text{IA}} p'$ and $r' \sqsubseteq_{\text{IA}} q'$, i.e., $(r', p' \wedge q') \in \mathcal{R}$. We can interleave the τ-transitions of the two transition sequences by Rules (T1) and (T2) and finally synchronize the two α-transitions according to Rule (O), and obtain $p \wedge q \stackrel{\alpha}{\Longrightarrow} p' \wedge q'$.
- $\alpha = \tau$*:* Analogous, but without the synchronized transition. □

Technically, this result states that $\wedge$ gives the greatest lower-bound wrt. $\sqsubseteq_{\text{IA}}$ (up to equivalence), and its proof uses the input-determinism property of IA. The theorem also implies compositional reasoning; from universal algebra one easily gets:

Corollary 2.5. *For IAs P, Q, R with states p, q and r: $p \sqsubseteq_{IA} q \implies p \wedge r \sqsubseteq_{IA} q \wedge r$.*

Proof. Assume $p \sqsubseteq_{\text{IA}} q$. Then, (always) $p \wedge r \sqsubseteq_{\text{IA}} p \wedge r \iff$ (by Thm. 2.4) $p \wedge r \sqsubseteq_{\text{IA}} p$ and $p \wedge r \sqsubseteq_{\text{IA}} r \implies$ (by assumption and transitivity) $p \wedge r \sqsubseteq_{\text{IA}} q$ and $p \wedge r \sqsubseteq_{\text{IA}} r \iff$ (by Thm. 2.4) $p \wedge r \sqsubseteq_{\text{IA}} q \wedge r$. □

2.2. **Disjunction on IA.** In analogy to conjunction we develop a disjunction operator on IA and discuss its properties; in particular, this operator should give the least upper bound.

Definition 2.6 (Disjunction on IA)**.** Let $P = (P, I, O, \longrightarrow_P)$ and $Q = (Q, I, O, \longrightarrow_Q)$ be IAs with common input and output alphabets and disjoint state sets P and Q. The disjunction $P \vee Q$ is defined by $(\{p \vee q \mid p \in P, q \in Q\} \cup P \cup Q, I, O, \longrightarrow)$, where $\longrightarrow$ is the least set satisfying $\longrightarrow_P \subseteq \longrightarrow$, $\longrightarrow_Q \subseteq \longrightarrow$ and the following operational rules:

(I)	$p \vee q \xrightarrow{a} p' \vee q'$	if $p \xrightarrow{a}_P p'$, $q \xrightarrow{a}_Q q'$ and $a \in I$
(OT1)	$p \vee q \xrightarrow{\alpha} p'$	if $p \xrightarrow{\alpha}_P p'$ and $\alpha \in O \cup \{\tau\}$
(OT2)	$p \vee q \xrightarrow{\alpha} q'$	if $q \xrightarrow{\alpha}_Q q'$ and $\alpha \in O \cup \{\tau\}$

Note that this definition preserves the input-determinism required of IA. The definition is roughly dual to the one of IA-conjunction, i.e., we take the 'intersection' of initial input behaviour and the 'union' of initial output behaviour. Strictly speaking, this would require the following additional rule for outputs $o \in O$:

(O3) $p \vee q \xrightarrow{o} p' \vee q'$ if $p \xrightarrow{o}_P p'$ and $q \xrightarrow{o}_Q q'$

However, the addition of this rule would in general result in disjunctions $p \vee q$ that are larger than the least upper bound of p and q wrt. $\sqsubseteq_{\text{IA}}$. The following theorem shows that our $\vee$-operator properly characterizes the least upper bound:

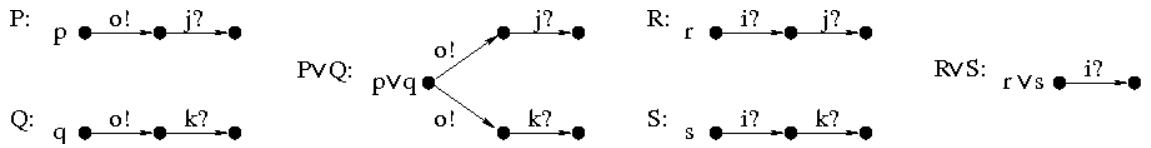

Figure 2: Example illustrating IA-disjunction's different treatment of inputs and outputs.

Theorem 2.7 ($\vee$ is Or)**.** *Let P, Q, R be IAs with states p, q and r, resp. Then, $p \vee q \sqsubseteq_{IA} r$ if and only if $p \sqsubseteq_{IA} r$ and $q \sqsubseteq_{IA} r$.*

Proof. "$\Longrightarrow$": We prove that $\mathcal{R} =_{\text{df}} \{(p, r) \mid \exists q.\, p \vee q \sqsubseteq_{\text{IA}} r\} \cup \sqsubseteq_{\text{IA}}$ is an alternating simulation relation. We let $(p, r) \in \mathcal{R}$ due to q – the case $p \sqsubseteq_{\text{IA}} r$ is obvious – and check the conditions of Def. 2.2:

- Let $r \xrightarrow{a}_R r'$ with $a \in I$. Hence, by $p \vee q \sqsubseteq_{\text{IA}} r$ and the only applicable Rule (I), $p \vee q \xrightarrow{a} p' \vee q'$ due to $p \xrightarrow{a}_P p'$ and $q \xrightarrow{a}_Q q'$ with $p' \vee q' \sqsubseteq_{\text{IA}} r'$. Since $(p', r') \in \mathcal{R}$ we are done.
- Let $p \xrightarrow{\alpha}_P p'$ with $\alpha \in O \cup \{\tau\}$. Hence, $p \vee q \xrightarrow{\alpha} p'$ by Rule (OT1) and, due to $p \vee q \sqsubseteq_{\text{IA}} r$, there exists some r' such that $r \stackrel{\hat{\alpha}}{\Longrightarrow} r'$ and $p' \sqsubseteq_{\text{IA}} r'$.

"$\Longleftarrow$": We show that $\mathcal{R} =_{\text{df}} \{(p \vee q, r) \mid p \sqsubseteq_{\text{IA}} r \text{ and } q \sqsubseteq_{\text{IA}} r\} \cup \sqsubseteq_{\text{IA}}$ is an alternating simulation relation. We let $(p \vee q, r) \in \mathcal{R}$ and consider the following cases:

(1) Let $r \xrightarrow{a}_R r'$ with $a \in I$. By $p \sqsubseteq_{\text{IA}} r$ and $q \sqsubseteq_{\text{IA}} r$ we have p' and q' such that $p \xrightarrow{a}_P p'$, $q \xrightarrow{a}_Q q'$, $p' \sqsubseteq_{\text{IA}} r'$ and $q' \sqsubseteq_{\text{IA}} r'$. Thus, we are done since $p \vee q \xrightarrow{a} p' \vee q'$ using Rule (I) and since $(p' \vee q', r') \in \mathcal{R}$.

(2) $p \vee q \xrightarrow{\alpha} p'$ with $\alpha \in O \cup \{\tau\}$. W.l.o.g., $p \xrightarrow{\alpha}_P p'$ due to Rule (OT1). Then, $r \stackrel{\hat{\alpha}}{\Longrightarrow}_R r'$ for some r' satisfying $p' \sqsubseteq_{\text{IA}} r'$, by $p \sqsubseteq_{\text{IA}} r$. □

Compositionality of disjunction can now be derived dually to the proof of Corollary 2.5 but using Thm. 2.7 instead of Thm. 2.4:

Corollary 2.8. *For IAs P, Q, R with states p, q and r: $p \sqsubseteq_{IA} q \implies p \vee r \sqsubseteq_{IA} q \vee r$.* □

The two examples of Fig. 2 round off our investigation of IA disjunction by illustrating the operator's different treatment of inputs and outputs. Regarding $p \vee q$ on the figure's left-hand side, the choice of which disjunct to implement is taken with the first action $o \in O$ if both disjuncts are implemented; this meets the intuition of an inclusive-or. In the analogous situation of $r \vee s$ on the figure's right-hand side, a branching on $i \in I$ is not allowed due to input-determinism, and the resulting IA is thus intuitively unsatisfactory. The root cause for this is that the IA-setting does not include sufficiently many automata and, therefore, the least upper bound is 'too large'. The shortcoming can be remedied by introducing disjunctive transitions, as we will do below in the dMTS- and MIA-settings. Then, we will have more automata and, indeed, will get a smaller least upper bound.

2.3. **Parallel Composition on IA.** We recall the parallel composition operator | on IA of [dH05], which is defined in two stages: first a standard product $\otimes$ between two IAs is introduced, where common actions are synchronized and hidden. Then, error states are identified, and all states are pruned from which reaching an error state is unavoidable.

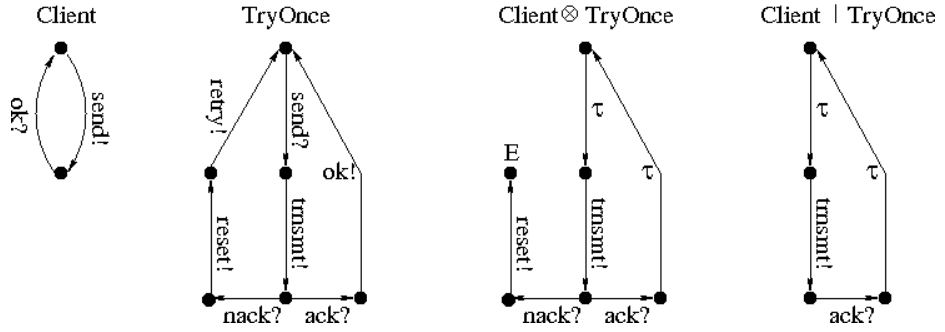

Figure 3: Example illustrating IA-parallel composition, where IA *TryOnce* has inputs {*send, ack, nack*} and outputs {*trnsmt, ok, reset, retry*}, while IA *Client* has inputs {*ok, retry*} and outputs {*send*}.

Definition 2.9 (Parallel Product on IA [dH05])**.** IAs P_1 and P_2 are called *composable* if $A_1 \cap A_2 = (I_1 \cap O_2) \cup (O_1 \cap I_2)$, i.e., each common action is input of one IA and output of the other IA. For such IAs we define the *product* $P_1 \otimes P_2 = (P_1 \times P_2, I, O, \longrightarrow)$, where $I = (I_1 \cup I_2) \setminus (O_1 \cup O_2)$ and $O = (O_1 \cup O_2) \setminus (I_1 \cup I_2)$ and where $\longrightarrow$ is given by the following operational rules:

(Par1) $(p_1, p_2) \xrightarrow{\alpha} (p_1', p_2)$ if $p_1 \xrightarrow{\alpha} p_1'$ and $\alpha \notin A_2$
(Par2) $(p_1, p_2) \xrightarrow{\alpha} (p_1, p_2')$ if $p_2 \xrightarrow{\alpha} p_2'$ and $\alpha \notin A_1$
(Par3) $(p_1, p_2) \xrightarrow{\tau} (p_1', p_2')$ if $p_1 \xrightarrow{a} p_1'$ and $p_2 \xrightarrow{a} p_2'$ for some a.

Note that, in case of synchronization and according to Rule (Par3), one only gets internal τ-transitions.

Definition 2.10 (Parallel Composition on IA [dH05])**.** A state (p_1, p_2) of a parallel product $P_1 \otimes P_2$ is an *error state* if there is some $a \in A_1 \cap A_2$ such that (a) $a \in O_1$, $p_1 \xrightarrow{a}$ and $p_2 \not\xrightarrow{a}$, or (b) $a \in O_2$, $p_2 \xrightarrow{a}$ and $p_1 \not\xrightarrow{a}$.

A state of $P_1 \otimes P_2$ is *incompatible* if it may reach an error state autonomously, i.e., only by output or internal actions that are, intuitively, locally controlled. Formally, the set $E \subseteq P_1 \times P_2$ of incompatible states is the least set such that $(p_1, p_2) \in E$ if (i) (p_1, p_2) is an error state or (ii) $(p_1, p_2) \xrightarrow{\alpha} (p_1', p_2')$ for some $\alpha \in O \cup \{\tau\}$ and $(p_1', p_2') \in E$.

The *parallel composition* $P_1|P_2$ of P_1, P_2 is obtained from $P_1 \otimes P_2$ by *pruning*, i.e., removing all states in E and all transitions involving such states as source or target. If $(p_1, p_2) \in P_1|P_2$, we write $p_1|p_2$ and call p_1 and p_2 *compatible*.

Parallel composition is well-defined since input-determinism is preserved.

Theorem 2.11 (Compositionality of IA-Parallel Composition [dH05])**.** *Let P_1, P_2 and Q_1 be IAs with $p_1 \in P_1$, $p_2 \in P_2$, $q_1 \in Q_1$ and $p_1 \sqsubseteq_{IA} q_1$. Assume that Q_1 and P_2 are composable; then, (a) P_1 and P_2 are composable and (b) if q_1 and p_2 are compatible, then so are p_1 and p_2 and $p_1|p_2 \sqsubseteq_{IA} q_1|p_2$.* □

This result relies on the fact that IAs are input-deterministic. While the theorem is already stated in [dH05], its proof is only sketched therein. Here, it is a simple corollary of Thm. 4.14 in Sec. 4.3 and Thms. 4.16 and 4.17(b) in Sec. 4.4 below.

We conclude by presenting a small example of IA-parallel composition in Fig. 3, which is adapted from [dH05]. *Client* does not accept its input *retry*. Thus, if the environment of $\mathit{Client} \otimes \mathit{TryOnce}$ would produce *nack*, the system would autonomously produce *reset* and run into a catastrophic error. To avoid this, the environment of $\mathit{Client} | \mathit{TryOnce}$ is required not to produce *nack*. This view is called optimistic: there exists an environment in which *Client* and *TryOnce* can cooperate without errors, and $\mathit{Client} | \mathit{TryOnce}$ describes the necessary requirements for such an environment. In the pessimistic view as advocated in [BHW11], *Client* and *TryOnce* are regarded as incompatible due to the potential error.

