

Department of Computer Science

**A Compositional Specification Theory
for Component Behaviours**

**Taolue Chen, Chris Chilton,
Bengt Jonsson, and Marta Kwiatkowska**

CS-RR-12-01



Department of Computer Science, University of Oxford
Wolfson Building, Parks Road, Oxford, OX1 3QD

Addendum: 28 February 2013

Please note that the proofs in this report have not been updated to reflect corrections made to the main body. The proofs should therefore serve as a guide.

Readers are advised that the content of this report has largely been superseded by:

C. Chilton, B. Jonsson, and M. Kwiatkowska.: An Algebraic Theory of Interface Automata. Technical Report CS-RR-13-02, Department of Computer Science, University of Oxford (2013).

A Compositional Specification Theory for Component Behaviours^{*}

Taolue Chen¹, Chris Chilton¹, Bengt Jonsson², and Marta Kwiatkowska¹

¹ Department of Computer Science, University of Oxford, UK

² Department of Information Technology, Uppsala University, Sweden

Abstract. We propose a compositional specification theory for reasoning about components that interact by synchronisation of input and output (I/O) actions, in which the specification of a component constrains the temporal ordering of interactions with the environment. Such a theory is motivated by the need to support composability of components, in addition to modelling environmental assumptions, and reasoning about run-time behaviour. Models can be specified operationally by means of I/O labelled transition systems augmented by an inconsistency predicate on states, or in a purely declarative manner by means of traces. We introduce a refinement preorder that supports safe-substitutivity of components. Our specification theory includes the operations of parallel composition for composing components at run-time, logical conjunction for independent development, and quotient for incremental development. We prove congruence properties of the operations and show correspondence between the operational and declarative frameworks.

Keywords: specification theory, compositionality, components, I/O automata, interface automata, logic LTS, refinement, conjunction, quotient.

1 Introduction

An important paradigm for developing complex reactive systems is component-based design, where systems are composed from components, which themselves can be realised by smaller components. Component-based design can be supported by a specification theory, which allows the mixing of specifications and implementations, admits refinement, and provides composition operators. A specification theory suitable for components should be equipped with a *refinement* preorder which is substitutive, to facilitate component reuse. As a minimum, the composition operators should include structural *parallel composition*, for inferring component interactions at run-time; *conjunction*, to facilitate independent development constrained by several specifications; and *quotienting*, which supports incremental development in the following sense: given a specification of the full system, together with components implementing part of that system,

^{*} Technical Report CS-RR-12-01, Department of Computer Science, University of Oxford (2012). This report, containing proofs, is an extended version of [1].

quotienting allows one to find the coarsest specification of the remaining portion of the system to be implemented. Further useful operators include: *disjunction*, which finds a common specification that a collection of components implement; and *hiding*, which supports abstraction of components.

In this paper, we consider systems of components that interact by synchronisation of input and output actions, in which outputs are non-blocking. A specification should describe properties on the ordering of a component’s interactions with its environment; it should also describe the assumptions on the environment under which these properties are guaranteed, thereby supporting assume-guarantee reasoning. A number of proposals for such specification theories have been put forward. As detailed in the survey of related work below, we find that they suffer from limitations or unnecessary complications.

The main contribution of our paper is a comprehensive, compositional specification theory for components that generalises existing frameworks by supporting all the above-mentioned operators, while retaining conceptual simplicity and strong algebraic properties of the operations. The framework permits the mixing of abstract component specifications and I/O labelled transition systems (called Logic IOLTSSs), without restricting to determinism as in [2]. Our refinement is based on traces, hence admitting a simpler formulation than similar notions based on e.g. alternating simulation [3] or modalities [4,5], so is more amenable to language-theoretic constructs. From this formulation, we demonstrate that the induced mutual refinement is a congruence for the operators.

In contrast to existing I/O automata [6] and interface automata [3], we are able to express: (1) *assumptions* on the input provided by the environment; (2) *underspecification*, meaning that it is uncertain what the allowable interactions are; and (3) various (run-time) *errors*, including communication mismatch and bad behaviour. We show that all these features can be expressed using only the single concept of *inconsistency*, which we have adapted from the Logic LTSs of Lüttgen and Vogler [7,8], where input and output actions are not distinguished. Inconsistency is a property of states or interaction traces, which represents the possibility of some abnormal condition. Once an inconsistency has occurred, there is no escaping from it. Following the lead of CSP [9], we thus allow for chaotic behaviour to ensue once an inconsistency has arisen.

Related work. Our specification theory, in particular Logic IOLTSSs, is inspired by the Logic LTS framework due to Lüttgen and Vogler [7], a compositional theory that admits as specifications LTSs without I/O distinctions. Their inconsistency predicate is induced from inequality of ready-sets, rather than communication mismatches as in our case. Refinement is based on ready-simulation; alphabetised parallel and conjunction are considered, but not quotient.

The operational component model in our framework has been greatly influenced by I/O automata [6] and interface automata [3]: both are based on I/O LTSs, with the proviso that I/O automata must be input-enabled, meaning that each state of the automaton is willing to accept any input. We differ from I/O automata by not imposing input-enabledness and from interface automata by working with an explicit representation of inconsistencies. Another difference is

refinement, which for interface automata is defined in terms of alternating simulation, rather than traces; the original definition in [3] is simplified in [10], but works only for input-deterministic interface automata. It should be noted that, unlike [3,10], we use an associative variant of parallel composition, which combines an input and output into an output (as in [11]). Furthermore, we provide a definition of conjunction corresponding to shared refinement of interface automata, which substantially generalises that of [12] for synchronous components. Moreover, our quotienting operator on Logic IOLTSs generalises that in [13] defined only for deterministic components.

There are a number of process-algebraic frameworks that deal with asynchronous I/O interaction. We mention a characterisation of I/O automata by De Nicola and Segala [14], which is actually a generalisation (and also applicable to interface automata), since the inconsistent process Ω allows to distinguish between good and bad inputs. Similarly to our approach, refinement in [14] is given by trace containment, but does not extend to inconsistent trace containment. This is because we allow a Logic IOLTS to become inconsistent after emitting an output, whereas a process can only become inconsistent through receiving a bad input. Finally, we remark that [14] supports a number of operators of a specification theory, but does not deal with conjunction or quotient.

Our work is also related to the *ioco* theory in model based testing [15]. The *ioco* relation is similar to our refinement, but lacks compositionality of operators, so is not well-suited to a specification theory for components.

There have been several CSP-based frameworks that deal with asynchronous communication; of these, the receptive process theory (RPT) [16] utilises a model of concurrency similar to ours in that outputs are non-blocking. RPT also considers quotient (referred to as factorisation), but for the restricted class of delay-insensitive networks [17] that differ from our setting.

A further class of component-based modelling formalisms is based on may/must modalities. A specification theory for components has been devised in [18] based on modalities [4,5], but the definition of quotient is more restrictive than ours. Larsen *et al.* have made an effort in relating modal transition systems with interface automata [2]. The approach of modal I/O automata is based on a game-like definition of refinement, which we claim to be more complex than ours, see, e.g., the discussion of parallel composition in [5]. The framework in [5] can support reasoning about liveness properties which our framework does not (although they both support reasoning of safety properties). However, our framework can be easily extended by introducing quiescent states, and additionally considering containment of quiescent traces to reason about liveness.

Outline. The paper begins by introducing declarative specifications in Section 2, before considering operational specifications in Section 3. We focus on three composition operators: parallel, conjunction and quotient; omitting disjunction and hiding for reasons of space. The paper ends with a statement of full-abstraction results in Section 4. Proofs for our claims can be found in the appendices.

2 A Declarative Theory of Components

In this section, we model components abstractly by means of declarative specifications. We introduce a substitutive refinement preorder together with three compositional operators on declarative specifications.

A declarative specification comes equipped with an interface, together with a set of behaviours over the interface. The interface is represented by a set of input actions and a set of output actions, which are necessarily disjoint, while the behaviour is characterised by traces.

Definition 1 (Declarative specification). *A declarative specification \mathcal{P} is a tuple $\langle \mathcal{A}_{\mathcal{P}}^I, \mathcal{A}_{\mathcal{P}}^O, T_{\mathcal{P}}, F_{\mathcal{P}} \rangle$ in which $\mathcal{A}_{\mathcal{P}}^I$ and $\mathcal{A}_{\mathcal{P}}^O$ are disjoint sets referred to as inputs and outputs respectively (the union of which is denoted by $\mathcal{A}_{\mathcal{P}}$), $T_{\mathcal{P}} \subseteq \mathcal{A}_{\mathcal{P}}^*$ is a non-empty set of permissible traces, and $F_{\mathcal{P}} \subseteq \mathcal{A}_{\mathcal{P}}^*$ is a set of inconsistent traces. The trace sets must satisfy the constraints:*

1. $F_{\mathcal{P}} \subseteq T_{\mathcal{P}}$
2. If $t \in T_{\mathcal{P}}$ and $i \in \mathcal{A}_{\mathcal{P}}^I$, then $ti \in T_{\mathcal{P}}$
3. $T_{\mathcal{P}}$ is prefix closed
4. If $t \in F_{\mathcal{P}}$ and $t' \in \mathcal{A}_{\mathcal{P}}^*$, then $tt' \in F_{\mathcal{P}}$.

Outputs are under the control of the component, whereas inputs are issued by the environment. This means that, after any successful interaction between the component and the environment, the environment can issue any input i , even if it will be refused by the component. Naturally, if i is refused by the component after the trace t , we deem ti to be an inconsistent trace, since a communication mismatch has occurred. Given this treatment of inputs, we say that our theory is *not* input-enabled, even though $T_{\mathcal{P}}$ is closed under input-extensions.