3. Conjunction and Disjunction for Modal Transition Systems

Modal Transition Systems (MTS) were investigated by Larsen [Lar90] as a specification framework based on labelled transition systems but with two kinds of transitions: must-transitions specify required behaviour, may-transitions specify allowed behaviour, and absent transitions specify forbidden behaviour. Any refinement of an MTS-specification must preserve required and forbidden behaviour and may turn allowed behaviour into required or forbidden behaviour. Technically, this is achieved via an alternating-style simulation relation, called *modal refinement*, where any must-transition of the specification must be simulated by an implementation, while any may-transition of the implementation must be simulated by the specification.

Our aim in this section is to extend MTS with conjunction and also disjunction. Larsen [Lar90] first defined conjunction and disjunction on MTS (without τ), but the resulting systems often violate syntactic consistency (they are not really MTSs) and are hard to understand. This construction was subsequently generalized by Larsen and Xinxin to Disjunctive MTS (DMTS) [LX90], again ignoring syntactic consistency. This shortcoming was recently fixed by Beneš et al. [BCK11] by exploiting the fact that an a-must-transition in a DMTS may have several alternative target states. However, this work does still not consider a weak setting, i.e., systems with τ. Below, we will define conjunction and disjunction on a syntactically consistent subclass of DMTS, called *dMTS*, but more generally in a weak setting as defined in [dH05, LNW07]; this subclass is sufficient for the purposes of the present article, and we leave the extension of our results to DMTS for future work. Since the treatment of τ-transitions is non-trivial and non-standard, we will motivate and explain it in detail.

Note that this section will not consider parallel composition for (d)MTS. This is because we are working towards the MIA-setting that will be introduced in the next section, which like IA and unlike (d)MTS distinguishes between inputs and outputs. (d)MTS parallel composition can simply be defined in the style similar to Def. 2.9; in particular, it does not have error states and thus fundamentally differs from conjunction as defined below.

3.1. Disjunctive Modal Transition Systems.

We extend standard MTS only as far as needed for defining conjunction and disjunction, by introducing disjunctive must-transitions that are disjunctive wrt. exit states only (see Fig. 5). The following extension also has no τ-must-transitions since these are not considered in the definition of the observational modal refinement of [LNW07].

Definition 3.1 (disjunctive Modal Transition System)**.** A *disjunctive Modal Transition System* (dMTS) is a tuple $Q = (Q, A, \longrightarrow, \dashrightarrow)$, where

(1) Q is a set of states,
(2) A is an alphabet not containing the special, silent action τ,
(3) $\longrightarrow \subseteq Q \times A \times (\mathcal{P}(Q) \setminus \emptyset)$ is the *must-transition* relation,
(4) $\dashrightarrow \subseteq Q \times (A \cup \{\tau\}) \times Q$ is the *may-transition* relation.

We require *syntactic consistency*, i.e., $q \xrightarrow{a} Q'$ implies $\forall q' \in Q'.\, q \overset{a}{\dashrightarrow} q'$.

More generally, the must-transition relation in a standard DMTS [LX90] may be a subset of $Q \times (\mathcal{P}(A \times Q) \setminus \emptyset)$. For notational convenience, we write $q \xrightarrow{a} q'$ whenever $q \xrightarrow{a} \{q'\}$; all must-transitions in standard MTS have this form.

Our refinement relation on dMTS abstracts from internal computation steps in the same way as [LNW07], i.e., by considering the following *weak may-transitions* for $\alpha \in A \cup \{\tau\}$: $q \overset{\varepsilon}{=\!=\!\Rightarrow} q'$ if $q \overset{\tau}{\dashrightarrow}{}^{*} q'$, and $q \overset{\alpha}{=\!=\!\Rightarrow} q'$ if $\exists q''.\, q \overset{\varepsilon}{=\!=\!\Rightarrow} q'' \overset{\alpha}{\dashrightarrow} q'$.

Definition 3.2 (Observational Modal Refinement, see [LNW07])**.** Let P, Q be dMTSs. Relation $\mathcal{R} \subseteq P \times Q$ is an *(observational) modal refinement relation* if for all $(p, q) \in \mathcal{R}$:

(i): $q \xrightarrow{a} Q'$ implies $\exists P'.\, p \xrightarrow{a} P'$ and $\forall p' \in P'\, \exists q' \in Q'.\, (p', q') \in \mathcal{R}$,

(ii): $p \overset{\alpha}{\dashrightarrow} p'$ implies $\exists q'.\, q \overset{\hat{\alpha}}{=\!=\!\Rightarrow} q'$ and $(p', q') \in \mathcal{R}$.

We write $p \sqsubseteq_{\mathrm{dMTS}} q$ and say that p *dMTS-refines* q if there exists an observational modal refinement relation $\mathcal{R}$ such that $(p, q) \in \mathcal{R}$.

Again, $\sqsubseteq_{\mathrm{dMTS}}$ is a preorder and the largest observational modal refinement relation. Except for disjunctiveness, dMTS-refinement is exactly defined as for MTS in [LNW07]. In the following figures, any (disjunctive) must-transition drawn also represents implicitly the respective may-transition(s), unless explicitly stated otherwise.

3.2. **Conjunction on dMTS.** Technically similar to parallel composition for IA, conjunction will be defined in two stages. State pairs can be logically inconsistent due to unsatisfiable must-transitions; in the second stage, we remove such pairs incrementally.

Definition 3.3 (Conjunctive Product on dMTS)**.** Let $P = (P, A, \longrightarrow_P, \dashrightarrow_P)$ and $Q = (Q, A, \longrightarrow_Q, \dashrightarrow_Q)$ be dMTSs with common alphabet. The conjunctive product $P \& Q =_{\mathrm{df}} (P \times Q, A, \longrightarrow, \dashrightarrow)$ is defined by its operational transition rules as follows:

(Must1)	$(p,q) \xrightarrow{a} \{(p',q') \mid p' \in P',\, q \overset{a}{=\!=\!\Rightarrow}_Q q'\}$	if	$p \xrightarrow{a}_P P'$ and $q \overset{a}{=\!=\!\Rightarrow}_Q$
(Must2)	$(p,q) \xrightarrow{a} \{(p',q') \mid p \overset{a}{=\!=\!\Rightarrow}_P p',\, q' \in Q'\}$	if	$p \overset{a}{=\!=\!\Rightarrow}_P$ and $q \xrightarrow{a}_Q Q'$
(May1)	$(p,q) \overset{\tau}{\dashrightarrow} (p',q)$	if	$p \overset{\tau}{=\!=\!\Rightarrow}_P p'$
(May2)	$(p,q) \overset{\tau}{\dashrightarrow} (p,q')$	if	$q \overset{\tau}{=\!=\!\Rightarrow}_Q q'$
(May3)	$(p,q) \overset{\alpha}{\dashrightarrow} (p',q')$	if	$p \overset{\alpha}{=\!=\!\Rightarrow}_P p'$ and $q \overset{\alpha}{=\!=\!\Rightarrow}_Q q'$

It might be surprising that a single transition in the product might stem from a transition sequence in one of the components (cf. the first four items above) and that the components can also synchronize on τ (cf. Rule (May3)). The necessity of this is discussed below; we only repeat here that conjunction is inherently different from parallel composition where, for instance, there is no synchronization on τ.

Definition 3.4 (Conjunction on dMTS)**.** Given a conjunctive product $P \& Q$, the set $F \subseteq P \times Q$ of *(logically) inconsistent states* is defined as the least set satisfying the following rules:

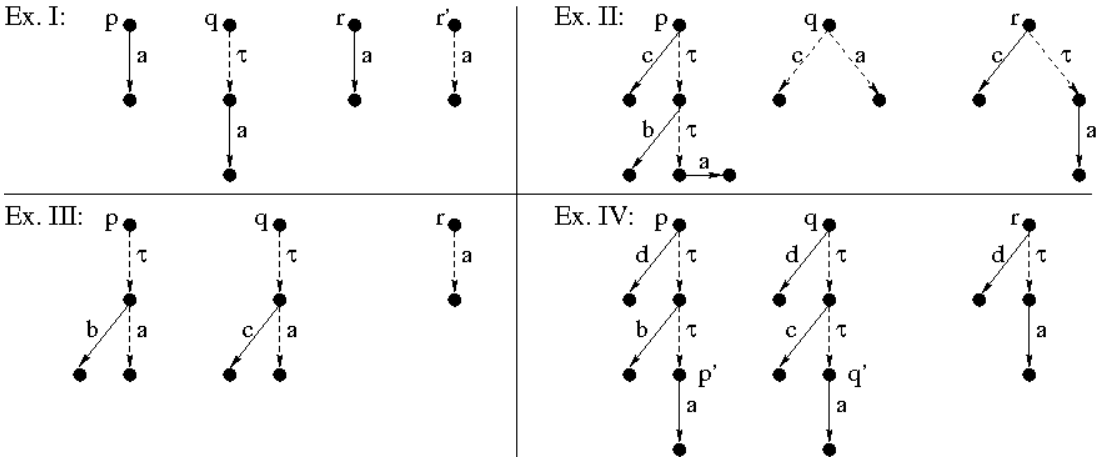

Figure 4: Examples motivating the rules of Def. 3.3.

(F1)	$p \xrightarrow{a}_P$, $q \not\stackrel{a}{=\!=\!\Rightarrow}_Q$	implies	$(p,q) \in F$
(F2)	$p \not\stackrel{a}{=\!=\!\Rightarrow}_P$, $q \xrightarrow{a}_Q$	implies	$(p,q) \in F$
(F3)	$(p,q) \xrightarrow{a} R'$ and $R' \subseteq F$	implies	$(p,q) \in F$

The conjunction $P \wedge Q$ of dMTSs P, Q is obtained by deleting all states $(p,q) \in F$ from $P \& Q$. This also removes any may- or must-transition exiting a deleted state and any may-transition entering a deleted state; in addition, deleted states are removed from targets of disjunctive must-transitions. We write $p \wedge q$ for the state (p,q) of $P \wedge Q$; these are the consistent states by construction, and $p \wedge q$ is only defined for such a state.

Regarding well-definedness, first observe that $P \& Q$ is a dMTS, where syntactic consistency follows from Rule (May3). Now, $P \wedge Q$ is a dMTS, too: if R' becomes empty for some $(p,q) \xrightarrow{a} R'$, then also (p,q) is deleted when constructing $P \wedge Q$ from $P \& Q$ according to (F3). Finally, our conjunction operator is also commutative and associative.

Before we formally state that operator $\wedge$ is indeed conjunction on dMTS, we present several examples depicted in Fig. 4, which motivate the rules of Def. 3.3. In each case, r is a common implementation of p and q (but not r' in Ex. I), whence these must be logically consistent. Thus, Ex. I explains Rule (Must1). If we only had $\stackrel{\tau}{\dashrightarrow}$ in the precondition of Rule (May1), $p \wedge q$ of Ex. II would just consist of a c-must- and an a-may-transition; the only τ-transition would lead to a state in F due to b. This would not allow the τ-transition of r, explaining Rule (May1). In Ex. III and with only $\stackrel{\alpha}{\dashrightarrow}$ in the preconditions of Rule (May3), $p \wedge q$ would just have three τ-transitions to inconsistent states (due to b, c, resp.). This explains the weak transitions for $\alpha \neq \tau$ in Rule (May3). According to Rules (May1) and (May2), $p \wedge q$ in Ex. IV has four τ-transitions to states in F (due to d). With preconditions based on at least one $\stackrel{\tau}{\dashrightarrow}$ instead of $\stackrel{\tau}{=\!=\!\Rightarrow}$ in the τ-case of Rule (May3), there would be three more τ-transitions to states in F (due to b or c). Thus, it is essential that Rule (May3) also allows the synchronization of two weak τ-transitions, which in this case gives $p \wedge q \stackrel{\tau}{\dashrightarrow} p' \wedge q'$.

Fig. 5 shows a small example illustrating the treatment of disjunctive must-transitions in the presence of inconsistency. In $P \& Q$, the a-must-transition of Q combines with the three a-transitions of P to a truly disjunctive must-transition with a three-element target set.

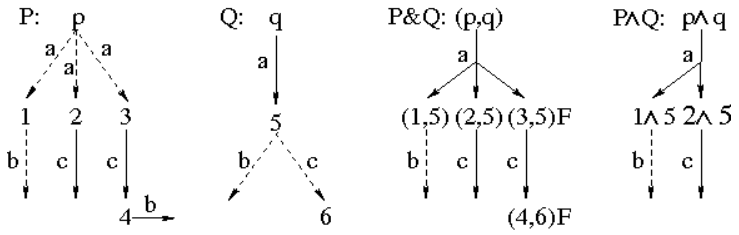

Figure 5: Example illustrating dMTS-conjunction.

The inconsistency of state $(4,6)$ due to b propagates back to state $(3,5)$. The inconsistent states are then removed in $P \wedge Q$.

Theorem 3.5 ($\wedge$ is And)**.** *Let P, Q, R be dMTSs. Then, (i) ($\exists r \in R.\, r \sqsubseteq_{dMTS} p$ and $r \sqsubseteq_{dMTS} q$) if and only if $p \wedge q$ is defined. In addition, in case $p \wedge q$ is defined: (ii) $r \sqsubseteq_{dMTS} p$ and $r \sqsubseteq_{dMTS} q$ if and only if $r \sqsubseteq_{dMTS} p \wedge q$.*

This key theorem states in Item (ii) that conjunction behaves as it should, i.e., $\wedge$ on dMTSs is the greatest lower bound wrt. $\sqsubseteq_{\text{dMTS}}$. Item (i) concerns the intuition that two specifications p and q are logically inconsistent if they do not have a common implementation; formally, $p \wedge q$ is undefined in this case. Alternatively, we could have added an explicit inconsistent element *ff* to our setting, so that $p \wedge q = ff$. This element *ff* would be defined to be a refinement of every p' and equivalent to any $(p', q') \in F$ of some $P\&Q$. Additionally, $ff \wedge p'$ and $p' \wedge ff$ would be defined as *ff*, for any p'.