Example 1. A drinks machine dispenses either a tea or a coffee after a coin has been inserted. The drinks machine has sufficient water to produce only 2 drinks, after which a further coin insertion renders the machine inoperable. This behaviour can be encoded by the declarative specification $DM = \langle \{\mathcal{L}\}, \{t, c\}, T, F_1 \cup F_2 \rangle$, where:

- $T = \{\epsilon, \mathcal{L}, \mathcal{L}(c+t), \mathcal{L}(c+t)\mathcal{L}, \mathcal{L}(c+t)\mathcal{L}(c+t)\} \cup F_1 \cup F_2$
- $F_1 = \mathcal{L}(c+t)\mathcal{L}(c+t)\mathcal{L}(\mathcal{L}+c+t)^*$ insertion of third coin after two dispensations
- $F_2 = (\epsilon + \mathcal{L}(c+t))\mathcal{L}\mathcal{L}(\mathcal{L}+c+t)^*$ insertion of second coin before dispensation.

From hereon let \mathcal{P} , \mathcal{Q} and \mathcal{R} be declarative specifications with signatures $\langle \mathcal{A}_{\mathcal{P}}^I, \mathcal{A}_{\mathcal{P}}^O, T_{\mathcal{P}}, F_{\mathcal{P}} \rangle$, $\langle \mathcal{A}_{\mathcal{Q}}^I, \mathcal{A}_{\mathcal{Q}}^O, T_{\mathcal{Q}}, F_{\mathcal{Q}} \rangle$ and $\langle \mathcal{A}_{\mathcal{R}}^I, \mathcal{A}_{\mathcal{R}}^O, T_{\mathcal{R}}, F_{\mathcal{R}} \rangle$ respectively.

2.1 Refinement

As refinement corresponds to safe substitutivity, for \mathcal{Q} to be used in place of \mathcal{P} we require that \mathcal{Q} must exist safely in *any* environment that \mathcal{P} can exist in safely.

Whether an environment is safe for a specification depends on the sequences of message exchanges afforded by the component. If an environment can prevent a component from performing an inconsistent trace, then the environment is said to be safe.

We do not insist that a component \mathcal{Q} must have the same interface as the component \mathcal{P} to be refined. Instead \mathcal{Q} must be accepting of at least all of \mathcal{P} 's inputs, while restricting to a subset of \mathcal{P} 's outputs. This can be formalised by the covariant relationship $\mathcal{A}_{\mathcal{P}}^I \subseteq \mathcal{A}_{\mathcal{Q}}^I$ on inputs and the contravariant constraint $\mathcal{A}_{\mathcal{Q}}^O \subseteq \mathcal{A}_{\mathcal{P}}^O$ on outputs.

In order to establish that refinement holds, we perform a weak form of alphabet equalisation on the inputs of the component to be refined. We refer to this operation as lifting. Informally, lifting extends the trace sets of \mathcal{P} by explicitly refusing any input in $\mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{P}}^I$, after which it allows for arbitrary behaviour.

Definition 2 (Lifting). *Let \mathcal{P} be a declarative specification, and let $\mathcal{A}_{\mathcal{Q}}^I$ be a set of input actions. The lifting of trace sets $T_{\mathcal{P}}$ and $F_{\mathcal{P}}$ to $\mathcal{A}_{\mathcal{Q}}^I$, written as $T_{\mathcal{P}} \uparrow \mathcal{A}_{\mathcal{Q}}^I$ and $F_{\mathcal{P}} \uparrow \mathcal{A}_{\mathcal{Q}}^I$ respectively, is defined as:*

$$\begin{aligned} - T_{\mathcal{P}} \uparrow \mathcal{A}_{\mathcal{Q}}^I &= T_{\mathcal{P}} \cup \{tit' : t \in T_{\mathcal{P}}, i \in \mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{P}}^I \text{ and } t' \in (\mathcal{A}_{\mathcal{Q}}^I \cup \mathcal{A}_{\mathcal{P}})^*\} \\ - F_{\mathcal{P}} \uparrow \mathcal{A}_{\mathcal{Q}}^I &= F_{\mathcal{P}} \cup \{tit' : t \in T_{\mathcal{P}}, i \in \mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{P}}^I \text{ and } t' \in (\mathcal{A}_{\mathcal{Q}}^I \cup \mathcal{A}_{\mathcal{P}})^*\}. \end{aligned}$$

Recall that an environment is safe for a component if the environment can prevent the component from performing an inconsistent trace. As outputs are under the control of the component itself, a safe environment must refuse to issue an input on any trace from which there is a sequence of output actions after the input that allows the trace to become inconsistent.

Under such an arrangement, for each declarative specification \mathcal{P} we can define the safe declarative specification $\mathcal{E}(\mathcal{P})$ containing all of \mathcal{P} 's permissible and inconsistent traces, but also satisfying the additional property: if $t \in T_{\mathcal{P}}$ and there exists $t' \in (\mathcal{A}_{\mathcal{P}}^O)^*$ such that $tt' \in F_{\mathcal{P}}$, then $t \in F_{\mathcal{E}(\mathcal{P})}$. This has the effect of forcing all inconsistent traces to become inconsistent on the environment's issue of a bad input. If the environment respects this safe specification, by not issuing any input that results in an inconsistent trace, then the component can never encounter an inconsistent trace. Note that if $\epsilon \in F_{\mathcal{E}(\mathcal{P})}$ then there is no environment that can prevent \mathcal{P} from performing an inconsistent trace. However, for uniformity we still refer to $\mathcal{E}(\mathcal{P})$ as the safe specification of \mathcal{P} .

Definition 3 (Safe specification). *Let \mathcal{P} be a declarative specification. The most general safe specification for \mathcal{P} is a declarative specification $\mathcal{E}(\mathcal{P}) = \langle \mathcal{A}_{\mathcal{P}}^I, \mathcal{A}_{\mathcal{P}}^O, T_{\mathcal{E}(\mathcal{P})}, F_{\mathcal{E}(\mathcal{P})} \rangle$, where $T_{\mathcal{E}(\mathcal{P})} = T_{\mathcal{P}} \cup F_{\mathcal{E}(\mathcal{P})}$ and $F_{\mathcal{E}(\mathcal{P})} = \{tt' \in \mathcal{A}_{\mathcal{P}}^* : t \in T_{\mathcal{P}} \text{ and } \exists t'' \in (\mathcal{A}_{\mathcal{P}}^O)^* \cdot tt'' \in F_{\mathcal{P}}\}$.*

We can now define our substitutive refinement preorder. From the safe specification associated with an arbitrary declarative specification, it is easy to see whether a declarative specification can be substituted safely in place of another. Note that $F_{\mathcal{Q}} \subseteq F_{\mathcal{P}} \uparrow \mathcal{A}_{\mathcal{Q}}^I$ would be too strong to use for the last clause, as we are only interested in trace containment up to the point where an environment can issue a bad input.

Definition 4 (Refinement). For declarative specifications \mathcal{P} and \mathcal{Q} , \mathcal{Q} is said to be a refinement of \mathcal{P} , written $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$, iff:

1. $\mathcal{A}_{\mathcal{P}}^I \subseteq \mathcal{A}_{\mathcal{Q}}^I$
2. $\mathcal{A}_{\mathcal{Q}}^O \subseteq \mathcal{A}_{\mathcal{P}}^O$
3. $T_{\mathcal{E}(\mathcal{Q})} \subseteq T_{\mathcal{E}(\mathcal{P})} \uparrow \mathcal{A}_{\mathcal{Q}}^I$
4. $F_{\mathcal{E}(\mathcal{Q})} \subseteq F_{\mathcal{E}(\mathcal{P})} \uparrow \mathcal{A}_{\mathcal{Q}}^I$.

As refinement is based on an extension of language inclusion, its complexity is in P, assuming regularity of the trace sets. Note that lifting maintains regularity.

Equivalence of declarative specifications in our framework is defined in terms of mutual refinement.

Definition 5 (Equivalence). Let \mathcal{P} and \mathcal{Q} be declarative specifications. Then \mathcal{P} and \mathcal{Q} are said to be equivalent, written $\mathcal{P} \equiv_{dec} \mathcal{Q}$, iff $\mathcal{P} \sqsubseteq_{dec} \mathcal{Q}$ and $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$.

Lemma 1 (Preorder). Refinement is both reflexive and transitive.

2.2 Parallel composition

The parallel composition operator on declarative specifications yields a declarative specification representing the combined effect of its operands running asynchronously. We do not consider synchronous parallel composition, as this does not make sense when dealing with non-blocking output actions. To preserve the effect that a single output from a component can be received by multiple components in the environment, we must define the parallel composition to repeatedly broadcast an output: this means that an input $a?$ and output $a!$ combine to form an output $a!$ (as in certain variants of I/O automata), rather than a hidden action τ as is the case in Milner's CCS.

Not all declarative specifications can be composed with one another; we restrict to those that are said to be *composable*. \mathcal{P} and \mathcal{Q} are composable for parallel composition only if $\mathcal{A}_{\mathcal{P}}^O \cap \mathcal{A}_{\mathcal{Q}}^O = \emptyset$. This restriction is meaningful if we consider inputs on an interface as buttons and outputs as lights. Given two distinct components, it is not possible for them to share a common light, whereas it is possible to push their buttons at the same time. In practice, issues of composability can be avoided by employing renaming, if this is considered to be appropriate.

Definition 6 (Parallel composition). Let \mathcal{P} and \mathcal{Q} be declarative specifications such that $\mathcal{A}_{\mathcal{P}}^O$ and $\mathcal{A}_{\mathcal{Q}}^O$ are disjoint. Then $\mathcal{P} \parallel \mathcal{Q}$ is the declarative specification $\langle \mathcal{A}_{\mathcal{P} \parallel \mathcal{Q}}^I, \mathcal{A}_{\mathcal{P} \parallel \mathcal{Q}}^O, T_{\mathcal{P} \parallel \mathcal{Q}}, F_{\mathcal{P} \parallel \mathcal{Q}} \rangle$, where:

- $\mathcal{A}_{\mathcal{P} \parallel \mathcal{Q}}^I = (\mathcal{A}_{\mathcal{P}}^I \cup \mathcal{A}_{\mathcal{Q}}^I) \setminus (\mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{Q}}^O)$
- $\mathcal{A}_{\mathcal{P} \parallel \mathcal{Q}}^O = \mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{Q}}^O$
- $T_{\mathcal{P} \parallel \mathcal{Q}} = \{t \in \mathcal{A}_{\mathcal{P} \parallel \mathcal{Q}}^* : t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}} \text{ and } t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{Q}}\} \cup F_{\mathcal{P} \parallel \mathcal{Q}}$
- $F_{\mathcal{P} \parallel \mathcal{Q}} = \{tt' \in \mathcal{A}_{\mathcal{P} \parallel \mathcal{Q}}^* : t \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}} \text{ and } t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{Q}}, \text{ or } t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}} \text{ and } t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in F_{\mathcal{Q}}\}$.