The proof of the above theorem requires us to first introduce the following concept for formally reasoning about inconsistent states:

Definition 3.6 (dMTS-Witness)**.** A *dMTS-witness* W of $P\&Q$ is a subset of $P \times Q$ such that the following conditions hold for all $(p, q) \in W$:

(W1) $p \xrightarrow{a}_P$ implies $q \overset{a}{=\!=\!\not\to}_Q$

(W2) $q \xrightarrow{a}_Q$ implies $p \overset{a}{=\!=\!\not\to}_P$

(W3) $(p, q) \xrightarrow{a} R'$ implies $R' \cap W \neq \emptyset$

Conditions (W1)–(W3) correspond to the negations of the premises of Conditions (F1)–(F3) in Def. 3.4. This implies Part (i) of the following lemma, while Part (ii) is essential for proving Thm. 3.5(i):

Lemma 3.7 (Concrete dMTS-Witness)**.** *Let $P\&Q$ be a conjunctive product of dMTSs and R be a dMTS.*

(i): *For any dMTS-witness W of $P\&Q$, we have $F \cap W = \emptyset$.*

(ii): *The set $\{(p, q) \in P \times Q \mid \exists r \in R.\, r \sqsubseteq_{dMTS} p \text{ and } r \sqsubseteq_{dMTS} q\}$ is a dMTS-witness of $P\&Q$.*

Proof. While the first statement of the lemma is quite obvious, we prove here that $W =_{\text{df}} \{(p, q) \in P \times Q \mid \exists r \in R.\, r \sqsubseteq_{\text{dMTS}} p \text{ and } r \sqsubseteq_{\text{dMTS}} q\}$ is a dMTS-witness of $P\&Q$ according to Def. 3.6:

(W1): $p \xrightarrow{a}_P P'$ implies $r \xrightarrow{a}_R R'$ by $r \sqsubseteq_{\text{dMTS}} p$. Choose some $r' \in R'$. Then, $r \overset{a}{\dashrightarrow}_R r'$ by syntactic consistency and $q \overset{a}{=\!=\!\not\to}_Q$ by $r \sqsubseteq_{\text{dMTS}} q$.

(W2): Analogous to (W1).

(W3): Consider $(p,q) \in W$ due to r, with $(p,q) \xrightarrow{a} S'$ due to $p \xrightarrow{a}_P P'$ and $S' = \{(p',q') \mid p' \in P',\, q \overset{a}{=\!=\!\twoheadrightarrow}_Q q'\}$ according to Rule (Must1). By $r \sqsubseteq_{\text{dMTS}} p$ we get some $R' \subseteq R$ such that $r \xrightarrow{a}_R R'$ and $\forall r' \in R'\, \exists p' \in P'.\, r' \sqsubseteq_{\text{dMTS}} p'$. Choose $r' \in R'$; now, $r \overset{a}{\dashrightarrow}_R r'$ due to syntactic consistency, and $q \overset{a}{=\!=\!\twoheadrightarrow}_Q q'$ with $r' \sqsubseteq_{\text{dMTS}} q'$ by $r \sqsubseteq_{\text{dMTS}} q$. Thus, we have $p' \in P'$ and q' such that $(p',q') \in W \cap S'$ due to r'. □

We are now able to prove Thm. 3.5:

Proof. (i)"$\Longrightarrow$": This follows from Lemma 3.7.

(i), (ii)"$\Longleftarrow$": It suffices to show that $\mathcal{R} =_{\text{df}} \{(r,p) \mid \exists q.\, r \sqsubseteq_{\text{dMTS}} p \wedge q\}$ is an observational modal refinement relation. Then, in particular, (i)"$\Longleftarrow$" follows by choosing $r = p \wedge q$. We check the two conditions of Def. 3.2:

- Let $p \xrightarrow{a}_P P'$; then, $q \overset{a}{=\!=\!\twoheadrightarrow}_Q$ since, otherwise, $p \wedge q$ would not be defined due to (F1). Hence, by Rule (Must1), $p \wedge q \xrightarrow{a} \{p' \wedge q' \mid p' \in P',\, q \overset{a}{=\!=\!\twoheadrightarrow}_Q q',\, p' \wedge q' \text{ defined}\}$. By $r \sqsubseteq_{\text{dMTS}} p \wedge q$, we get $r \xrightarrow{a}_R R'$ such that $\forall r' \in R'\ \exists p' \wedge q'.\ p' \in P',\ q \overset{a}{=\!=\!\twoheadrightarrow}_Q q'$ and $r' \sqsubseteq_{\text{dMTS}} p' \wedge q'$. Hence, $\forall r' \in R'\, \exists p' \in P'.\, (r',p') \in \mathcal{R}$.
- $r \overset{\alpha}{\dashrightarrow}_R r'$ implies $\exists p' \wedge q'.\, p \wedge q \overset{\hat{\alpha}}{=\!=\!\twoheadrightarrow} p' \wedge q'$ and $r' \sqsubseteq_{\text{dMTS}} p' \wedge q'$. The contribution of p in this weak transition sequence gives $p \overset{\hat{\alpha}}{=\!=\!\twoheadrightarrow}_P p'$, and we have $(r',p') \in \mathcal{R}$ due to q'.

(ii)"$\Longrightarrow$": Here, we show that $\mathcal{R} = \{(r, p \wedge q) \mid r \sqsubseteq_{\text{dMTS}} p \text{ and } r \sqsubseteq_{\text{dMTS}} q\}$ is an observational modal refinement relation. By Part (i), $p \wedge q$ is defined and $(r, p \wedge q) \in \mathcal{R}$ whenever $r \sqsubseteq_{\text{dMTS}} p$ and $r \sqsubseteq_{\text{dMTS}} q$. We now verify the conditions of Def. 3.2:

- Let $p \wedge q \xrightarrow{a} S'$, w.l.o.g. due to $p \xrightarrow{a}_P P'$ and $S' = \{p' \wedge q' \mid p' \in P',\, q \overset{a}{=\!=\!\twoheadrightarrow}_Q q',\, p' \wedge q' \text{ defined}\}$. Because of $r \sqsubseteq_{\text{dMTS}} p$, we have $r \xrightarrow{a}_R R'$ so that $\forall r' \in R'\, \exists p' \in P'.\, r' \sqsubseteq_{\text{dMTS}} p'$. Consider some arbitrary $r' \in R'$ and the respective $p' \in P'$. Then, $r \overset{a}{\dashrightarrow}_R r'$ by syntactic consistency and, due to $r \sqsubseteq_{\text{dMTS}} q$, there exists some q' with $q \overset{a}{=\!=\!\twoheadrightarrow}_Q q'$ and $r' \sqsubseteq_{\text{dMTS}} q'$. Thus, $p' \wedge q' \in S'$ and $(r', p' \wedge q') \in \mathcal{R}$.
- Let $r \overset{\alpha}{\dashrightarrow}_R r'$ and consider $p \overset{\hat{\alpha}}{=\!=\!\twoheadrightarrow}_P p'$ and $q \overset{\hat{\alpha}}{=\!=\!\twoheadrightarrow}_Q q'$ satisfying $r' \sqsubseteq_{\text{dMTS}} p'$ and $r' \sqsubseteq_{\text{dMTS}} q'$. Thus, $(r', p' \wedge q') \in \mathcal{R}$. Further, if $\alpha \neq \tau$, we have $p \wedge q \overset{\alpha}{\dashrightarrow} p' \wedge q'$ by Rule (May3). Otherwise, either $p \overset{\tau}{=\!=\!\twoheadrightarrow}_P p'$ and $q \overset{\tau}{=\!=\!\twoheadrightarrow}_Q q'$ and we are done by Rule (May3) again, or w.l.o.g. $p \overset{\tau}{=\!=\!\twoheadrightarrow}_P p'$ and $q = q'$ and we are done by Rule (May1), or $p = p'$ and $q = q'$. □

The following corollary of Thm. 3.5 now easily follows:

Corollary 3.8. *dMTS-refinement is compositional wrt. conjunction, i.e., if $p \sqsubseteq_{dMTS} q$ and $p \wedge r$ is defined, then $q \wedge r$ is defined and $p \wedge r \sqsubseteq_{dMTS} q \wedge r$.*

Proof. Assume $p \sqsubseteq_{\text{dMTS}} q$ and $p \wedge r$ is defined. Then, (always) $p \wedge r \sqsubseteq_{\text{dMTS}} p \wedge r \Longleftrightarrow$ (by Thm. 3.5) $p \wedge r \sqsubseteq_{\text{dMTS}} p$ and $p \wedge r \sqsubseteq_{\text{dMTS}} r \Longrightarrow$ (by assumption and transitivity) $p \wedge r \sqsubseteq_{\text{dMTS}} q$ and $p \wedge r \sqsubseteq_{\text{dMTS}} r \Longrightarrow$ (by Thm. 3.5(i)) $q \wedge r$ is defined and (by Thm. 3.5(ii)) $p \wedge r \sqsubseteq_{\text{dMTS}} q \wedge r$. □

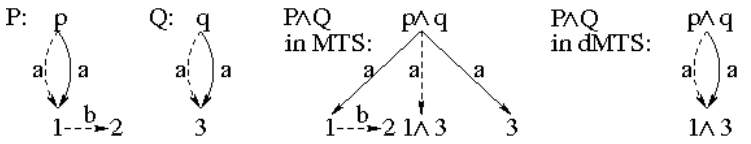

Figure 6: Example illustrating Larsen's MTS-conjunction; $\overset{a}{\dashrightarrow}$ drawn separately.

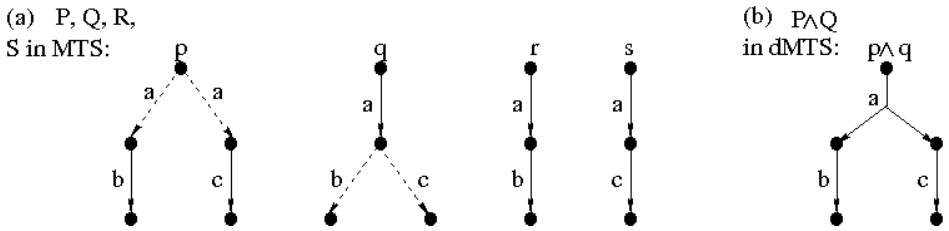

Figure 7: Example showing that conjunction cannot be defined on MTS. (A similar example is given in [BCK11] without proof.)

Thus, we have succeeded in our ambition to define a syntactically consistent conjunction for MTS, for a weak MTS-variant with disjunctive must-transitions.

Larsen [Lar90] also defines a conjunction operator on MTS, but almost always the result violates syntactic consistency. A simple example is shown in Fig. 6 where q refines p in Larsen's setting as well as in our dMTS-setting; in this figure, may-transitions are drawn explicitly, i.e, a must- is not necessarily also a may-transition. Since Larsen's $p \wedge q$ is not syntactically consistent, this $p \wedge q$ and q are, contrary to the first impression, equivalent. In our dMTS-setting, $P \wedge Q$ is isomorphic to Q which will also hold for our MIA-setting below (with action b read as output and where a could be either an input or an output).

Indeed, conjunction cannot be defined on MTS in general, e.g., for the P and Q in Fig. 7(a). The states p and q have r as well as s as common implementations; thus, r and s must be implementations of $p \wedge q$. An MTS $P \wedge Q$ would need in state $p \wedge q$ (i) an immediate a-must-transition (due to q) followed by (ii) a must-b and no c or a must-c and no b (due to p). In the first (second) case, s (r) is not an implementation of $p \wedge q$, which is a contradiction. Using dMTS, the conjunction $P \wedge Q$ is as shown in Fig. 7(b).

The above shortcoming of MTS has been avoided by Larsen et al. in [LSW95] by limiting conjunction to so-called *independent* specifications that make inconsistencies obsolete; this restriction also excludes the above example. Recently, Bauer et al. [BJL$^+$12] have defined conjunction for a version of MTS extended by partially ordered labels; when refining an MTS, also the labels can be refined, and this has various applications. However, the conjunction operator is only defined under some restriction, which corresponds to requiring determinism in the standard MTS-setting. Another MTS-inspired theory including a conjunction operator has been introduced by Raclet et al. [RBB$^+$11]. While their approach yields the desired $p \wedge q$ as in our dMTS-setting, it is language-based and thus deals with deterministic systems only.

3.3. **Disjunction on dMTS.** We will see in Sec. 3.4 that input-transitions (output-transitions) in IA correspond to must-transitions (may-transitions) in dMTS. In this light, the

following definition of disjunction corresponds closely to the one for IA. In particular, initial must-transitions are also combined, but this time the choice between disjuncts is not delayed.