Informally, a trace is permissible in $\mathcal{P} \parallel \mathcal{Q}$ if its projection onto $\mathcal{A}_{\mathcal{P}}$ is a trace of \mathcal{P} and its projection onto $\mathcal{A}_{\mathcal{Q}}$ is a trace of \mathcal{Q} . A trace is inconsistent if it has a prefix whose projection onto the alphabet of one of the components is inconsistent and the projection onto the alphabet of the other component is a permissible trace of that component.

We demonstrate the following result, a corollary of which is that mutual refinement is a congruence for parallel, subject to composability.

Theorem 1 (Compositionality of parallel). *Let \mathcal{P} , \mathcal{Q} and \mathcal{R} be declarative specifications such that \mathcal{P} and \mathcal{R} are composable for parallel composition, and $\mathcal{A}_{\mathcal{Q}} \cap \mathcal{A}_{\mathcal{R}} \cap \mathcal{A}_{\mathcal{P} \parallel \mathcal{R}} \subseteq \mathcal{A}_{\mathcal{P}} \cap \mathcal{A}_{\mathcal{R}}$. If $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$, then $\mathcal{Q} \parallel \mathcal{R} \sqsubseteq_{dec} \mathcal{P} \parallel \mathcal{R}$.*

2.3 Conjunction

The conjunction operator on declarative specifications can be thought of as finding a common implementation for a number of properties, each of which are represented by declarative specifications. Naturally, any implementation of these properties should be a refinement of each of the properties to be implemented. The conjunction (or shared refinement) of two declarative specifications \mathcal{P} and \mathcal{Q} is the *coarsest* declarative specification that refines both \mathcal{P} and \mathcal{Q} . Thus conjunction is the meet operator on the refinement preorder.

As for parallel composition, conjunction can only be performed on composable components. \mathcal{P} and \mathcal{Q} are composable for conjunction only if the sets $\mathcal{A}_{\mathcal{P}}^I \cup \mathcal{A}_{\mathcal{Q}}^I$ and $\mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{Q}}^O$ are disjoint.

Definition 7 (Conjunction). *Let \mathcal{P} and \mathcal{Q} be declarative specifications such that $\mathcal{A}_{\mathcal{P}}^I \cup \mathcal{A}_{\mathcal{Q}}^I$ and $\mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{Q}}^O$ are disjoint. Then $\mathcal{P} \wedge \mathcal{Q}$ is the declarative specification $\langle \mathcal{A}_{\mathcal{P} \wedge \mathcal{Q}}^I, \mathcal{A}_{\mathcal{P} \wedge \mathcal{Q}}^O, T_{\mathcal{P} \wedge \mathcal{Q}}, F_{\mathcal{P} \wedge \mathcal{Q}} \rangle$, where:*

- $\mathcal{A}_{\mathcal{P} \wedge \mathcal{Q}}^I = \mathcal{A}_{\mathcal{P}}^I \cup \mathcal{A}_{\mathcal{Q}}^I$
- $\mathcal{A}_{\mathcal{P} \wedge \mathcal{Q}}^O = \mathcal{A}_{\mathcal{P}}^O \cap \mathcal{A}_{\mathcal{Q}}^O$
- $T_{\mathcal{P} \wedge \mathcal{Q}} = T_{\mathcal{P}} \uparrow \mathcal{A}_{\mathcal{Q}}^I \cap T_{\mathcal{Q}} \uparrow \mathcal{A}_{\mathcal{P}}^I$
- $F_{\mathcal{P} \wedge \mathcal{Q}} = F_{\mathcal{P}} \uparrow \mathcal{A}_{\mathcal{Q}}^I \cap F_{\mathcal{Q}} \uparrow \mathcal{A}_{\mathcal{P}}^I$.

Conjunction has strong connections with the logical ‘and’ operator in Boolean algebra, as shown below. Mutual refinement is a congruence for conjunction, subject to composability.

Theorem 2 (Properties of conjunction).

- *Conjunction is the greatest lower bound operator for \sqsubseteq_{dec}*
- *$\mathcal{R} \sqsubseteq_{dec} \mathcal{P}$ and $\mathcal{R} \sqsubseteq_{dec} \mathcal{Q}$ iff $\mathcal{R} \sqsubseteq_{dec} \mathcal{P} \wedge \mathcal{Q}$*
- *$\mathcal{P} \wedge \mathcal{Q} \equiv_{dec} \mathcal{Q}$ iff $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$.*

Theorem 3 (Compositionality of conjunction). *Let \mathcal{P} , \mathcal{Q} and \mathcal{R} be declarative specifications such that \mathcal{P} is composable with \mathcal{R} for conjunction. If $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$, then $\mathcal{Q} \wedge \mathcal{R} \sqsubseteq_{dec} \mathcal{P} \wedge \mathcal{R}$.*

2.4 Quotient

The final operation that we consider on the specification theory is that of quotienting, which has strong connections to synthesis. Given a specification for a system \mathcal{R} , together with a component \mathcal{P} implementing part of \mathcal{R} , the quotient yields the *coarsest* specification for the remaining part of \mathcal{R} to be implemented. Thus, the parallel composition of the quotient with \mathcal{P} should be a refinement of the system-wide specification \mathcal{R} . Therefore, quotient can be thought of as the adjoint of parallel composition.

As \mathcal{P} is a sub-component of \mathcal{R} , we make the reasonable assumption that $\mathcal{A}_{\mathcal{P}} \subseteq \mathcal{A}_{\mathcal{R}}$. Moreover, a necessary condition for the existence of the quotient is that $\mathcal{A}_{\mathcal{P}}^O \subseteq \mathcal{A}_{\mathcal{R}}^O$, otherwise refinement will fail on the alphabet containment checks.

Definition 8 (Quotient). *Let \mathcal{P} and \mathcal{R} be declarative specifications such that $\mathcal{A}_{\mathcal{P}}^O \subseteq \mathcal{A}_{\mathcal{R}}^O$ and $\mathcal{A}_{\mathcal{P}} \subseteq \mathcal{A}_{\mathcal{R}}$. The quotient of \mathcal{P} from \mathcal{R} is the specification \mathcal{R}/\mathcal{P} with signature $\langle \mathcal{A}_{\mathcal{R}/\mathcal{P}}^I, \mathcal{A}_{\mathcal{R}/\mathcal{P}}^O, T_{\mathcal{R}/\mathcal{P}}, F_{\mathcal{R}/\mathcal{P}} \rangle$, where:*

- $\mathcal{A}_{\mathcal{R}/\mathcal{P}}^I = \mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{R}}^I$
- $\mathcal{A}_{\mathcal{R}/\mathcal{P}}^O = \mathcal{A}_{\mathcal{R}}^O \setminus \mathcal{A}_{\mathcal{P}}^O$
- $T_{\mathcal{R}/\mathcal{P}} = \{t \in \mathcal{A}_{\mathcal{R}}^* : \forall t' \text{ a prefix of } t \cdot L(t') \text{ and } \forall t'' \in \mathcal{A}_{\mathcal{R}/\mathcal{P}}^* \cdot L(tt'')\}$
- $F_{\mathcal{R}/\mathcal{P}} = \{t \in \mathcal{A}_{\mathcal{R}}^* : (t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}} \implies t \in F_{\mathcal{E}(\mathcal{R})}) \text{ and } \forall t' \text{ a prefix of } t \cdot L(t')\}$
- $L(t) = (t \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}} \implies t \in F_{\mathcal{E}(\mathcal{R})}) \text{ and } (t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}} \implies t \in T_{\mathcal{R}})$.

The alphabet of the quotient contains all of the actions from $\mathcal{A}_{\mathcal{R}}$ and $\mathcal{A}_{\mathcal{P}}$ so that \mathcal{R}/\mathcal{P} can fully control \mathcal{P} and emulate the behaviour of \mathcal{R} . Yet still, simple examples reveal that there may not exist a component \mathcal{Q} over an interface consisting of inputs $\mathcal{A}_{\mathcal{R}/\mathcal{P}}^I$ and outputs $\mathcal{A}_{\mathcal{R}/\mathcal{P}}^O$ such that $\mathcal{P} \parallel \mathcal{Q} \sqsubseteq_{dec} \mathcal{R}$. Unfortunately, the existence of the quotient cannot be ascertained by a syntactic check on the alphabets of \mathcal{P} and \mathcal{R} .

In Definition 8 we referred to \mathcal{R}/\mathcal{P} as a specification, but not a declarative specification. As the following theorem shows, if $T_{\mathcal{R}/\mathcal{P}}$ is non-empty (a condition of being a declarative specification), then the quotient exists.

Theorem 4 (Existence of quotient). *Let \mathcal{P} and \mathcal{R} be declarative specifications such that $\mathcal{A}_{\mathcal{P}}^O \subseteq \mathcal{A}_{\mathcal{R}}^O$ and $\mathcal{A}_{\mathcal{P}} \subseteq \mathcal{A}_{\mathcal{R}}$. Then there exists a declarative specification \mathcal{Q} with input actions $\mathcal{A}_{\mathcal{R}/\mathcal{P}}^I$ and output actions $\mathcal{A}_{\mathcal{R}/\mathcal{P}}^O$ such that $\mathcal{P} \parallel \mathcal{Q} \sqsubseteq_{dec} \mathcal{R}$ iff $T_{\mathcal{R}/\mathcal{P}} \neq \emptyset$.*

The next two theorems show that \mathcal{R}/\mathcal{P} satisfies the required properties of quotient when $T_{\mathcal{R}/\mathcal{P}}$ is non-empty, and that quotient is well-behaved with respect to refinement.

Theorem 5 (Properties of quotient). *Let \mathcal{P} and \mathcal{R} be declarative specifications such that $\mathcal{A}_{\mathcal{P}}^O \subseteq \mathcal{A}_{\mathcal{R}}^O$ and $\mathcal{A}_{\mathcal{P}} \subseteq \mathcal{A}_{\mathcal{R}}$. If $T_{\mathcal{R}/\mathcal{P}} \neq \emptyset$, then $\mathcal{P} \parallel (\mathcal{R}/\mathcal{P}) \sqsubseteq_{dec} \mathcal{R}$ and for any declarative specification \mathcal{Q} over inputs $\mathcal{A}_{\mathcal{R}/\mathcal{P}}^I$ and outputs $\mathcal{A}_{\mathcal{R}/\mathcal{P}}^O$ such that $\mathcal{P} \parallel \mathcal{Q} \sqsubseteq_{dec} \mathcal{R}$ it holds that $\mathcal{Q} \sqsubseteq_{dec} \mathcal{R}/\mathcal{P}$.*

Theorem 6 (Compositionality of quotient). *Let \mathcal{P} , \mathcal{Q} and \mathcal{R} be declarative specifications such that $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$.*

- *If \mathcal{Q}/\mathcal{R} and \mathcal{P}/\mathcal{R} are defined, then $\mathcal{Q}/\mathcal{R} \sqsubseteq_{dec} \mathcal{P}/\mathcal{R}$.*
- *If \mathcal{R}/\mathcal{Q} and \mathcal{R}/\mathcal{P} are defined, and $(\mathcal{A}_{\mathcal{Q}}^I \cap \mathcal{A}_{\mathcal{R}}^O) \setminus \mathcal{A}_{\mathcal{P}} = \emptyset$, then $\mathcal{R}/\mathcal{Q} \sqsupseteq_{dec} \mathcal{R}/\mathcal{P}$.*

3 An Operational Theory of Components

In this section we take an operational view of components, by specifying their allowable interactions in terms of Logic IOLTSs, an I/O version of labelled transition systems augmented by an inconsistency predicate on states. We remain faithful to the trace-based substitutive preorder, and cast refinement at the operational level in terms of declarative refinement. For any operational model, we can derive an equivalent declarative specification, meaning that the observable safe interactions between the models and an arbitrary environment are indistinguishable.

To support a compositional theory of components, we define the operations of parallel composition, conjunction and quotient directly on our operational models. We further show that compositionality results for the operators on the declarative framework carry over to the operational framework as well.

An explicit definition of implementation is not provided for our models, although there are a number of candidates. One such suggestion for the characterisation of implementations would be the set of specifications in which no inconsistent states are reachable. We leave this for the user to decide.

We can now define the operational models formally. For a set \mathcal{A} , write \mathcal{A}^τ as shorthand for $\mathcal{A} \cup \{\tau\}$, where it is assumed that $\tau \notin \mathcal{A}$.

Definition 9 (IOLTS). *An I/O labelled transition system (IOLTS) \mathcal{P} is a tuple $\langle S_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}}^I, \mathcal{A}_{\mathcal{P}}^O, \longrightarrow_{\mathcal{P}} \rangle$, where $S_{\mathcal{P}}$ is a (possibly infinite) collection of processes (states), $\mathcal{A}_{\mathcal{P}}^I$ and $\mathcal{A}_{\mathcal{P}}^O$ are disjoint sets referred to as the inputs and outputs (the union of which we denote by $\mathcal{A}_{\mathcal{P}}$), and $\longrightarrow_{\mathcal{P}} \subseteq S_{\mathcal{P}} \times \mathcal{A}_{\mathcal{P}}^\tau \times S_{\mathcal{P}}$ is the transition relation.*

Note that since we do not insist on our components being fully input-enabled (unlike I/O automata [6]), meaning that at any stage a component can refuse to accept an input issued by the environment or another component, we must extend IOLTSs to reason about potential communication mismatches that occur during interactions. We accomplish this by augmenting IOLTSs with an inconsistency predicate for tracking mismatches. The resulting model, called a Logic IOLTS, takes its inspiration from the Logic LTSs of Lüttgen and Vogler [7,8], although we have a different interpretation of inconsistency.

Definition 10 (Logic IOLTS). *A Logic IOLTS \mathcal{P} is a tuple $\langle S_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}}^I, \mathcal{A}_{\mathcal{P}}^O, \longrightarrow_{\mathcal{P}}, F_{\mathcal{P}} \rangle$ in which $\langle S_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}}^I, \mathcal{A}_{\mathcal{P}}^O, \longrightarrow_{\mathcal{P}} \rangle$ is an IOLTS, and $F_{\mathcal{P}} \subseteq S_{\mathcal{P}}$ is an inconsistency predicate on states.*

The inconsistency predicate annotates states that correspond to run-time errors such as communication mismatches or underspecification. Regardless of why a state is inconsistent, we assume that on encountering an inconsistency, unspecified behaviour can ensue. Consequently, inconsistent states are resemblant of the process CHAOS from CSP [9].

Figure 1 shows a number of Logic IOLTSs represented pictorially. We adopt the convention of enclosing the transition system within a box corresponding to the interface of the component. Labelled arrows pointing at the interface correspond to inputs, whereas arrows emanating from the interface correspond to outputs. As a matter of clarity, we only represent the states that are reachable by a sequence of transitions from the process we are interested in. States annotated with an F are deemed to be inconsistent.

We introduce nomenclature for handling stability and hidden τ -transitions. A relation $\xRightarrow{\epsilon}_{\mathcal{P}} \subseteq S_{\mathcal{P}} \times S_{\mathcal{P}}$ is defined by $p \xRightarrow{\epsilon}_{\mathcal{P}} p'$ iff $p(\xrightarrow{\tau}_{\mathcal{P}})^* p'$. Generalising $\xRightarrow{\epsilon}_{\mathcal{P}}$ for visible actions $a \in \mathcal{A}$, we obtain $p \xRightarrow{a}_{\mathcal{P}} p'$ iff there exist p_a and p'_a such that $p \xRightarrow{\epsilon}_{\mathcal{P}} p_a \xrightarrow{a}_{\mathcal{P}} p'_a \xRightarrow{\epsilon}_{\mathcal{P}} p'$, and $p \xRightarrow{-a}_{\mathcal{P}} p'$ iff there exists p_a such that $p \xrightarrow{a}_{\mathcal{P}} p_a \xRightarrow{\epsilon}_{\mathcal{P}} p'$. The extension to words $w = a_1 \dots a_n$ is defined in the natural way by $p \xRightarrow{w}_{\mathcal{P}} p'$ iff $p \xRightarrow{a_1}_{\mathcal{P}} \dots \xRightarrow{a_n}_{\mathcal{P}} p'$.

Furthermore, for a compositional operator \oplus , and sets A and B , we write $A \oplus B$ for the set $\{a \oplus b : a \in A \text{ and } b \in B\}$. This allows us to use a process-algebraic notation for states.

From hereon, let $\mathcal{P} = \langle S_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}}^I, \mathcal{A}_{\mathcal{P}}^O, \xrightarrow{\cdot}_{\mathcal{P}}, F_{\mathcal{P}} \rangle$, $\mathcal{Q} = \langle S_{\mathcal{Q}}, \mathcal{A}_{\mathcal{Q}}^I, \mathcal{A}_{\mathcal{Q}}^O, \xrightarrow{\cdot}_{\mathcal{Q}}, F_{\mathcal{Q}} \rangle$ and $\mathcal{R} = \langle S_{\mathcal{R}}, \mathcal{A}_{\mathcal{R}}^I, \mathcal{A}_{\mathcal{R}}^O, \xrightarrow{\cdot}_{\mathcal{R}}, F_{\mathcal{R}} \rangle$ be three Logic IOLTSs, and let $p_{\mathcal{P}}$, $q_{\mathcal{Q}}$ and $r_{\mathcal{R}}$ be processes in the Logic IOLTSs \mathcal{P} , \mathcal{Q} and \mathcal{R} respectively.

3.1 Refinement

In keeping with the declarative framework, we wish refinement to correspond to safe-substitutivity. Hence, we cast refinement at the operational level in terms of refinement at the declarative level. To do this, we define a mapping $\llbracket \cdot \rrbracket^*$ from operational models to declarative models (Definition 13) that preserves the environments that the models can interact harmoniously with.

An essential feature of operational refinement is that the mapping from operational to declarative models preserves the safe traces of the component. For a declarative specification \mathcal{P} , a trace t is said to be immediately-safe iff t is permissible, but not inconsistent (i.e., t lies within $T_{\mathcal{P}} \setminus F_{\mathcal{P}}$). If t is contained within $T_{\mathcal{P}} \setminus F_{\mathcal{E}(\mathcal{P})}$, we say that t is safe. The calculation of the safe traces for a Logic IOLTS is slightly more involved, because it is necessary to deal with non-determinism and τ -transitions.