Definition 3.9 (Disjunction on dMTS)**.** Let $P = (P, A, \longrightarrow_P, \dashrightarrow_P)$ and $Q = (Q, A, \longrightarrow_Q, \dashrightarrow_Q)$ be dMTSs with common alphabet. The disjunction $P \vee Q$ is defined as the tuple $(\{p \vee q \mid p \in P, q \in Q\} \cup P \cup Q, A, \longrightarrow, \dashrightarrow)$, where $\longrightarrow$ and $\dashrightarrow$ are the least sets satisfying $\longrightarrow_P \subseteq \longrightarrow$, $\dashrightarrow_P \subseteq \dashrightarrow$, $\longrightarrow_Q \subseteq \longrightarrow$, $\dashrightarrow_Q \subseteq \dashrightarrow$ and the following operational rules:

$$\begin{array}{lll} \text{(Must)} & p \vee q \xrightarrow{a} P' \cup Q' & \text{if } p \xrightarrow{a}_P P',\ q \xrightarrow{a}_Q Q' \\ \text{(May1)} & p \vee q \overset{\alpha}{\dashrightarrow} p' & \text{if } p \overset{\alpha}{\dashrightarrow}_P p' \\ \text{(May2)} & p \vee q \overset{\alpha}{\dashrightarrow} q' & \text{if } q \overset{\alpha}{\dashrightarrow}_Q q' \end{array}$$

This definition clearly yields well-defined dMTSs respecting syntactic consistency. It also gives us the desired least-upper-bound property:

Theorem 3.10 ($\vee$ is Or)**.** *Let P, Q, and R be dMTSs with states p, q and r, resp. Then, $p \vee q \sqsubseteq_{dMTS} r$ if and only if $p \sqsubseteq_{dMTS} r$ and $q \sqsubseteq_{dMTS} r$.*

Proof. "$\Longrightarrow$": We establish that $\mathcal{R} =_{\text{df}} \{(p, r) \mid \exists q.\, p \vee q \sqsubseteq_{\text{dMTS}} r\} \cup \sqsubseteq_{\text{dMTS}}$ is a modal refinement relation. To do so, we let $(p, r) \in \mathcal{R}$ due to q and check the conditions of Def. 3.2:

- **(i):** Let $r \xrightarrow{a}_R R'$. By $p \vee q \sqsubseteq_{\text{dMTS}} r$ and the only applicable Rule (Must), $p \vee q \xrightarrow{a} P' \cup Q'$ due to $p \xrightarrow{a}_P P'$ and $q \xrightarrow{a}_Q Q'$ such that $\forall p' \in P' \cup Q'\, \exists r' \in R'.\ p' \sqsubseteq_{\text{dMTS}} r'$. Hence, $\forall p' \in P'\, \exists r' \in R'.\ p' \sqsubseteq_{\text{dMTS}} r'$ and, thus, $(p', r') \in \mathcal{R}$.
- **(ii):** Let $p \overset{\alpha}{\dashrightarrow}_P p'$. Hence, $p \vee q \overset{\alpha}{\dashrightarrow} p'$ by Rule (May1) and, due to $p \vee q \sqsubseteq_{\text{dMTS}} r$, there exists some r' such that $r \overset{\hat{\alpha}}{=\!=\!\!\!\twoheadrightarrow} r'$ and $p' \sqsubseteq_{\text{dMTS}} r'$.

"$\Longleftarrow$": We prove that $\mathcal{R} =_{\text{df}} \{(p \vee q, r) \mid p \sqsubseteq_{\text{dMTS}} r \text{ and } q \sqsubseteq_{\text{dMTS}} r\} \cup \sqsubseteq_{\text{dMTS}}$ is a modal refinement relation. Let $(p \vee q, r) \in \mathcal{R}$ and consider the following cases:

- **(i):** Let $r \xrightarrow{a}_R R'$. By $p \sqsubseteq_{\text{dMTS}} r$ and $q \sqsubseteq_{\text{dMTS}} r$ we have P', Q' satisfying $p \xrightarrow{a}_P P'$, $q \xrightarrow{a}_Q Q'$ such that $\forall p' \in P'\, \exists r' \in R'.\ p' \sqsubseteq_{\text{dMTS}} r'$ and $\forall q' \in Q'\, \exists r' \in R'.\ q' \sqsubseteq_{\text{dMTS}} r'$. Thus, $p \vee q \xrightarrow{a} P' \cup Q'$ using Rule (Must) and we are done.
- **(ii):** $p \vee q \overset{\alpha}{\dashrightarrow} p'$. W.l.o.g., this is due to Rule (May1) and $p \overset{\alpha}{\dashrightarrow}_P p'$. Then, $r \overset{\hat{\alpha}}{=\!=\!\!\!\twoheadrightarrow}_R r'$ for some r' satisfying $p' \sqsubseteq_{\text{dMTS}} r'$, by $p \sqsubseteq_{\text{dMTS}} r$. □

Analogously to the IA-setting we may obtain the following corollary to the above theorem:

Corollary 3.11. *dMTS-refinement is compositional wrt. disjunction.* □

3.4. **Embedding of IA into dMTS.** We can now adopt the embedding of IA into MTS from [LNW07] to our setting:

Definition 3.12 (IA-Embedding)**.** Let P be an IA with $A = I \cup O$. Then, the embedding $[P]_{\text{dMTS}}$ of P into (d)MTS is defined as the (d)MTS $(P \cup \{u_P\}, A, \longrightarrow, \dashrightarrow)$, where $u_P \notin P$ and:

$$\begin{array}{lll} p \overset{\alpha}{\dashrightarrow} p' & \text{if} & p \xrightarrow{\alpha}_P p' \text{ and } \alpha \in A \cup \{\tau\}; \\ p \xrightarrow{a} p' & \text{if} & p \xrightarrow{a}_P p' \text{ and } a \in I; \\ p \overset{a}{\dashrightarrow} u_P & \text{if} & p \overset{a}{\not\longrightarrow}_P \text{ and } a \in I; \\ u_P \overset{a}{\dashrightarrow} u_P & \text{if} & a \in A. \end{array}$$

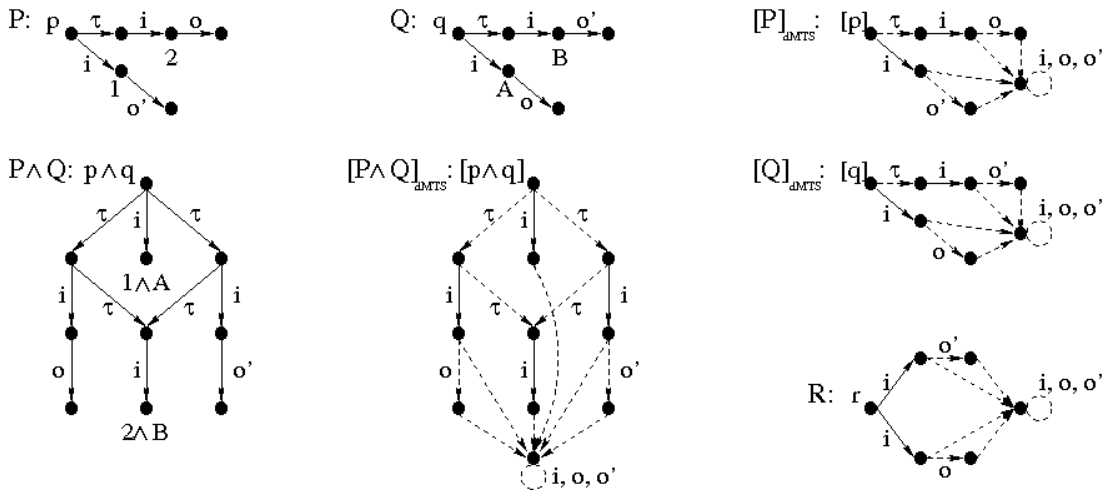

Figure 8: Example refuting the reverse refinement in Prop. 3.13(a). All non-labelled transitions depict i-may-transitions.

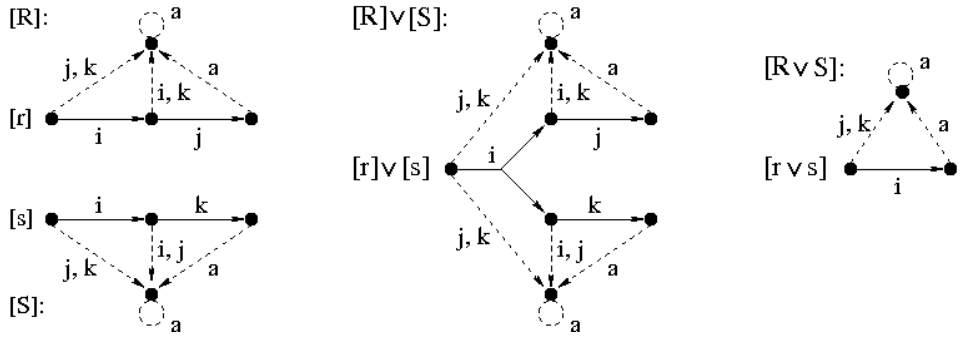

Figure 9: Example refuting the reverse refinement in Prop. 3.13(b) ($a \in A = \{i, j, k\}$).

For the remainder of this section we simply write $[p]$ for $p \in [P]_{\text{dMTS}}$. Observe that $[P]_{\text{dMTS}}$ does not have truly disjunctive transitions; hence, it is an MTS. In [LNW07], it is shown that this embedding respects refinement, i.e., $p \sqsubseteq_{\text{IA}} q$ if and only if $[p] \sqsubseteq_{\text{dMTS}} [q]$. Since conjunction (disjunction) on IA and dMTS is the greatest lower bound (least upper bound) wrt. $\sqsubseteq_{\text{IA}}$ and $\sqsubseteq_{\text{dMTS}}$ (up to equivalence), resp., we have by general order theory:

Proposition 3.13 (Conjunction/Disjunction and IA-Embedding)**.** *For all IAs P and Q with $p \in P$ and $q \in Q$:*

(a): $[p \wedge q] \sqsubseteq_{dMTS} [p] \wedge [q]$*;*

(b): $[p \vee q] \sqsupseteq_{dMTS} [p] \vee [q]$. □

The reverse refinements do not hold due to the additional dMTSs that are not embeddings of IA. To see this for conjunction, consider the example in Fig. 8, where P and Q are IAs. State r in dMTS R is a common implementation of state $[p]$ and state $[q]$, i.e., their conjunction is sufficiently large to cover r. However, r does not refine $[p \wedge q]$ since the initial i-must-transition of the latter cannot be matched by the former. Hence, $[p \wedge q]$ and $[p] \wedge [q]$

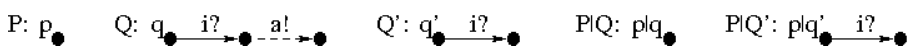

Figure 10: Example demonstrating the compositionality flaw of IOMTS.

cannot be equivalent. To see this for disjunction, consider r and s in Fig. 2 on the right. Fig. 9 shows all relevant dMTSs, and $[r \vee s]$ does not refine $[r] \vee [s]$ since it does not have a must-transition after i.

4. Modal Interface Automata

An essential point of Larsen, Nyman and Wasowski's paper [LNW07] is to enrich IA with modalities to get a flexible specification framework where inputs and outputs can be prescribed, allowed or prohibited. To do so, they consider IOMTS, i.e., MTS where visible actions are partitioned into inputs and outputs, and define parallel composition in IA-style.

Our example of Fig. 10 shows that their approach has a serious flaw, namely observational modal refinement is not a precongruence for the parallel composition of [LNW07]. In this example, the IOMTS P has input alphabet $\{a\}$ and empty output alphabet, while Q and Q' have input alphabet $\{i\}$ and output alphabet $\{a\}$. Obviously, $q' \sqsubseteq_{\text{dMTS}} q$. When composing P and Q in parallel, $p|q$ would reach an error state after an i-must-transition in [LNW07] since the potential output a of Q is not expected by P. In contrast, $p|q'$ has an i-must- and i-may-transition not allowed by $P|Q$, so that $p|q' \not\sqsubseteq_{\text{dMTS}} p|q$. This counterexample also holds for (strong) modal refinement as defined in [LNW07] and is particularly severe since all systems are deterministic and all must-transitions concern inputs only. The problem is that $p|q$ forbids input i.

In [LNW07], precongruence of parallel composition is not mentioned. Instead, a theorem relates the parallel composition of two IOMTSs to a different composition on two refining implementations, where an implementation in [LNW07] is an IOMTS in which may- and must-transitions coincide. This theorem is incorrect as is pointed out in [RBB+11] and repaired in the deterministic setting of that paper; the repair is again not a precongruence result, but still compares the results of two different composition operators. However, a natural solution to the precongruence problem can be adopted from the IA-framework [dH05] where inputs are always allowed implicitly. Consequently, if an input transition is specified, it will always be a must.

In the remainder, we thus define and study a new specification framework, called *Modal Interface Automata* (MIA), that takes the dMTS-setting for an alphabet consisting of input and output actions, requires input-determinism, and demands that every input-may-transition is also an input-must-transition. The advantage over IA is that outputs can be prescribed via output-must-transitions, which precludes trivial implementations like *Black-Hole* discussed in Sec. 2.

Definition 4.1 (Modal Interface Automaton)**.** A *Modal Interface Automaton* (MIA) is a tuple $Q = (Q, I, O, \longrightarrow, \dashrightarrow)$, where $(Q, I \cup O, \longrightarrow, \dashrightarrow)$ is a dMTS with disjoint alphabets I for inputs and O for outputs and where for all $i \in I$: (a) $q \xrightarrow{i} Q'$ and $q \xrightarrow{i} Q''$ implies $Q' = Q''$; (b) $q \overset{i}{\dashrightarrow} q'$ implies $\exists Q'.\, q \xrightarrow{i} Q'$ and $q' \in Q'$.

In the conference version of this article, we have considered truly disjunctive must-transitions only for outputs, so as to satisfy input determinism; this suffices for developing MIA-conjunction. However, for disjunction we have seen that such transitions are also needed for inputs. The above definition of MIA therefore permits one disjunctive must-transition for each input. This allows some choice on performing an input but, surprisingly, it is input-deterministic enough to support compositionality for parallel composition (cf. Thm. 4.14).