Definition 11 (Immediately-safe states). *The set of immediately-safe states that a process $p_{\mathcal{P}}$ can be in after following the trace t is given by $h_{p_{\mathcal{P}}}(t)$, where $h_{p_{\mathcal{P}}} : \mathcal{A}_{\mathcal{P}}^* \longrightarrow 2^{S_{\mathcal{P}}}$ is defined as:*

$$\begin{aligned}
 - h_{p_P}(\epsilon) &= \begin{cases} \emptyset & \text{if } p_P \xrightarrow{\epsilon}_P p' \text{ with } p' \in F_P \\ \{p' \in S_P : p \xrightarrow{\epsilon}_P p'\} & \text{otherwise} \end{cases} \\
 - h_{p_P}(to) &= \begin{cases} \emptyset & \text{if } \exists p' \in h_{p_P}(t) \text{ such that } p' \xrightarrow{-o}_P p'' \text{ with } p'' \in F_P \\ \{p'' \in S_P : \exists p' \in h_{p_P}(t) \cdot p' \xrightarrow{-o}_P p''\} & \text{otherwise} \end{cases} \\
 &\text{when } o \in \mathcal{A}_P^O \\
 - h_{p_P}(ti) &= \begin{cases} \emptyset & \text{if } \exists p' \in h_{p_P}(t) \text{ such that } p' \xrightarrow{-i}_P p'' \text{ with } p'' \in F_P, \text{ or } p' \not\xrightarrow{i}_P \\ \{p'' \in S_P : \exists p' \in h_{p_P}(t) \cdot p' \xrightarrow{-i}_P p''\} & \text{otherwise} \end{cases} \\
 &\text{when } i \in \mathcal{A}_P^I.
 \end{aligned}$$

Definition 12 (Safe traces). A trace t of p_P is *immediately-safe* iff $h_{p_P}(t) \neq \emptyset$ and is *safe* iff $h_{p_{\mathcal{E}(P)}}(t) \neq \emptyset$, where \mathcal{E} propagates inconsistencies backwards over output and τ transitions. The set of immediately-safe traces of p_P is denoted $IST(p_P)$, while the set of safe traces is denoted $ST(p_P)$.

An immediately-safe trace t of a process p characterises a permissible exchange between p and an arbitrary environment, such that t will never encounter an inconsistent state under any resolution of p 's non-determinism. Relating this intuition to Definitions 11 and 12, suppose p and the environment can safely communicate on the trace t . If from some state that p is in after following t it can perform an output o , and every o it can output will never make the system inconsistent, then the environment must be willing to accept that output. Conversely, the environment can only safely issue an input i after t if i can be accepted from every state the process is in after following t , without making the system inconsistent. We must impose these restrictions to account for the fact that the process cannot be expected to know how to resolve its non-determinism prior to its communication with the environment.

Definition 13 (Model mapping). The model mapping function $\llbracket \cdot \rrbracket^*$ from Logic IOLTSs to declarative specifications is defined by $\llbracket p_P \rrbracket^* = \langle \mathcal{A}_P^I, \mathcal{A}_P^O, T_{\llbracket p_P \rrbracket^*}, F_{\llbracket p_P \rrbracket^*} \rangle$, where:

$$\begin{aligned}
 - T_{\llbracket p_P \rrbracket^*} &= \{t : p_P \xrightarrow{t}_P\} \cup F \cup FI \\
 - F_{\llbracket p_P \rrbracket^*} &= F \cup FI \\
 - F &= \{tt' : p_P \xrightarrow{t}_P p', p' \in F_P \text{ and } t' \in \mathcal{A}_P^*\} \\
 - FI &= \{tit' : p_P \xrightarrow{t}_P p', i \in \mathcal{A}_P^I, p' \not\xrightarrow{i}_P \text{ and } t' \in \mathcal{A}_P^*\}.
 \end{aligned}$$

Theorem 7 (Model mapping preserves safe traces). For an arbitrary process p_P , $IST(p_P) = T_{\llbracket p_P \rrbracket^*} \setminus F_{\llbracket p_P \rrbracket^*}$ and $ST(p_P) = T_{\llbracket p_P \rrbracket^*} \setminus F_{\mathcal{E}(\llbracket p_P \rrbracket^*)}$.

Having defined a mapping from operational to declarative models, we can now define operational refinement in the obvious way.

Definition 14 (Operational refinement). Process q_Q is said to be a refinement of process p_P , written $q_Q \sqsubseteq_{op} p_P$, iff $\llbracket q_Q \rrbracket^* \sqsubseteq_{dec} \llbracket p_P \rrbracket^*$.

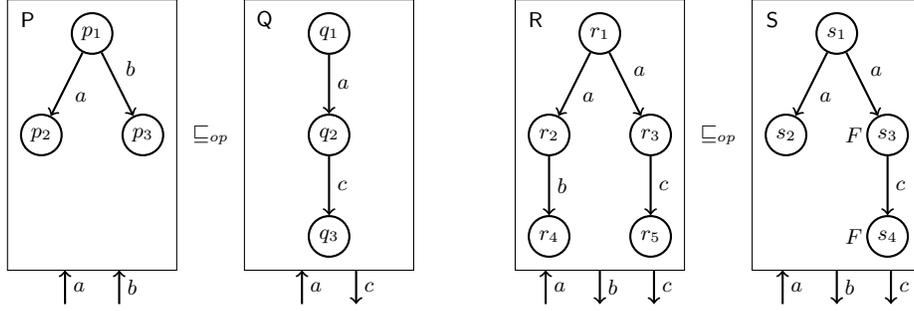


Fig. 1. Refinement of Logic IOLTSs.

Lemma 2 (Operational preorder). *Refinement is reflexive and transitive.*

Under the assumption of finiteness, we note that refinement checking is PSPACE-complete. This is similar to traces refinement in CSP, where the worst-case is rarely observed in practice.

Definition 15 (Operational equivalence). *Processes p_P and q_Q are said to be equivalent, written $p_P \equiv_{op} q_Q$, iff $q_Q \sqsubseteq_{op} p_P$ and $p_P \sqsubseteq_{op} q_Q$.*

Looking at the refinements in Figure 1, from q_1 the environment can safely issue a , after which it must be willing to accept c . Clearly a can be safely accepted by p_1 , and as p_2 does not issue a c output the environment will be perfectly happy. Moreover, as the environment is not permitted to issue a b in q_1 there is no harm in p_1 being able to handle this behaviour. Hence $p_1 \sqsubseteq_{op} q_1$. Now, $r_1 \sqsubseteq_{op} s_1$ as r_1 is willing to accept the input a from the environment, which is not the case in s_1 . This is because we cannot trust s_1 to resolve its non-determinism on a in an optimistic way by always moving to s_2 .

Example 2. To formally check $p_1 \sqsubseteq_{op} q_1$, it is necessary to resort to the definition of refinement on declarative specifications (Definition 4). It can easily be checked that all of the conditions of that definition hold by considering the sets below, obtained by computing the model mapping of the processes p_1 and q_1 .

$$\begin{aligned}
& - F_{\llbracket p_1 \rrbracket^*} = F_{\mathcal{E}(\llbracket p_1 \rrbracket^*)} = (a + b)(a + b)^+ \\
& - F_{\llbracket q_1 \rrbracket^*} = F_{\mathcal{E}(\llbracket q_1 \rrbracket^*)} = (a + ac)a(a + c)^* \\
& - T_{\llbracket p_1 \rrbracket^*} = (a + b)^* \\
& - T_{\llbracket q_1 \rrbracket^*} = \{\epsilon, a, ac\} \cup (a + ac)a(a + c)^* \\
& - X \uparrow \mathcal{A}_P^I = X \cup (\epsilon + a + ac + (aa + aca)(a + c)^*)b(a + b + c)^* \text{ for } X \in \\
& \quad \{F_{\llbracket q_1 \rrbracket^*}, T_{\llbracket q_1 \rrbracket^*}\}.
\end{aligned}$$

3.2 Error-completion

In order to simplify the definitions of the operators in our specification theory for the operational framework, we introduce the error-completion of a Logic

IOLTS. This is a transformation that leaves the mapping from a Logic IOLTS to a declarative specification unchanged.

The error-completion of a Logic IOLTS provides an explicit operational representation for the inconsistent traces that would arise in mapping the Logic IOLTS to its corresponding declarative specification. Consequently, an error-completed Logic IOLTS is closed under input extensions. It is this property that simplifies the definitions of the operators in our framework. We do not say that an error-completed Logic IOLTS is input-enabled, however, as we can distinguish good inputs from bad inputs.

Definition 16 (Error-completion). *Let P be a Logic IOLTS, and assume $f_P \notin S_P$. The error-completion of P is a Logic IOLTS $P_\perp = \langle S_{P_\perp}, \mathcal{A}_P^I, \mathcal{A}_P^O, \longrightarrow_{P_\perp}, F_{P_\perp} \rangle$, where:*

- $S_{P_\perp} = S_P \cup \{f_P\}$
- $\longrightarrow_{P_\perp} = \longrightarrow_P \cup \{(f, a, f) : f \in F_{P_\perp} \text{ and } a \in \mathcal{A}_P\} \cup \{(s, a, f_P) : a \in \mathcal{A}_P^I \text{ and } \nexists s' \cdot s \xrightarrow{a}_P s'\}$
- $F_{P_\perp} = F_P \cup \{f_P\}$.

As remarked, the error-completion of a Logic IOLTS preserves the mapping from Logic IOLTSs to declarative specifications, as the next lemma shows. Note that the corresponding declarative specifications are equal, rather than declaratively equivalent.

Lemma 3 (Error-completion respects mappings). *For any process p_P , $\llbracket p_P \rrbracket^* = \llbracket p_{P_\perp} \rrbracket^*$.*

Besides simplifying the definition of the compositional operators in our specification theory, error-completion of a Logic IOLTS also simplifies the definition of the model mapping function.

Lemma 4 (Simplified model mapping). *Let p be a process in Logic IOLTS P_\perp . Then $\llbracket p \rrbracket^* = \langle \mathcal{A}_P^I, \mathcal{A}_P^O, T_{\llbracket p \rrbracket^*}, F_{\llbracket p \rrbracket^*} \rangle$, where:*

- $T_{\llbracket p \rrbracket^*} = \{t : p \xRightarrow{t}_{P_\perp}\}$
- $F_{\llbracket p \rrbracket^*} = \{tt' : p \xRightarrow{t}_{P_\perp} p' \xRightarrow{t'}_{P_\perp} \text{ and } p' \in F_{P_\perp}\}$.

3.3 Parallel composition

As for declarative specifications, the parallel composition of Logic IOLTSs yields a Logic IOLTS representing the combined effect of its operands running asynchronously. We insist that any given output should be under the control of one component only. Therefore Logic IOLTSs P and Q are composable for parallel composition only if $\mathcal{A}_P^O \cap \mathcal{A}_Q^O = \emptyset$.

Definition 17 (Parallel composition). *Let P and Q be Logic IOLTSs composable for parallel composition. Then the parallel composition of P and Q is a Logic IOLTS $P \parallel Q = \langle S, \mathcal{A}^I, \mathcal{A}^O, \longrightarrow, F \rangle$, where:*

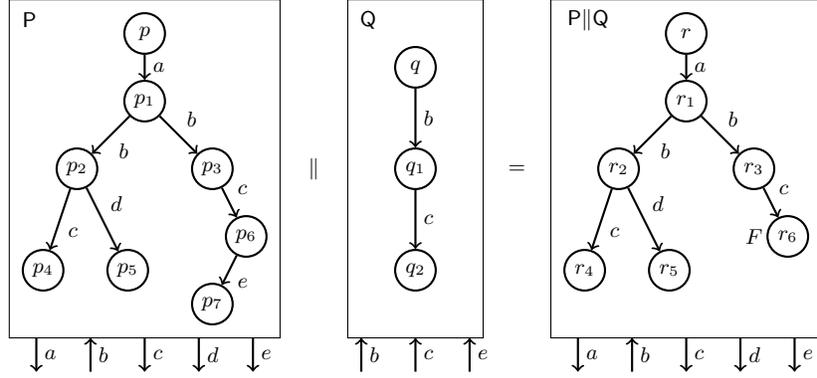


Fig. 2. Example of parallel composition on Logic IOLTSSs.

- $S = S_{P_{\perp}} \parallel S_{Q_{\perp}}$
- $\mathcal{A}^I = (\mathcal{A}_P^I \cup \mathcal{A}_Q^I) \setminus (\mathcal{A}_P^O \cup \mathcal{A}_Q^O)$
- $\mathcal{A}^O = \mathcal{A}_P^O \cup \mathcal{A}_Q^O$
- \longrightarrow is the smallest relation satisfying the following rules:
 - P1. If $p \xrightarrow{a}_{P_{\perp}} p'$ with $a \in \mathcal{A}_P^I \setminus \mathcal{A}_Q$, then $p \parallel q \xrightarrow{a} p' \parallel q$
 - P2. If $q \xrightarrow{a}_{Q_{\perp}} q'$ with $a \in \mathcal{A}_Q^I \setminus \mathcal{A}_P$, then $p \parallel q \xrightarrow{a} p \parallel q'$
 - P3. If $p \xrightarrow{a}_{P_{\perp}} p'$ and $q \xrightarrow{a}_{Q_{\perp}} q'$ with $a \in \mathcal{A}_P \cap \mathcal{A}_Q$, then $p \parallel q \xrightarrow{a} p' \parallel q'$.
- $F = (S_{P_{\perp}} \parallel F_{Q_{\perp}}) \cup (F_{P_{\perp}} \parallel S_{Q_{\perp}})$.

Conditions P1 to P3 ensure that the parallel composition of Logic IOLTSSs interleave on independent actions and synchronise on common actions. For P3, given the parallel composability constraint, synchronisation can take place between an output and an input, or two inputs. Figure 2 shows how the parallel composition operator works in practice, although we omit non-enabled input transitions to inconsistent states. In particular, the example demonstrates how inconsistencies can be introduced through non-input enabledness, as in state r_6 corresponding to $p_6 \parallel q_2$.

Reassuringly, parallel composition of Logic IOLTSSs yields a Logic IOLTSS. The following theorem shows the relationship between parallel composition on Logic IOLTSSs and parallel composition on declarative specifications.

Theorem 8 (Parallel correspondences). *Let P and Q be Logic IOLTSSs composable for parallel composition. For processes p_P and q_Q , it holds that $\llbracket p_P \parallel q_Q \rrbracket^* = \llbracket p_P \rrbracket^* \parallel \llbracket q_Q \rrbracket^*$.*

3.4 Conjunction

In keeping with conjunction of declarative specifications, Logic IOLTSSs P and Q are composable for conjunction only if the sets $\mathcal{A}_P^I \cup \mathcal{A}_Q^I$ and $\mathcal{A}_P^O \cup \mathcal{A}_Q^O$ are disjoint.

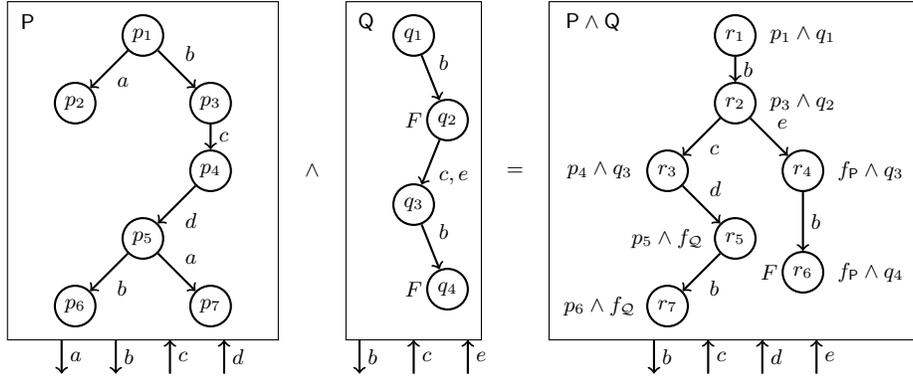


Fig. 3. Example of conjunction on Logic IOLTSSs.

Definition 18 (Conjunction). Let P and Q be Logic IOLTSSs composable for conjunction. Then the conjunction of P and Q is a Logic IOLTSS $P \wedge Q = \langle S, \mathcal{A}_P^I \cup \mathcal{A}_Q^I, \mathcal{A}_P^O \cap \mathcal{A}_Q^O, \longrightarrow, F \rangle$, where:

- $S = S_{P_\perp} \wedge S_{Q_\perp}$
- \longrightarrow is the smallest relation satisfying the following rules:
 - C1. If $a \in \mathcal{A}_P^O \cap \mathcal{A}_Q^O$, $p \xrightarrow{a}_{P_\perp} p'$ and $q \xrightarrow{a}_{Q_\perp} q'$, then $p \wedge q \xrightarrow{a} p' \wedge q'$
 - C2. If $a \in \mathcal{A}_P^I \cap \mathcal{A}_Q^I$, $p \xrightarrow{a}_{P_\perp} p'$ and $q \xrightarrow{a}_{Q_\perp} q'$, then $p \wedge q \xrightarrow{a} p' \wedge q'$
 - C3. If $a \in \mathcal{A}_P^I \setminus \mathcal{A}_Q^I$ and $p \xrightarrow{a}_{P_\perp} p'$, then $p \wedge q \xrightarrow{a} p' \wedge f_Q$
 - C4. If $a \in \mathcal{A}_Q^I \setminus \mathcal{A}_P^I$ and $q \xrightarrow{a}_{Q_\perp} q'$, then $p \wedge q \xrightarrow{a} f_P \wedge q'$
 - C5. If $p \xrightarrow{\tau}_{P_\perp} p'$, then $p \wedge q \xrightarrow{\tau} p' \wedge q$
 - C6. If $q \xrightarrow{\tau}_{Q_\perp} q'$, then $p \wedge q \xrightarrow{\tau} p \wedge q'$
- $F = F_{P_\perp} \wedge F_{Q_\perp}$.

The idea behind the definition of conjunction for $p \wedge q$ is that p and q must synchronise on common actions, interleave on τ -transitions, and on encountering independent input actions behave like the respective component to which the action belongs. On encountering a state $p \wedge q$ in which one of $p \in F_P$ or $q \in F_Q$ holds, let it be p , we know that whatever the behaviour of $p \wedge q$ it will always be a refinement of p . So the most general refinement of $p \wedge q$ will actually be q . This is supported by the fact that inconsistent states in the error-completed Logic IOLTSS admit arbitrary behaviour.

Figure 3 shows the conjunction of processes p_1 and q_1 in the Logic IOLTSSs P and Q (although for clarity we omit inputs leading to inconsistent states). In state r_1 corresponding to $p_1 \wedge q_1$, the b -output transitions of p_1 and q_1 synchronise. Independent output actions such as the a -transition in p_1 are not permitted to proceed, because it would not be the case that r_1 could be used safely in place of q_1 if this transition were to be permitted. State r_2 can evolve into r_3 by synchronising the c -inputs of p_3 and q_2 , while it can also evolve into r_4 by

proceeding on the independent input e of q_2 . From this point, r_4 behaves like q_3 , because e is an input-violation of p_3 . Similar reasoning applies to r_3 's evolution into r_5 by receiving the d -input.

As for parallel composition, there is a correspondence between conjunction at the operational and declarative levels.

Theorem 9 (Conjunction correspondences). *Let P and Q be Logic IOLTSs composable for conjunction. For processes p_P and q_Q , it holds that $\llbracket p_P \wedge q_Q \rrbracket^* = \llbracket p_P \rrbracket^* \wedge \llbracket q_Q \rrbracket^*$.*

3.5 Quotient

Non-determinism and τ -transitions arising in Logic IOLTSs make the definition of quotient more involved than the other operators we have considered on operational models. To ensure that the quotient is the coarsest specification, it is necessary to track the non-determinism of the system-wide specification and its partial implementation. This is because the non-determinism can affect the safe traces of a Logic IOLTS.

As for declarative specifications, we only compute the quotient of process p_P from r_R when $\mathcal{A}_P^O \subseteq \mathcal{A}_R^O$ and $\mathcal{A}_P \subseteq \mathcal{A}_R$. The quotient is the coarsest specification q over an interface consisting of inputs $\mathcal{A}_P^O \cup \mathcal{A}_R^I$ and outputs $\mathcal{A}_R^O \setminus \mathcal{A}_P^O$ such that $p_P \parallel q \sqsubseteq_{op} r_R$. If such a q exists, we denote it by r_R/p_P .

Before defining the quotient-construction, we introduce some functions and predicates that simplify the presentation.