Definition 4.2 (MIA-Refinement)**.** Let P, Q be MIAs with common input and output alphabets. Relation $\mathcal{R} \subseteq P \times Q$ is an *(observational) MIA-refinement relation* if for all $(p,q) \in \mathcal{R}$:

(i): $q \xrightarrow{a} Q'$ implies $\exists P'.\, p \xrightarrow{a} P'$ and $\forall p' \in P'\, \exists q' \in Q'.\, (p',q') \in \mathcal{R}$,

(ii): $p \overset{\alpha}{\dashrightarrow} p'$ with $\alpha \in O \cup \{\tau\}$ implies $\exists q'.\, q \overset{\hat{\alpha}}{=\!=\!\Rightarrow} q'$ and $(p',q') \in \mathcal{R}$.

We write $p \sqsubseteq_{\mathrm{MIA}} q$ and say that p *MIA-refines* q if there exists an observational MIA-refinement relation $\mathcal{R}$ such that $(p,q) \in \mathcal{R}$. Moreover, we also write $p =_{\mathrm{MIA}} q$ in case $p \sqsubseteq_{\mathrm{MIA}} q$ and $q \sqsubseteq_{\mathrm{MIA}} p$ (which is an equivalence weaker than 'bisimulation').

One can easily check that $\sqsubseteq_{\mathrm{MIA}}$ is a preorder and the largest observational MIA-refinement relation. Its definition coincides with dMTS-refinement except that Cond. (ii) is restricted to outputs and the silent action τ. Thus, inputs are always allowed implicitly and, in effect, treated just like in IA-refinement. Due to the output-must-transitions in the MIA-setting, MIA-refinement can model, e.g., STG-bisimilarity [VW02] for systems without internal actions; this is a kind of alternating simulation refinement used for digital circuits.

4.1. **Conjunction on MIA.** Similar to conjunction on dMTS, we define conjunction on MIA by first constructing a conjunctive product and then eliminating all inconsistent states.

Definition 4.3 (Conjunctive Product on MIA)**.** Let $P = (P, I, O, \longrightarrow_P, \dashrightarrow_P)$ and $Q = (Q, I, O, \longrightarrow_Q, \dashrightarrow_Q)$ be MIAs with common input and output alphabets and disjoint state sets P and Q. The conjunctive product $P \& Q =_{\mathrm{df}} ((P \times Q) \cup P \cup Q, I, O, \longrightarrow, \dashrightarrow)$ inherits the transitions of P and Q and has additional transitions as follows, where $i \in I$, $o \in O$ and $\alpha \in O \cup \{\tau\}$:

(OMust1)	$(p,q) \xrightarrow{o} \{(p',q') \mid p' \in P',\, q \overset{o}{=\!=\!\Rightarrow}_Q q'\}$	if $p \xrightarrow{o}_P P'$ and $q \overset{o}{=\!=\!\Rightarrow}_Q$
(OMust2)	$(p,q) \xrightarrow{o} \{(p',q') \mid p \overset{o}{=\!=\!\Rightarrow}_P p',\, q' \in Q'\}$	if $p \overset{o}{=\!=\!\Rightarrow}_P$ and $q \xrightarrow{o}_Q Q'$
(IMust1)	$(p,q) \xrightarrow{i} P'$	if $p \xrightarrow{i}_P P'$ and $q \not\xrightarrow{i}_Q$
(IMust2)	$(p,q) \xrightarrow{i} Q'$	if $p \not\xrightarrow{i}_P$ and $q \xrightarrow{i}_Q Q'$
(IMust3)	$(p,q) \xrightarrow{i} P' \times Q'$	if $p \xrightarrow{i}_P P'$ and $q \xrightarrow{i}_Q Q'$
(May1)	$(p,q) \overset{\tau}{\dashrightarrow} (p',q)$	if $p \overset{\tau}{=\!=\!\Rightarrow}_P p'$
(May2)	$(p,q) \overset{\tau}{\dashrightarrow} (p,q')$	if $q \overset{\tau}{=\!=\!\Rightarrow}_Q q'$
(May3)	$(p,q) \overset{\alpha}{\dashrightarrow} (p',q')$	if $p \overset{\alpha}{=\!=\!\Rightarrow}_P p'$ and $q \overset{\alpha}{=\!=\!\Rightarrow}_Q q'$
(IMay1)	$(p,q) \overset{i}{\dashrightarrow} p'$	if $p \overset{i}{\dashrightarrow}_P p'$ and $q \not\overset{i}{\dashrightarrow}_Q$
(IMay2)	$(p,q) \overset{i}{\dashrightarrow} q'$	if $p \not\overset{i}{\dashrightarrow}_P$ and $q \overset{i}{\dashrightarrow}_Q q'$
(IMay3)	$(p,q) \overset{i}{\dashrightarrow} (p',q')$	if $p \overset{i}{\dashrightarrow}_P p'$ and $q \overset{i}{\dashrightarrow}_Q q'$

This product is defined analogously to IA-conjunction for inputs (plus the corresponding 'may' rules) and to the dMTS-product for outputs and τ. Thus, it combines the effects

shown in Fig. 1 (where all outputs are treated as may) and Fig. 5 (where all actions are outputs).

Definition 4.4 (Conjunction on MIA)**.** Given a conjunctive product $P\&Q$, the set $F \subseteq P \times Q$ of (logically) *inconsistent states* is defined as the least set satisfying the following rules:

(F1) $p \xrightarrow{o}_P$, $q \not\overset{o}{=\!=\!\!\rightarrow}_Q$, $o \in O$ implies $(p,q) \in F$
(F2) $p \not\overset{o}{=\!=\!\!\rightarrow}_P$, $q \xrightarrow{o}_Q$, $o \in O$ implies $(p,q) \in F$
(F3) $(p,q) \xrightarrow{a} R'$ and $R' \subseteq F$ implies $(p,q) \in F$

The conjunction $P \wedge Q$ of MIAs P, Q with common input and output alphabets is obtained by deleting all states $(p,q) \in F$ from $P\&Q$ as for dMTS in Def. 3.4. We write $p \wedge q$ for state (p,q) of $P \wedge Q$; all such states are defined – and consistent – by construction.

The conjunction $P \wedge Q$ is a MIA and is thus well-defined. This can be seen by a similar argument as we have used above in the context of dMTS-conjunction, while input-determinism can be established by an argument similar to that in the IA-setting. Note that, in contrast to the dMTS-situation, Rules (F1) and (F2) only apply to outputs. Fig. 5 is also an example for conjunction in the MIA-setting if all actions are read as outputs.

To reason about inconsistency we use a notion of witness again. This may be defined analogously to the witness notion for dMTS but replacing $a \in A$ in Def. 3.6(W1) and (W2) by $a \in O$. We then obtain the analogous lemma to Lemma 3.7, which is needed in the proof of the analogue theorem to Thm. 3.5:

Definition 4.5 (MIA-Witness)**.** A *MIA-witness* W of $P\&Q$ is a subset of $(P \times Q) \cup P \cup Q$ such that the following conditions hold for all $(p,q) \in W$:

(W1) $p \xrightarrow{o}_P$ with $o \in O$ implies $q \overset{o}{=\!=\!\!\rightarrow}_Q$
(W2) $q \xrightarrow{o}_Q$ with $o \in O$ implies $p \overset{o}{=\!=\!\!\rightarrow}_P$
(W3) $(p,q) \xrightarrow{a} R'$ implies $R' \cap W \neq \emptyset$

Lemma 4.6. *Let $P\&Q$ be a conjunctive product of MIAs. Then, for any MIA-witness W of $P\&Q$, we have (i) $F \cap W = \emptyset$. Moreover, (ii) the set $W =_{df} \{(p,q) \in P \times Q \mid \exists$ MIA R and $r \in R.\, r \sqsubseteq_{MIA} p$ and $r \sqsubseteq_{MIA} q\} \cup P \cup Q$ is a MIA-witness of $P\&Q$.*

Proof. Since Part (i) is again obvious, we directly proceed to proving Part (ii), for which it suffices to consider the elements of $\{(p,q) \in P \times Q \mid \exists r \in R.\, r \sqsubseteq_{\text{MIA}} p \text{ and } r \sqsubseteq_{\text{MIA}} q\}$; thus, let $(p,q) \in W$ due to MIA R and $r \in R$:

(W1): $p \xrightarrow{o}_P P'$ implies $r \xrightarrow{o}_R R'$ by $r \sqsubseteq_{\text{MIA}} p$. Choose some $r' \in R'$. Then, $r \overset{o}{\dashrightarrow}_R r'$ by syntactic consistency, and $q \overset{o}{=\!=\!\!\rightarrow}_Q$ by $r \sqsubseteq_{\text{MIA}} q$.

(W2): Analogous to (W1).

(W3): Assume $(p,q) \xrightarrow{a}$. According to the operational rules for conjunction, we distinguish the following cases:

(OMust1): Then, $(p,q) \xrightarrow{a} S'$ for $a \in O$, i.e., $p \xrightarrow{a}_P P'$ and $S' = \{(p',q') \mid p' \in P',\, q \overset{a}{=\!=\!\!\rightarrow}_Q q'\}$. By $r \sqsubseteq_{\text{MIA}} p$ we obtain some $R' \subseteq R$ such that $r \xrightarrow{a}_R R'$ and $\forall r' \in R'$ $\exists p' \in P'.\, r' \sqsubseteq_{\text{MIA}} p'$. Choose $r' \in R'$ and the respective $p' \in P'$; now, $r \overset{a}{\dashrightarrow}_R r'$ due to syntactic consistency, and $q \overset{a}{=\!=\!\!\rightarrow}_Q q'$ with $r' \sqsubseteq_{\text{MIA}} q'$ for some q' by $r \sqsubseteq_{\text{MIA}} q$. Thus, we have $p' \in P'$ and q' such that $(p',q') \in W \cap S'$ due to r'. Case (OMust2) is analogous.

(IMust1): Then, $(p,q) \xrightarrow{a} P'$ for $a \in I$, and we are done. Case (IMust2) is analogous.
(IMust3): Then, $(p,q) \xrightarrow{a} P' \times Q'$ for $a \in I$ due to $p \xrightarrow{a}_P P'$ and $q \xrightarrow{a}_Q Q'$. By $r \sqsubseteq_{\mathrm{MIA}} p$, $r \sqsubseteq_{\mathrm{MIA}} q$ and input-determinism, we have some R' and $r' \in R'$ such that $r \xrightarrow{a}_R R'$, $\exists\, p' \in P'.\, r' \sqsubseteq_{\mathrm{MIA}} p'$ and $\exists\, q' \in Q'.\, r' \sqsubseteq_{\mathrm{MIA}} q'$. Thus, $(p',q') \in W$ due to r'. $\square$

We can now state and prove the desired largest-lower-bound theorem, from which compositionality of $\sqsubseteq_{\mathrm{MIA}}$ wrt. $\wedge$ follows in analogy to the IA- and dMTS-settings:

Theorem 4.7 ($\wedge$ is And)**.** *Let P, Q be MIAs. We have* (i) *($\exists$ MIA R and $r \in R.\, r \sqsubseteq_{MIA} p$ and $r \sqsubseteq_{MIA} q$) if and only if $p \wedge q$ is defined. Further, in case $p \wedge q$ is defined and for any MIA R and $r \in R$:* (ii) *$r \sqsubseteq_{MIA} p$ and $r \sqsubseteq_{MIA} q$ if and only if $r \sqsubseteq_{MIA} p \wedge q$.*

Proof. (i)"$\Longrightarrow$": This follows directly from Lemma 4.6 above.

(ii)"$\Longleftarrow$": For a MIA R we show that $\mathcal{R} =_{\mathrm{df}} \{(r,p) \in R \times P \mid \exists q \in Q.\, r \sqsubseteq_{\mathrm{MIA}} p \wedge q\} \cup \sqsubseteq_{\mathrm{MIA}}$ is a MIA-refinement relation, by checking the two conditions of Def. 4.2 for some $(r,p) \in \mathcal{R}$ due to q:

- Let $p \xrightarrow{a}_P P'$ and consider the following cases depending on whether action a is an input or an output:
  - $a \in O$: Then, $q \overset{a}{=\!=\!\!\twoheadrightarrow}_Q$ since, otherwise, $p \wedge q$ would not be defined due to (F1). Thus, by Rule (OMust1), $p \wedge q \xrightarrow{a} \{p' \wedge q' \mid p' \in P',\, q \overset{a}{=\!=\!\!\twoheadrightarrow}_Q q',\, p' \wedge q' \text{ defined}\}$. By $r \sqsubseteq_{\mathrm{MIA}} p \wedge q$, we get some $R' \subseteq R$ such that $r \xrightarrow{a}_R R'$ and $\forall r' \in R'\ \exists p' \wedge q'.\ p' \in P',\, q \overset{a}{=\!=\!\!\twoheadrightarrow}_Q q'$ and $r' \sqsubseteq_{\mathrm{MIA}} p' \wedge q'$. Hence, $\forall r' \in R' \exists p' \in P'.\ (r',p') \in \mathcal{R}$.
  - $a \in I$: This can lead to a transition of $p \wedge q$ in two ways:
    (IMust1): $q \overset{a}{\not\longrightarrow}_Q$, whence $p \wedge q \xrightarrow{a} P'$. By $r \sqsubseteq_{\mathrm{MIA}} p \wedge q$, there is some R' such that $r \xrightarrow{a}_R R'$ and $\forall r' \in R' \exists p' \in P'.\, r' \sqsubseteq_{\mathrm{MIA}} p'$.
    (IMust3): $q \xrightarrow{a}_Q Q'$, whence $p \wedge q \xrightarrow{a} (P' \times Q') \setminus F$. By $r \sqsubseteq_{\mathrm{MIA}} p \wedge q$, there is some R' such that $r \xrightarrow{a}_R R'$ and $\forall r' \in R' \exists p' \wedge q' \in P' \times Q'.\ r' \sqsubseteq_{\mathrm{MIA}} p' \wedge q'$ and, thus, $(r',p') \in \mathcal{R}$ due to q'.
- $r \overset{\alpha}{\dashrightarrow}_R r'$ with $\alpha \in O \cup \{\tau\}$ implies $\exists p' \wedge q'.\, p \wedge q \overset{\hat{\alpha}}{=\!=\!\!\twoheadrightarrow} p' \wedge q'$ and $r' \sqsubseteq_{\mathrm{MIA}} p' \wedge q'$. The contribution of p in this weak transition sequence gives $p \overset{\hat{\alpha}}{=\!=\!\!\twoheadrightarrow}_P p'$, and we have $(r',p') \in \mathcal{R}$ due to q'.