Definition 19. *For Logic IOLTS P , set of states $S \subseteq S_{P_\perp}$ and action $a \in \mathcal{A}_P$, define:*

- $\text{succ}_P^\epsilon(S) = \{s' : s \xrightarrow{\epsilon}_{P_\perp} s' \text{ with } s \in S\}$
- $\text{succ}_P^a(S) = \{s' : s \xrightarrow{a}_{P_\perp} s' \text{ with } s \in S\}$.

Definition 20 (Quotient Logic IOLTS). *Let P and R be Logic IOLTSs such that $\mathcal{A}_P^O \subseteq \mathcal{A}_R^O$ and $\mathcal{A}_P \subseteq \mathcal{A}_R$. The quotient of P from R is the Logic IOLTS $R/P = \langle S_{R/P}, \mathcal{A}_{R/P}^I, \mathcal{A}_{R/P}^O, \longrightarrow, F_{R/P} \rangle$, where:*

- $S_{R/P} = \{R/P : R \subseteq S_{R_\perp} \text{ and } P \subseteq S_{P_\perp}\}$
- $\mathcal{A}_{R/P}^I = \mathcal{A}_P^I \cup \mathcal{A}_R^I$
- $\mathcal{A}_{R/P}^O = \mathcal{A}_R^O \setminus \mathcal{A}_P^O$
- \longrightarrow is the smallest relation satisfying the following rules:
 - Q1. $R'/P' \xrightarrow{a} \text{succ}_R^a(R')/\text{succ}_P^a(P')$ providing:
 - (a) $a \in \mathcal{A}_P^I \cap \mathcal{A}_R^I$ implies $\text{succ}_R^a(R') \cap F_{\mathcal{E}(R)} = \emptyset$
 - (b) $a \in \mathcal{A}_P^O \cap \mathcal{A}_R^O$ implies $\text{succ}_R^a(R') \cap F_{\mathcal{E}(R)} = \emptyset$ and $\text{succ}_P^a(P') \neq \emptyset$
 - (c) $a \in \mathcal{A}_P^I \cap \mathcal{A}_R^O$ implies $\text{succ}_P^a(P') \cap F_P = \emptyset$ and $\text{succ}_R^a(R') \neq \emptyset$
 - Q2. $R'/P' \xrightarrow{a} \text{succ}_R^a(R')/P'$ providing:
 - (a) $a \in \mathcal{A}_R^I \setminus \mathcal{A}_P^I$ implies $\text{succ}_R^a(R') \cap F_{\mathcal{E}(R)} = \emptyset$
 - (b) $a \in \mathcal{A}_R^O \setminus \mathcal{A}_P^O$ implies $\text{succ}_R^a(R') \neq \emptyset$.

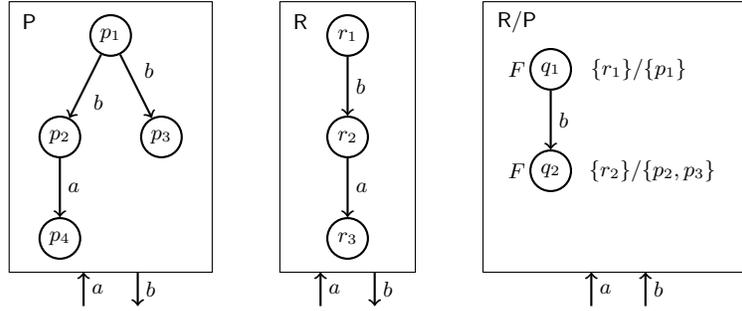


Fig. 4. Example showing non-existence of quotient on Logic IOLTSs.

- $R'/P' \in F_{R/P}$ iff at least one of the following rules holds:
 - F1. $R' = \emptyset$ or $P' = \emptyset$
 - F2. $F_{\mathcal{E}(R_\perp)} \cap R' \neq \emptyset$ or $F_{P_\perp} \cap P' \neq \emptyset$
 - F3. $R'/P' \xrightarrow{a} R''/P''$ with $a \in \mathcal{A}_{R/P}^I$ and $R''/P'' \in F_{R/P}$.

Definition 21 (Quotient). Let P and R be Logic IOLTSs such that $\mathcal{A}_P^O \subseteq \mathcal{A}_R^O$ and $\mathcal{A}_P \subseteq \mathcal{A}_R$. The quotient of process p_P from process r_R , written r_R/p_P , is the process $\text{succ}_R^\varepsilon(r_R)/\text{succ}_P^\varepsilon(p_P)$ in the Logic IOLTS R/P obtained from R/P by removing all transitions immediately leading to a state in $F_{R/P}$, and removing all states R/P such that $R/P \in F_{R/P}$ and $R \notin F_{\mathcal{E}(R)}$. If $\text{succ}_R^\varepsilon(r_R)/\text{succ}_P^\varepsilon(p_P)$ is not contained in R/P , then the quotient is not defined.

As for declarative specifications, the quotient of p_P from r_R may not exist. The following theorem shows that definedness of the quotient according to the previous definition coincides precisely with the existence of such a quotient.

Theorem 10 (Existence of quotient). Let P and R be Logic IOLTSs such that $\mathcal{A}_P^O \subseteq \mathcal{A}_R^O$ and $\mathcal{A}_P \subseteq \mathcal{A}_R$. Then r_R/p_P is defined (i.e. $r_R \in F_{\mathcal{E}(R)}$ or $r_R/p_P \notin F_{R/P}$) iff there exists a process q in a Logic IOLTS with inputs $\mathcal{A}_{R/P}^I$ and outputs $\mathcal{A}_{R/P}^O$ such that $p_P \parallel q \sqsubseteq_{op} r_R$.

Consequently, the constraint $r_R \notin F_{\mathcal{E}(R)}$ and $r_R/p_P \in F_{R/P}$ gives a precise characterisation of whether the quotient exists or not. When the quotient does exist, it behaves in exactly the same way as for declarative specifications.

Theorem 11 (Quotient correspondences). Let P and R be Logic IOLTSs such that $\mathcal{A}_P^O \subseteq \mathcal{A}_R^O$ and $\mathcal{A}_P \subseteq \mathcal{A}_R$. If $r_R/p_P \notin F_{R/P}$ or $r_R \in F_{\mathcal{E}(R)}$, then $\llbracket r_R/p_P \rrbracket^* = \llbracket r_R \rrbracket^* / \llbracket p_P \rrbracket^*$.

Figure 4 provides an example in which processes p_1 and r_1 have no quotient. This tallies with Theorem 10, as we have $\{r_1\}/\{p_1\} \in F_{R/P}$ when $r_1 \notin F_{\mathcal{E}(R)}$. On the other hand, quotients exist for the processes p_1 and r_1 of Figures 5 and 6. This is also supported by Theorem 10, as $\{r_1\}/\{p_1\} \notin F_{R/P}$ for the processes in both figures.

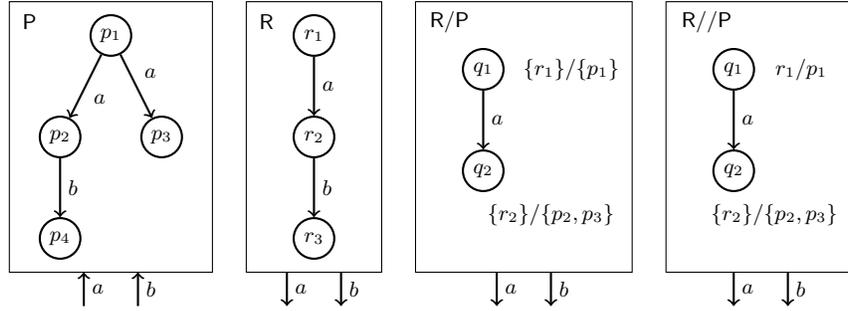


Fig. 5. Example of quotient on Logic IOLTSs with no inconsistencies.

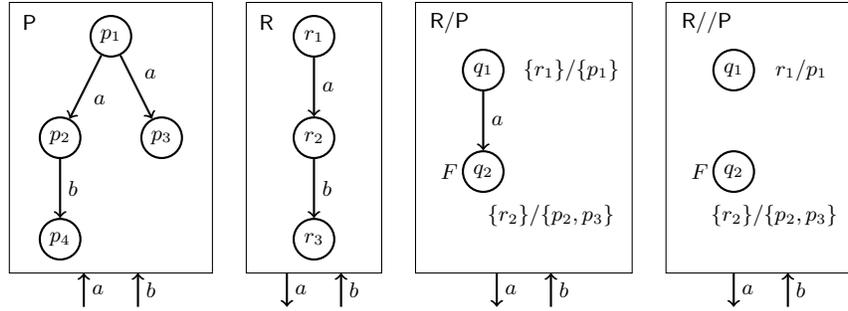


Fig. 6. Example of quotient on Logic IOLTSs with inconsistencies.

For Figure 6, the quotient is the single consistent state $\{r_1\}/\{p_1\}$. This is because in going from R/P to R//P we remove the transition labelled by the output a between the processes $\{r_1\}/\{p_1\}$ and $\{r_2\}/\{p_2, p_3\}$, as the latter state is inconsistent. Maintaining this transition would yield an invalid quotient as $p_1 \parallel (r_1/p_1)$ would be inconsistent when r_1 is consistent. It is safe to discard this transition only because it is an output. Recalling the definition of \sqsubseteq_{op} , for safe-substitutivity it is perfectly safe to suppress outputs on the left that would have occurred on the right.

4 Full-Abstraction Results

In this section we present full-abstraction results that relate our declarative and operational equivalences based on trace containment to a simple equivalence, which ensures that an inconsistent process must have an inconsistent specification. The result is shown by employing a testing scenario where processes are placed in parallel with an arbitrary composable process in order to establish their equivalence with regard to the observation of consistency.

Definition 22 (Declarative inconsistency equivalence). *Let \mathcal{P} and \mathcal{Q} be declarative specifications. Declarative inconsistency equivalence, denoted by \equiv_{dec}^F , is given by $\mathcal{P} \equiv_{dec}^F \mathcal{Q}$ iff $\mathcal{A}_{\mathcal{P}}^I = \mathcal{A}_{\mathcal{Q}}^I$, $\mathcal{A}_{\mathcal{P}}^O = \mathcal{A}_{\mathcal{Q}}^O$ and $\epsilon \in F_{\mathcal{P}} \iff \epsilon \in F_{\mathcal{Q}}$.*

Declarative equivalence can be established by placing each process in parallel with arbitrary composable tester processes and observing whether the simple inconsistency equivalence is maintained.