(i)"$\Longleftarrow$": This follows from (ii)"$\Longleftarrow$" by choosing $R = P \wedge Q$ and $r = p \wedge q$.

(ii)"$\Longrightarrow$": Let R be a MIA R. We show that the relation $\mathcal{R} =_{\mathrm{df}} \{(r, p \wedge q) \mid r \in R,\, r \sqsubseteq_{\mathrm{MIA}} p \text{ and } r \sqsubseteq_{\mathrm{MIA}} q\} \cup \sqsubseteq_{\mathrm{MIA}}$ is a MIA-refinement relation. Due to Part (i), $p \wedge q$ is defined whenever $r \sqsubseteq_{\mathrm{MIA}} p$ and $r \sqsubseteq_{\mathrm{MIA}} q$. We now verify the conditions of Def. 4.2 for $(r, p \wedge q) \in \mathcal{R}$:

- Let $p \wedge q \xrightarrow{a}$ and distinguish the following cases by our operational rules:
  - $p \wedge q \xrightarrow{a} S'$ with $a \in O$: By Rule (OMust1) this is w.l.o.g. due to $p \xrightarrow{a}_P P'$ and $S' = \{p' \wedge q' \mid p' \in P',\, q \overset{a}{=\!=\!\!\twoheadrightarrow}_Q q',\, p' \wedge q' \text{ defined}\}$. By $r \sqsubseteq_{\mathrm{MIA}} p$, we have some $R' \subseteq R$ such that $r \xrightarrow{a}_R R'$ and $\forall r' \in R'\ \exists p' \in P'.\, r' \sqsubseteq_{\mathrm{MIA}} p'$. Consider some arbitrary $r' \in R'$ and the respective $p' \in P'$. Then, we have $r \overset{a}{\dashrightarrow}_R r'$ by syntactic consistency and, due to $r \sqsubseteq_{\mathrm{MIA}} q$, some q' with $q \overset{a}{=\!=\!\!\twoheadrightarrow}_Q q'$ and $r' \sqsubseteq_{\mathrm{MIA}} q'$. Thus, $p' \wedge q' \in S'$ and $(r', p' \wedge q') \in \mathcal{R}$.

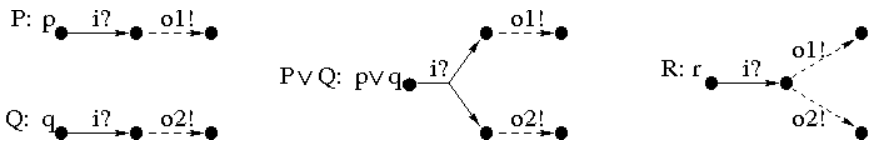

Figure 11: MIA-disjunction is more intuitive than IA-disjunction.

- $p \wedge q \xrightarrow{a} P'$ with $a \in I$: This is w.l.o.g. due to Rule (IMust1): $p \xrightarrow{a}_P P'$ and $q \not\xrightarrow{a}_Q$. By $r \sqsubseteq_{\text{MIA}} p$, we have some R' such that $r \xrightarrow{a}_R R'$ and $\forall r' \in R'\, \exists p' \in P'.\, r' \sqsubseteq_{\text{MIA}} p'$, whence $(r', p') \in \mathcal{R}$.
- $p \wedge q \xrightarrow{a} (P' \times Q') \setminus F$ with $a \in I$: This is due to Rule (IMust3), i.e., $p \xrightarrow{a}_P P'$ and $q \xrightarrow{a}_Q Q'$. By $r \sqsubseteq_{\text{MIA}} p$ and $r \sqsubseteq_{\text{MIA}} q$, we get a unique $r \xrightarrow{a}_R R'$ (by input-determinism) such that $\forall r' \in R'\, \exists p' \in P',\, q' \in Q'.\, r' \sqsubseteq_{\text{MIA}} p'$ and $r' \sqsubseteq_{\text{MIA}} q'$; thus, $(r', p' \wedge q') \in \mathcal{R}$.

- Let $r \overset{\alpha}{\dashrightarrow}_R r'$ with $\alpha \in O \cup \{\tau\}$ and consider $p \overset{\hat{\alpha}}{=\!=\!\twoheadrightarrow}_P p'$ and $q \overset{\hat{\alpha}}{=\!=\!\twoheadrightarrow}_Q q'$ satisfying $r' \sqsubseteq_{\text{MIA}} p'$ and $r' \sqsubseteq_{\text{MIA}} q'$. Thus, $(r', p' \wedge q') \in \mathcal{R}$. Further, if $\alpha \neq \tau$, we have $p \wedge q \overset{\alpha}{\dashrightarrow} p' \wedge q'$ by Rule (May3). Otherwise, either $p \overset{\tau}{=\!=\!\twoheadrightarrow}_P p'$ and $q \overset{\tau}{=\!=\!\twoheadrightarrow}_Q q'$ and we are done by Rule (May3), or w.l.o.g. $p \overset{\tau}{=\!=\!\twoheadrightarrow}_P p'$ and $q = q'$ and we are done by Rule (May1), or $p = p'$ and $q = q'$. □

In analogy to Corollary 3.8 we obtain:

Corollary 4.8. *MIA-refinement is compositional wrt. conjunction.* □

4.2. **Disjunction on MIA.** The disjunction of two MIAs P and Q can be defined in the same way as for dMTS, except for the special treatment of inputs in the may-rules which guarantees that $P \vee Q$ is a MIA and, especially, that Def. 4.1(b) is satisfied:

Definition 4.9 (Disjunction on MIA)**.** Let $P = (P, I, O, \longrightarrow_P, \dashrightarrow_P)$, $Q = (Q, I, O, \longrightarrow_Q, \dashrightarrow_Q)$ be MIAs with common input and output alphabets and disjoint state sets P and Q. The disjunction $P \vee Q$ is defined by $(\{p \vee q \mid p \in P, q \in Q\} \cup P \cup Q, I, O, \longrightarrow, \dashrightarrow)$, where $\longrightarrow$ and $\dashrightarrow$ are the least sets satisfying $\longrightarrow_P \subseteq \longrightarrow$, $\dashrightarrow_P \subseteq \dashrightarrow$, $\longrightarrow_Q \subseteq \longrightarrow$, $\dashrightarrow_Q \subseteq \dashrightarrow$ and the following operational rules:

(Must) $p \vee q \xrightarrow{a} P' \cup Q'$ if $p \xrightarrow{a}_P P'$ and $q \xrightarrow{a}_Q Q'$

(May1) $p \vee q \overset{\alpha}{\dashrightarrow} p'$ if $p \overset{\alpha}{\dashrightarrow}_P p'$ and, in case $\alpha \in I$, also $q \overset{\alpha}{\dashrightarrow}_Q$

(May2) $p \vee q \overset{\alpha}{\dashrightarrow} q'$ if $q \overset{\alpha}{\dashrightarrow}_Q q'$ and, in case $\alpha \in I$, also $p \overset{\alpha}{\dashrightarrow}_P$

It is easy to see that this definition is well-defined, i.e., the resulting disjunctions are indeed MIAs, and we additionally have:

Theorem 4.10 ($\vee$ is Or)**.** *Let P, Q and R be MIAs with states p, q and r, resp. Then, $p \vee q \sqsubseteq_{MIA} r$ if and only if $p \sqsubseteq_{MIA} r$ and $q \sqsubseteq_{MIA} r$.* □

The theorem's proof is as for dMTS (cf. Thm. 3.10) but, in the (ii)-cases, only $\alpha \in O \cup \{\tau\}$ has to be considered. Analogously to dMTS we obtain the following corollary to Thm. 4.10:

Corollary 4.11. *MIA-refinement is compositional wrt. disjunction.* □

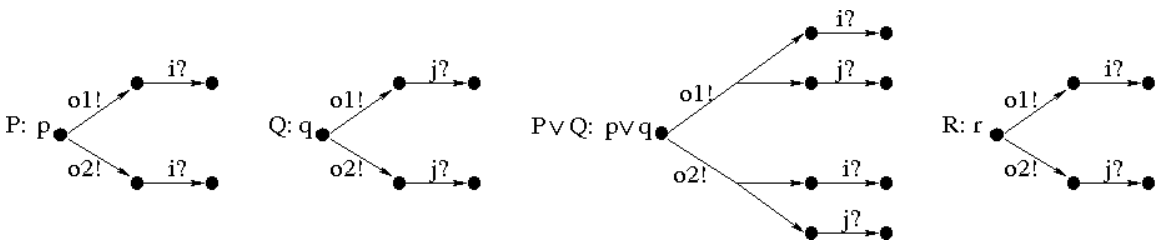

Figure 12: MIA-disjunction is an inclusive-or.

To conclude this section we argue that MIA-disjunction is more intuitive than IA-disjunction. The example in Fig. 11 shows MIAs P, Q, $P \vee Q$ as well as a MIA R, where state r corresponds to the IA-disjunction of states p and q when we understand P and Q as IAs. As expected (cf. p. 7), $p \vee q$ is a refinement of r, but not vice versa. MIA-disjunction can now be considered to be more intuitive since the first transition in the disjunction decides which disjunct has to be satisfied afterward, in contrast to IA-disjunction.

Moreover, Fig. 12 shows that MIA-disjunction is an inclusive-or: an implementation of $p \vee q$ can have an $o1$-transition followed by i and another $o1$-transition followed by j; interestingly, $r \sqsubseteq_{\text{MIA}} p \vee q$ satisfies 'half' of p and 'half' of q. In general, for each action $a \in A$ separately, a refinement of some disjunction has to satisfy at least all initial a-must-transitions of one of its disjuncts.

4.3. **Parallel Composition on MIA.** In analogy to the IA-setting [dH05] we provide a parallel operator on MIA. Here, error states are identified, and all states are removed from which reaching an error state is unavoidable in some implementation, as is done for IOMTS in [LNW07].

Definition 4.12 (Parallel Product on MIA)**.** MIAs P_1 and P_2 are *composable* if $A_1 \cap A_2 = (I_1 \cap O_2) \cup (O_1 \cap I_2)$, as in IA. For such MIAs we define the *product* $P_1 \otimes P_2 = (P_1 \times P_2, I, O, \longrightarrow, \dashrightarrow)$, where $I = (I_1 \cup I_2) \setminus (O_1 \cup O_2)$ and $O = (O_1 \cup O_2) \setminus (I_1 \cup I_2)$ and where $\longrightarrow$ and $\dashrightarrow$ are defined as follows:

(Must1) $(p_1, p_2) \xrightarrow{a} P_1' \times \{p_2\}$ if $p_1 \xrightarrow{a} P_1'$ and $a \notin A_2$

(Must2) $(p_1, p_2) \xrightarrow{a} \{p_1\} \times P_2'$ if $p_2 \xrightarrow{a} P_2'$ and $a \notin A_1$

(May1) $(p_1, p_2) \overset{\alpha}{\dashrightarrow} (p_1', p_2)$ if $p_1 \overset{\alpha}{\dashrightarrow} p_1'$ and $\alpha \notin A_2$

(May2) $(p_1, p_2) \overset{\alpha}{\dashrightarrow} (p_1, p_2')$ if $p_2 \overset{\alpha}{\dashrightarrow} p_2'$ and $\alpha \notin A_1$

(May3) $(p_1, p_2) \overset{\tau}{\dashrightarrow} (p_1', p_2')$ if $p_1 \overset{a}{\dashrightarrow} p_1'$ and $p_2 \overset{a}{\dashrightarrow} p_2'$ for some a.

Recall that there are no τ-must-transitions since they are irrelevant for refinement.

Definition 4.13 (Parallel Composition on MIA)**.** Given a parallel product $P_1 \otimes P_2$, a state (p_1, p_2) is an *error state* if there is some $a \in A_1 \cap A_2$ such that (a) $a \in O_1$, $p_1 \overset{a}{\dashrightarrow}$ and $p_2 \overset{a}{\not\longrightarrow}$, or (b) $a \in O_2$, $p_2 \overset{a}{\dashrightarrow}$ and $p_1 \overset{a}{\not\longrightarrow}$.

Again we define the set $E \subseteq P_1 \times P_2$ of *incompatible* states as the least set such that $(p_1, p_2) \in E$ if (i) (p_1, p_2) is an error state or (ii) $(p_1, p_2) \overset{\alpha}{\dashrightarrow} (p_1', p_2')$ for some $\alpha \in O \cup \{\tau\}$ and $(p_1', p_2') \in E$.

The *parallel composition* $P_1|P_2$ of P_1 and P_2 is now obtained from $P_1 \otimes P_2$ by *pruning*, namely removing all states in E and every transition that involves such states as its source, its target or one of its targets; all may-transitions underlying a removed must-transition are deleted, too. If $(p_1, p_2) \in P_1|P_2$, we write $p_1|p_2$ and call p_1 and p_2 *compatible*.

Parallel products and parallel compositions are well-defined MIAs. Syntactic consistency is preserved, as is input-determinism since input-transitions are directly inherited from one of the *composable* systems. In particular, Cond. (b) in Def. 4.1 holds due to the additional clause regarding the deletion of may-transitions. In addition, targets of disjunctive must-transitions are never empty since all must-transitions that remain after pruning are taken from the product without modification.

As an example why pruning is needed, consider Fig. 3 again and read the τ-transitions as may-transitions and all other transitions as must-transitions. Further observe that pruning is different from removing inconsistent states in conjunction. For truly disjunctive transitions $(p_1, p_2) \xrightarrow{a} P'$ of the product $P_1 \otimes P_2$, the state (p_1, p_2) is removed already if $P' \cap E \neq \emptyset$, i.e., there exists some $(p_1', p_2') \in P' \cap E$, and not only if $P' \subseteq E$. This is clear for $a \in O$ since $(p_1, p_2) \overset{a}{\dashrightarrow} (p_1', p_2')$ by syntactic consistency and, therefore, (p_1, p_2) is deleted itself by Cond. (ii) above. Note that Cond. (ii) corresponds directly to the IA-case since output-transitions there correspond to may-transitions here (see Sec. 3.4). For $a \in I$, reaching the error state can only be prevented if the environment does not provide a; intuitively, this is because P' has w.l.o.g. the form $P_1' \times \{p_2\}$ in the product of P_1 and P_2 (i.e., $p_2' = p_2$). The implementor of P_1 might choose to implement $p_1 \xrightarrow{a} p_1'$ such that – when P_1's implementation is composed with P_2's – the error state is reached. To express the requirement on the environment not to exhibit a, must-transition $(p_1, p_2) \xrightarrow{a} P'$ and all underlying may-transitions have to be deleted.

Theorem 4.14 (Compositionality of MIA-Parallel Composition)**.** *Let P_1, P_2 and Q_1 be MIAs with $p_1 \in P_1$, $p_2 \in P_2$, $q_1 \in Q_1$ and $p_1 \sqsubseteq_{MIA} q_1$. Assume that Q_1 and P_2 are composable; then:*

(a): *P_1 and P_2 are composable.*
(b): *If q_1 and p_2 are compatible, then so are p_1, p_2 and $p_1|p_2 \sqsubseteq_{MIA} q_1|p_2$.*

Proof. Part (a) follows immediately since MIA Q_1 has the same input and output alphabets as MIA P_1, due to $p_1 \sqsubseteq_{\text{MIA}} q_1$. Regarding Part (b), the first claim is implied by the following auxiliary result:

> Let E_P be the E-set of $P_1 \otimes P_2$ and E_Q be the one of $Q_1 \otimes P_2$. Then, $(p_1, p_2) \in E_P$ and $p_1 \sqsubseteq_{\text{MIA}} q_1$ together imply $(q_1, p_2) \in E_Q$.

The proof of this result is by induction on the length of a path from (p_1, p_2) to an error state of $P_1 \otimes P_2$:

(Base): Let (p_1, p_2) be an error state.

- Let $p_1 \overset{a}{\dashrightarrow}_{P_1}$ with $a \in O_1 \cap I_2$ and $p_2 \overset{a}{\not\longrightarrow}_{P_2}$. Then, for some q_1', we have $q_1 \overset{\varepsilon}{=\!=\!\!\blacktriangleright}_{Q_1} q_1' \overset{a}{\dashrightarrow}_{Q_1}$ by $p_1 \sqsubseteq_{\text{MIA}} q_1$; therefore, $(q_1, p_2) \overset{\varepsilon}{=\!=\!\!\blacktriangleright} (q_1', p_2) \in E_Q$ and $(q_1, p_2) \in E_Q$, too.
- Let $p_2 \overset{a}{\dashrightarrow}_{P_2}$ with $a \in O_2 \cap I_1$ and $p_1 \overset{a}{\not\longrightarrow}_{P_1}$. If $q_1 \xrightarrow{a}_{Q_1}$, we have a contradiction to $p_1 \sqsubseteq_{\text{MIA}} q_1$; otherwise, (q_1, p_2) is an error state.

(Step): For a shortest path from (p_1, p_2) to an error state, consider the first transition $(p_1, p_2) \overset{\alpha}{\dashrightarrow} (p_1', p_2') \in E_P$ with $\alpha \in O \cup \{\tau\}$. The transition is due to Rule (May1),

(May2) or (May3). In all cases we show $p_1' \sqsubseteq_{\text{MIA}} q_1'$, which implies $(q_1', p_2') \in E_Q$ by induction hypothesis.

(May1): $p_1 \xrightarrow{\alpha}_{P_1} p_1'$, $p_2 = p_2'$, $\alpha \notin A_2$, and $\alpha \in O_1 \cup \{\tau\}$ by $\alpha \in O \cup \{\tau\}$. Hence, there is some q_1' such that $q_1 \stackrel{\hat{\alpha}}{\Longrightarrow}_{Q_1} q_1'$ and $p_1' \sqsubseteq_{\text{MIA}} q_1'$, due to $p_1 \sqsubseteq_{\text{MIA}} q_1$, and $(q_1, p_2) \stackrel{\hat{\alpha}}{\Longrightarrow} (q_1', p_2)$ by applications of Rule (May1). By induction hypothesis, $(q_1', p_2) \in E_Q$ and, thus, $(q_1, p_2) \in E_Q$.

(May2): $p_1 = p_1'$, $p_2 \xrightarrow{\alpha}_{P_2} p_2'$ and $\alpha \notin A_1$. Now, since P_1 and Q_1 have the same alphabets by $p_1 \sqsubseteq_{\text{MIA}} q_1$, we can apply Rule (May2) again and obtain $(q_1, p_2) \xrightarrow{\alpha} (q_1, p_2')$, so that $(q_1, p_2') \in E_Q$ by induction hypothesis. Hence, $(q_1, p_2) \in E_Q$, too.

(May3): $\alpha = \tau$.

- $p_1 \xrightarrow{a}_{P_1} p_1'$ with $a \in O_1$, and $p_2 \xrightarrow{a}_{P_2} p_2'$ with $a \in I_2$. By $p_1 \sqsubseteq_{\text{MIA}} q_1$, we have $q_1 \stackrel{\varepsilon}{\Longrightarrow}_{Q_1} q_1'' \xrightarrow{a}_{Q_1} q_1'$ for some q_1', q_1'' with $p_1' \sqsubseteq_{\text{MIA}} q_1'$. Hence, $(q_1, p_2) \stackrel{\varepsilon}{\Longrightarrow} (q_1'', p_2) \xrightarrow{\tau} (q_1', p_2')$ via Rules (May1) and (May3). By induction hypothesis, $(q_1', p_2') \in E_Q$ and, thus, $(q_1, p_2) \in E_Q$, too.
- $p_1 \xrightarrow{a}_{P_1} p_1'$ with $a \in I_1$, and $p_2 \xrightarrow{a}_{P_2} p_2'$ with $a \in O_2$. If $q_1 \not\xrightarrow{a}_{Q_1}$, then $q_1 \not\xrightarrow{a}_{Q_1}$ by syntactic consistency and (q_1, p_2) is thus an error state. If $q_1 \xrightarrow{a}_{Q_1}$, then there exist unique $p_1 \xrightarrow{a}_{P_1} P'$ and $q_1 \xrightarrow{a}_{Q_1} Q'$. We have $p_1' \in P'$ by Def. 4.1(b) and $\exists q_1' \in Q'.\, p_1' \sqsubseteq_{\text{MIA}} q_1'$ since $p_1 \sqsubseteq_{\text{MIA}} q_1$. Hence, $q_1 \xrightarrow{a}_{Q_1} q_1'$ by syntactic consistency and $(q_1, p_2) \xrightarrow{\tau} (q_1', p_2')$ due to Rule (May3). By induction hypothesis, $(q_1', p_2') \in E_Q$ and, therefore, $(q_1, p_2) \in E_Q$.

This completes the proof of the auxiliary result. We can now prove that

$$\mathcal{R} =_{\text{df}} \{(p_1|p_2, q_1|p_2) \mid p_1 \sqsubseteq_{\text{MIA}} q_1,\ p_1, p_2 \text{ as well as } q_1, p_2 \text{ compatible}\}$$

is a MIA-refinement relation, for which we let $(p_1|p_2, q_1|p_2) \in \mathcal{R}$ and check the conditions of Def. 4.2:

(i): Let $q_1|p_2 \xrightarrow{a} Q'$ with $Q' \cap E_Q = \emptyset$ due to either Rule (Must1) or (Must2).

(Must1): $q_1 \xrightarrow{a}_{Q_1} Q_1'$ and $Q' = Q_1' \times \{p_2\}$. Then, by $p_1 \sqsubseteq_{\text{MIA}} q_1$, there is some $P_1' \subseteq P_1$ such that $p_1 \xrightarrow{a}_{P_1} P_1'$ and $\forall p_1' \in P_1'\, \exists q_1' \in Q_1'.\, p_1' \sqsubseteq_{\text{MIA}} q_1'$. Now, $(p_1, p_2) \xrightarrow{a} P_1' \times \{p_2\}$ by Rule (Must1) and since $a \notin A_2$. For $p_1' \in P_1'$ we have a suitable $q_1' \in Q_1'$, and $(p_1', p_2) \notin E_P$ since $(q_1', p_2) \notin E_Q$ and due to the auxiliary result above. Thus, for the arbitrary $p_1'|p_2$, we also have $(p_1'|p_2, q_1'|p_2) \in \mathcal{R}$.

(Must2): $p_2 \xrightarrow{a}_{P_2} P_2'$ and $Q' = \{q_1\} \times P_2'$. Then, $(p_1, p_2) \xrightarrow{a} P' = \{p_1\} \times P_2'$ by Rule (Must2) and as P_1, Q_1 have the same alphabets by $p_1 \sqsubseteq_{\text{MIA}} q_1$. For $(p_1, p_2') \in P'$, we get $(p_1, p_2') \notin E_P$ since $(q_1, p_2') \notin E_Q$ and due to the auxiliary result above. Thus, $p_1|p_2 \xrightarrow{a} P'$ and, for $p_1|p_2' \in P'$, we have $q_1|p_2' \in Q'$ with $(p_1|p_2', q_1|p_2') \in \mathcal{R}$.

(ii): Let $p_1|p_2 \xrightarrow{\alpha} p_1'|p_2' \notin E_P$ with $\alpha \in O \cup \{\tau\}$. The transition arises from one of the Rules (May1), (May2) or (May3):

(May1): $p_2' = p_2$ and $p_1 \xrightarrow{\alpha}_{P_1} p_1'$. By $p_1 \sqsubseteq_{\text{MIA}} q_1$, we have $q_1 \stackrel{\hat{\alpha}}{\Longrightarrow}_{Q_1} q_1'$ for some q_1' such that $p_1' \sqsubseteq_{\text{MIA}} q_1'$. Hence, $(q_1, p_2) \stackrel{\hat{\alpha}}{\Longrightarrow} (q_1', p_2)$ by repeated application of Rule (May1) and since $\omega \notin A_2$. If any state on this transition sequence

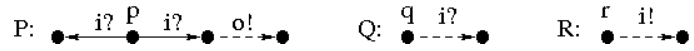

Figure 13: Example illustrating the need of input-determinism for MIA.

were in E_Q, then also $(q_1, p_2) \in E_Q$ which contradicts $(p_1|p_2, q_1|p_2) \in \mathcal{R}$. Thus, $q_1|p_2 \stackrel{\hat{\alpha}}{=\!=\!\Rightarrow} q_1'|p_2$ with $(p_1'|p_2, q_1'|p_2) \in \mathcal{R}$.

(May2): $p_1' = p_1$ and $p_2 \stackrel{\alpha}{\dashrightarrow}_{P_2} p_2'$. Then, $(q_1, p_2) \stackrel{\alpha}{\dashrightarrow} (q_1, p_2')$ by Rule (May2) and since P_1 and Q_1 have the same alphabets due to $p_1 \sqsubseteq_{\text{MIA}} q_1$. If the latter state (q_1, p_2') were in E_Q, then also the former state (q_1, p_2). Therefore, we have $q_1|p_2 \stackrel{\alpha}{\dashrightarrow} q_1|p_2'$ and, moreover, $(p_1|p_2', q_1|p_2') \in \mathcal{R}$.

(May3): $\alpha = \tau$, $p_1 \stackrel{a}{\dashrightarrow}_{P_1} p_1'$ and $p_2 \stackrel{a}{\dashrightarrow}_{P_2} p_2'$ for some a.

- $a \in O_1 \cap I_2$: Then, $q_1 \stackrel{\varepsilon}{=\!=\!\Rightarrow}_{Q_1} q_1'' \stackrel{a}{\dashrightarrow}_{Q_1} q_1'$ for q_1', q_1'' with $p_1' \sqsubseteq_{\text{MIA}} q_1'$, due to $p_1 \sqsubseteq_{\text{MIA}} q_1$. Now, $(q_1, p_2) \stackrel{\varepsilon}{=\!=\!\Rightarrow} (q_1'', p_2) \stackrel{\tau}{\dashrightarrow} (q_1', p_2')$ by Rules (May1), (May3). As in Case (May1) above, $q_1|p_2 \stackrel{\varepsilon}{=\!=\!\Rightarrow} q_1'|p_2'$ and $(p_1'|p_2', q_1'|p_2') \in \mathcal{R}$.
- $a \in I_1 \cap O_2$: If $q_1 \not\stackrel{a}{\dashrightarrow}_{Q_1}$, then (q_1, p_2) would be an error state, which is a contradiction. Therefore, $q_1 \stackrel{a}{\dashrightarrow}_{Q_1}$ and, by Def. 4.1(b), there exist unique $p_1 \stackrel{a}{\longrightarrow}_{P_1} P'$ and $q_1 \stackrel{a}{\longrightarrow}_{Q_1} Q'$ by input-determinism. We have $p_1' \in P'$ and $\exists q_1' \in Q'.\, p_1' \sqsubseteq_{\text{MIA}} q_1'$ since $p_1 \sqsubseteq_{\text{MIA}} q_1$. Thus, $(q_1, p_2) \stackrel{\tau}{\dashrightarrow} (q_1', p_2')$ by Rule (May3) and syntactic consistency, and $(q_1', p_2') \notin E_Q$ by the same reasoning as above. Hence, $q_1|p_2 \stackrel{\tau}{\dashrightarrow} q_1'|p_2'$ with $(p_1'|p_2', q_1'|p_2') \in \mathcal{R}$. □

This precongruence property of MIA-refinement would not hold if we would do away with input-determinism in MIA. To see this, consider the example of Fig. 13 for which $p \sqsubseteq_{\text{MIA}} q$; however, $p|r \sqsubseteq_{\text{MIA}} q|r$ does not hold since q and r are compatible while p and r are not. An analogue reasoning applies to IA, although we do not know of a reference in the IA literature where this has been observed.

4.4. **Embedding of IA into MIA.** To conclude, we provide an embedding of IA into MIA in the line of [LNW07]:

Definition 4.15 (IA-Embedding)**.** Let P be an IA. The embedding $[P]_{\text{MIA}}$ of P into MIA is defined as the MIA $(P, I, O, \longrightarrow, \dashrightarrow)$, where (i) $p \stackrel{i}{\longrightarrow} p'$ if $p \stackrel{i}{\longrightarrow}_P p'$ and $i \in I$, and (ii) $p \stackrel{\alpha}{\dashrightarrow} p'$ if $p \stackrel{\alpha}{\longrightarrow}_P p'$ and $\alpha \in I \cup O \cup \{\tau\}$.

In the remainder of this section we simply write $[p]$ for $p \in [P]_{\text{MIA}}$. This embedding is much simpler than the one of [LNW07] since MIA more closely resembles IA than IOMTS does. In particular, the following theorem is obvious:

Theorem 4.16 (IA-Embedding Respects Refinement)**.** *For IAs P, Q with $p \in P$, $q \in Q$: $p \sqsubseteq_{IA} q$ if and only if $[p] \sqsubseteq_{MIA} [q]$.*

Our embedding respects operators $\wedge$ and $|$, unlike the one in [LNW07]:

Theorem 4.17 (IA-Embedding is a Homomorphism)**.** *For IAs P, Q with $p \in P$, $q \in Q$:*
(a): $[p] \wedge [q] =_{MIA} [p \wedge q]$*;*
(b): $[p] \,|\, [q] =_{MIA} [p|q]$*.*

Proof. Part (b) follows directly from the definitions of parallel composition on IA and MIA, whereas Part (a)"$\sqsupseteq_{\text{MIA}}$" is an immediate consequence of Thms. 4.7 and 4.16 by general order theory. We are thus left with proving Part (a)"$\sqsubseteq_{\text{MIA}}$".

Both sides only differ in additional transitions $\overset{\alpha}{\dashrightarrow}$ with $\alpha \in O \cup \{\tau\}$ in $[P]_{\text{MIA}} \wedge [Q]_{\text{MIA}}$, where on the other side $\overset{\varepsilon}{=\!=\!\twoheadrightarrow}\ \overset{\alpha}{\dashrightarrow}$. Formally, we define the relation $\mathcal{R} =_{\text{df}} \{([p] \wedge [q], [p \wedge q]) \mid p \in P,\, q \in Q\} \cup \text{id}_P \cup \text{id}_Q$ and argue that $\mathcal{R}$ is a MIA-refinement relation:

- Firstly, $[P]_{\text{MIA}} \wedge [Q]_{\text{MIA}}$ and $[P \wedge Q]_{\text{MIA}}$ are isomorphic on input-transitions since the Rules (IMust1)–(IMust3) (and Rules (IMay1–(IMay3)) exactly correspond to Rules (I1)–(I3), as well as on P and Q.
- Secondly, consider a transition $[p] \wedge [q] \overset{\tau}{\dashrightarrow} [p'] \wedge [q]$ according to Rule (May1) and $[p] \overset{\tau}{=\!=\!\twoheadrightarrow}_P [p']$. Then, $p \wedge q \overset{\tau}{\Longrightarrow} p' \wedge q$ in IA by repeated application of Rule (T1) and, therefore, $[p \wedge q] \overset{\tau}{=\!=\!\twoheadrightarrow} [p' \wedge q]$ in the IA-embedding. Rule (May2) is analogous, and Rule (May3) for $\alpha = \tau$ is similar (with interleaving of τ-steps). In addition, Rule (May3) for $\alpha \in O$ is similar, too, except that the τ-steps are followed by an α-transition according to Rule (O). □

We observe that the IA-embedding into MIA is 'better' wrt. conjunction than that into dMTS since refinement holds in both directions. The reason is that MIA-refinement is coarser (i.e., larger) than dMTS-refinement applied to MIAs (which are dMTSs after all): input may-transitions do not have to be matched in the former. Thus, there can be more lower bounds wrt. MIA-refinement and the greatest lower bound can be larger.

Proposition 4.18 (Disjunction and IA-Embedding)**.** *For IAs P, Q with $p \in P$, $q \in Q$, we have:* $[p] \vee [q] \sqsubseteq_{MIA} [p \vee q]$.

This result holds by general order theory due to Thm. 4.16. The reverse refinement for disjunction is not valid as we have already seen in Fig. 11, and this difference repairs a shortcoming of IA-disjunction as discussed on p. 7.

5. Conclusions and Future Work

We introduced *Modal Interface Automata* (MIA), an interface theory that is more expressive than *Interface Automata* (IA) [dH05]: it allows one to mandate that a specification's refinement must implement some output, thus excluding trivial implementations, e.g., one that accepts all inputs but never emits any output. This was also the motivation behind *IOMTS* [LNW07] that extends *Modal Transition Systems* (MTS) [Lar90] by inputs and outputs; however, the IOMTS-parallel operator in the style of IA is not compositional. Apart from having disjunctive must-transitions, MIA is a subset of IOMTS, but it has a different refinement relation that is a precongruence for parallel composition.

Most importantly and in contrast to IA and IOMTS, the MIA theory is equipped with a conjunction operator for reasoning about components that satisfy multiple interfaces simultaneously. Along the way, we also introduced conjunction on IA and a disjunctive extension of MTS – as well as disjunction on IA, MTS and MIA – and proved these operators to be the desired greatest lower bounds (resp., least upper bounds) and thus compositional. Compared to the language-based modal interface theory of [RBB+11], our formalism supports nondeterministic specifications and allows limited nondeterminism (in the sense of deterministic *disjunctive* transitions) even for inputs. Hence, MIA establishes a theoretically clean and practical interface theory that fixes the shortcomings of related work.

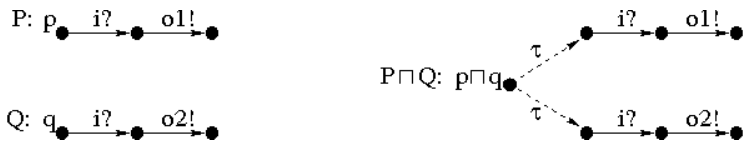

Figure 14: In Logic LTS [LV10], disjunction is internal choice.

From a technical perspective, our MIA-theory borrows from our earlier work on Logic LTS [LV10]. There, we started from a very different conjunction operator appropriate for a deadlock-sensitive CSP-like process theory, and then derived a 'best' suitable refinement relation. In [LV10], disjunction is simply internal choice $\sqcap$, as sketched in Fig. 14. For MIA, $p \sqcap q$ is not suited at all since both p and q require that input i is performed immediately.

Future work shall follow both theoretical and practical directions. On the theoretical side, we firstly wish to study MIA's expressiveness in comparison to other theories via thoroughness [FFELS09]. More substantially, however, we intend to enrich MIA with temporal-logic operators, in the spirit of truly mixing operational and temporal-logic styles of specification in the line of our *Logic LTS* in [LV11]. Important guidance for this will be the work of Feuillade and Pinchinat [FP07], who have introduced a temporal logic for modal interfaces that is equally expressive to MTS. In contrast to [LV11], their setting is not mixed, does not consider nondeterminism, and does not include a refinement relation. Indeed, a unique feature of Logic LTS is that its refinement relation subsumes the standard temporal-logic satisfaction relation.

On the practical side, we plan to study the algorithmic complexity implied by MIA-refinement, on the basis of existing literature for MTS. For example, Antonik et al. [AHL$^+$10] discuss related decision problems such as the existence of a common implementation; Fischbein and Uchitel [FU08] generalize the conjunction of [LSW95] and study its algorithmic aspects; Beneš et al. [BCK11] show that refinement problems for DMTS are not harder than in the case of MTS and also consider conjunction; Raclet et al. [RBB$^+$11] advocate deterministic automata for modal interface theories in order to reduce complexity. In addition, we wish to adapt existing tool support for interface theories to MIA, e.g., the *MIO Workbench* [BMSH10].

Acknowledgement

We thank the anonymous reviewers for their constructive comments and for pointing out additional related work. Part of this research was supported by the DFG (German Research Foundation) under grant nos. LU 1748/3-1 and VO 615/12-1 ("Foundations of Heterogeneous Specifications Using State Machines and Temporal Logic").

References

[AHL$^+$10] A. Antonik, M. Huth, K.G. Larsen, U. Nyman, and A. Wasowski. Modal and mixed specifications: Key decision problems and their complexities. *Mathematical Structures in Computer Science*, 20(1):75–103, 2010.

[AL95] M. Abadi and L. Lamport. Conjoining specifications. *ACM TOPLAS*, 1(3):507–534, 1995.

[BCHS07] D. Beyer, A. Chakrabarti, T.A. Henzinger, and S.A. Seshia. An application of web-service interfaces. In *ICWS*, pages 831–838. IEEE, 2007.

[BCK11] N. Beneš, I. Cerná, and J. Křetínský. Modal transition systems: Composition and LTL model checking. In *ATVA*, volume 6996 of *LNCS*, pages 228–242. Springer, 2011.

[BHW11] S. Bauer, R. Hennicker, and M. Wirsing. Interface theories for concurrency and data. *Theoret. Comp. Sc.*, 412(28):3101–3121, 2011.

[BJL+12] S. Bauer, L. Juhl, K. G. Larsen, A. Legay, and J. Srba. Extending modal transition systems with structured labels. *Mathematical Structures in Computer Science*, 22(4):581–617, 2012.

[BMSH10] S. Bauer, P. Mayer, A. Schroeder, and R. Hennicker. On weak modal compatibility, refinement, and the MIO Workbench. In *TACAS*, volume 6015 of *LNCS*, pages 175–189. Springer, 2010.

[CCJK12] T. Chen, C. Chilton, B. Jonsson, and M. Kwiatkowska. A compositional specification theory for component behaviours. In *ESOP*, volume 7211 of *LNCS*, pages 148–168. Springer, 2012.

[dH01] L. de Alfaro and T.A. Henzinger. Interface automata. In *FSE*, pages 109–120. ACM, 2001.

[dH05] L. de Alfaro and T.A. Henzinger. Interface-based design. In *Engineering Theories of Software-Intensive Systems*, volume 195 of *NATO Science Series*. Springer, 2005.

[DHJP08] L. Doyen, T.A. Henzinger, B. Jobstmann, and T. Petrov. Interface theories with component reuse. In *EMSOFT*, pages 79–88. ACM, 2008.

[Dil89] D.L. Dill. *Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits*. MIT Press, 1989.

[FFELS09] H. Fecher, D. de Frutos-Escrig, G. Lüttgen, and H. Schmidt. On the expressiveness of refinement settings. In *FSEN*, volume 5961 of *LNCS*, pages 276–291. Springer, 2009.

[FP07] G. Feuillade and S. Pinchinat. Modal specifications for the control theory of discrete event systems. *J. Discrete Event Dyn. Syst.*, 17:211–232, 2007.

[FU08] D. Fischbein and S. Uchitel. On correct and complete strong merging of partial behaviour models. In *SIGSOFT FSE*, pages 297–307. ACM, 2008.

[HLL+12] J. Hatcliff, G. T. Leavens, K. R. M. Leino, P. Müller, and M. Parkinson. Behavioral interface specification languages. *ACM Computing Surveys*, 44(3):16, 2012.

[Lar90] K.G. Larsen. Modal specifications. In *Automatic Verification Methods for Finite State Systems*, volume 407 of *LNCS*, pages 232–246. Springer, 1990.

[LNW07] K.G. Larsen, U. Nyman, and A. Wasowski. Modal I/O automata for interface and product line theories. In *ESOP*, volume 4421 of *LNCS*, pages 64–79. Springer, 2007.

[LSW95] K.G. Larsen, B. Steffen, and C. Weise. A constraint oriented proof methodology based on modal transition systems. In *TACAS*, volume 1019 of *LNCS*, pages 17–40. Springer, 1995.

[LV10] G. Lüttgen and W. Vogler. Ready simulation for concurrency: It's logical! *Inform. and Comput.*, 208:845–867, 2010.

[LV11] G. Lüttgen and W. Vogler. Safe reasoning with Logic LTS. *Theoret. Comp. Sc.*, 412(28):3337–3357, 2011.

[LX90] K.G. Larsen and L. Xinxin. Equation solving using modal transition systems. In *LICS*, pages 108–117. IEEE, 1990.

[MB03] L. G. Meredith and S. Bjorg. Contracts and types. *C. ACM*, 46(10):41–47, 2003.

[Mey92] B. Meyer. Applying design by contract. *IEEE Computer*, 25(10):40–51, 1992.

[MG05] W. Maydl and L. Grunske. Behavioral types for embedded software – A survey. In *Component-Based Software Development*, volume 3778 of *LNCS*, pages 82–106. Springer, 2005.

[RBB+11] J. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay, and R. Passerone. A modal interface theory for component-based design. *Fund. Inform.*, 107:1–32, 2011.

[VW02] W. Vogler and R. Wollowski. Decomposition in asynchronous circuit design. In *Concurrency and Hardware Design*, volume 2549 of *LNCS*, pages 152–190. Springer, 2002.