Theorem 12. *Let \mathcal{P} and \mathcal{Q} be declarative specifications. Then:*

$$\mathcal{P} \equiv_{dec} \mathcal{Q} \text{ iff } \forall \mathcal{R} \cdot \mathcal{A}_{\mathcal{R}}^O \cap (\mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{Q}}^O) = \emptyset \implies \mathcal{E}(\mathcal{P} \parallel \mathcal{R}) \equiv_{dec}^F \mathcal{E}(\mathcal{Q} \parallel \mathcal{R}).$$

From this characterisation of \equiv_{dec} , we obtain a full-abstraction result with respect to parallel composition and \equiv_{dec}^F . Our definition of full-abstraction is taken from [19] (Definition 16), which means that \equiv_{dec} is the coarsest congruence with respect to the operators of our specification theory and \equiv_{dec}^F .

Corollary 1 (Declarative full-abstraction). *Declarative equivalence \equiv_{dec} is fully-abstract with respect to parallel, conjunction, quotient and \equiv_{dec}^F .*

We can now present analogous results for our operational models.

Definition 23 (Operational inconsistency equivalence). *Let $p_{\mathcal{P}}$ and $q_{\mathcal{Q}}$ be processes of Logic IOLTSs \mathcal{P} and \mathcal{Q} . Operational inconsistency equivalence, denoted by \equiv_{op}^F , is given by $p_{\mathcal{P}} \equiv_{op}^F q_{\mathcal{Q}}$ iff $\mathcal{A}_{\mathcal{P}}^I = \mathcal{A}_{\mathcal{Q}}^I$, $\mathcal{A}_{\mathcal{P}}^O = \mathcal{A}_{\mathcal{Q}}^O$ and $p_{\mathcal{P}} \in F_{\mathcal{P}} \iff q_{\mathcal{Q}} \in F_{\mathcal{Q}}$.*

Theorem 13. *Let $p_{\mathcal{P}}$ and $q_{\mathcal{Q}}$ be processes of Logic IOLTSs \mathcal{P} and \mathcal{Q} . Then:*

$$p_{\mathcal{P}} \equiv_{op} q_{\mathcal{Q}} \text{ iff } \forall r_{\mathcal{R}} \cdot \mathcal{A}_{\mathcal{R}}^O \cap (\mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{Q}}^O) = \emptyset \implies \mathcal{E}(p_{\mathcal{P}} \parallel r_{\mathcal{R}}) \equiv_{op}^F \mathcal{E}(q_{\mathcal{Q}} \parallel r_{\mathcal{R}}),$$

where \mathcal{E} applied to processes in Logic IOLTSs propagates the inconsistency predicate backwards over all output and τ labelled transitions.

Corollary 2 (Operational full-abstraction). *Operational equivalence \equiv_{op} is fully-abstract with respect to parallel, conjunction, quotient and \equiv_{op}^F .*

5 Conclusion and Future Work

We have developed a compositional specification theory for components that may be modelled operationally, closely mirroring actual implementations, or in an abstract manner by means of declarative specifications. Both frameworks admit a simple refinement relation, defined in terms of traces, which corresponds to safe-substitutivity. We define asynchronous parallel composition, conjunction and quotient, and prove that the induced equivalence is a congruence for these operations. It is straightforward to extend our framework with disjunction and hiding. The simplicity of our formalism facilitates reasoning about the temporal ordering of interactions needed for assume-guarantee inference. Although not considered in this paper, our framework supports reasoning about safety properties in the context of assume-guarantee. Liveness properties may also be considered, but this requires the introduction of quiescence or infinite behaviours, the latter being achieved with the help of ω -automata techniques.

Acknowledgments. The authors are supported by EU FP7 project CONNECT and ERC Advanced Grant VERIWARE. We would also like to thank the anonymous reviewers for their insightful comments.

References

1. Chen, T., Chilton, C., Jonsson, B., Kwiatkowska, M.: A Compositional Specification Theory for Component Behaviours. In Seidl, H., ed.: *Programming Languages and Systems, Proc. 21st European Symposium on Programming (ESOP'12)*. Volume 7211 of *Lecture Notes in Computer Science.*, Springer-Verlag (2012) 148–168
2. Larsen, K.G., Nyman, U., Wasowski, A.: Modal I/O automata for interface and product line theories. In Nicola, R.D., ed.: *ESOP. Volume 4421 of LNCS.*, Springer (2007) 64–79
3. de Alfaro, L., Henzinger, T.A.: Interface automata. *SIGSOFT Softw. Eng. Notes* **26** (2001) 109–120
4. Raclet, J.B., Badouel, E., Benveniste, A., Caillaud, B., Passerone, R.: Why are modalities good for Interface Theories? In: *Proc. 9th International Conference on Application of Concurrency to System Design. ACS D '09, IEEE Computer Society* (2009) 119–127
5. Raclet, J.B., Badouel, E., Benveniste, A., Caillaud, B., Legay, A., Passerone, R.: Modal Interfaces: Unifying Interface Automata and Modal Specifications. In: *Proc. 7th International Conference on Embedded Software. EMSOFT '09, ACM* (2009) 87–96
6. Lynch, N.A., Tuttle, M.R.: An introduction to input/output automata. *CWI Quarterly* **2** (1989) 219–246
7. Lüttgen, G., Vogler, W.: Conjunction on processes: Full abstraction via ready-tree semantics. *Theor. Comput. Sci.* **373** (2007) 19–40
8. Lüttgen, G., Vogler, W.: Ready simulation for concurrency: It's logical! *Inf. Comput.* **208** (2010) 845–867
9. Brookes, S.D., Hoare, C.A.R., Roscoe, A.W.: A theory of communicating sequential processes. *J. ACM* **31** (1984) 560–599
10. de Alfaro, L., Henzinger, T.A.: Interface-based design. In Broy, M., Grünbauer, J., Harel, D., Hoare, T., eds.: *Engineering Theories of Software Intensive Systems. Volume 195 of NATO Science Series II: Mathematics, Physics and Chemistry.* Springer (2005) 83–104
11. de Alfaro, L.: Game models for open systems. In Dershowitz, N., ed.: *Verification: Theory and Practice. Volume 2772 of LNCS.* Springer (2004) 192–193
12. Doyen, L., Henzinger, T.A., Jobstmann, B., Petrov, T.: Interface theories with component reuse. In: *Proc. 8th ACM international conference on Embedded software. EMSOFT '08, ACM* (2008) 79–88
13. Bhaduri, P., Ramesh, S.: Interface synthesis and protocol conversion. *Form. Asp. Comput.* **20** (2008) 205–224
14. Nicola, R.D., Segala, R.: A process algebraic view of input/output automata. *Theor. Comput. Sci.* **138** (1995) 391–423
15. Tretmans, J.: Model-based testing and some steps towards test-based modelling. In Bernardo, M., Issarny, V., eds.: *SFM. Volume 6659 of LNCS.*, Springer (2011) 297–326
16. Josephs, M.B.: Receptive process theory. *Acta Inf.* **29** (1992) 17–31
17. Josephs, M.B., Kapoor, H.K.: Controllable delay-insensitive processes. *Fundam. Inf.* **78** (2007) 101–130
18. Raclet, J.B.: Residual for component specifications. *Electr. Notes Theor. Comput. Sci.* **215** (2008) 93–110
19. Glabbeek, R.J.v.: Full abstraction in structural operational semantics (extended abstract). In: *Proc. 3rd Intl Conference on Methodology and Software Technology: Algebraic Methodology and Software Technology, Springer* (1994) 75–82

A Proofs for Declarative Specification Theory

A.1 Refinement

Proof 1 (Lemma 1) For reflexivity, note that $\mathcal{A}_{\mathcal{P}}^I \subseteq \mathcal{A}_{\mathcal{P}}^I$, $\mathcal{A}_{\mathcal{P}}^O \subseteq \mathcal{A}_{\mathcal{P}}^O$, $\mathcal{P} \uparrow \mathcal{A}_{\mathcal{P}}^I = \mathcal{P}$ and $\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{P}}^I = \mathcal{E}(\mathcal{P})$. Hence $T_{\mathcal{P}} = T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{P}}^I}$ and $F_{\mathcal{E}(\mathcal{P})} = F_{\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{P}}^I}$. Thus $\mathcal{P} \sqsubseteq_{dec} \mathcal{P}$.

For transitivity, assume $\mathcal{R} \sqsubseteq_{dec} \mathcal{Q}$ and $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$. Considering the alphabets, $\mathcal{A}_{\mathcal{P}}^I \subseteq \mathcal{A}_{\mathcal{R}}^I$ and $\mathcal{A}_{\mathcal{P}}^I \subseteq \mathcal{A}_{\mathcal{Q}}^I$ implies $\mathcal{A}_{\mathcal{P}}^I \subseteq \mathcal{A}_{\mathcal{R}}^I$, while $\mathcal{A}_{\mathcal{Q}}^O \subseteq \mathcal{A}_{\mathcal{P}}^O$ and $\mathcal{A}_{\mathcal{R}}^O \subseteq \mathcal{A}_{\mathcal{Q}}^O$ implies $\mathcal{A}_{\mathcal{R}}^O \subseteq \mathcal{A}_{\mathcal{P}}^O$. Looking to the permissible traces, we know that $T_{\mathcal{Q}} \subseteq T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$ and $T_{\mathcal{R}} \subseteq T_{\mathcal{Q} \uparrow \mathcal{A}_{\mathcal{R}}^I}$. Let $t \in T_{\mathcal{R}}$. Then $t \equiv t't''$ where $t' \in T_{\mathcal{Q}}$ and $t'' \in (\mathcal{A}_{\mathcal{R}}^I \cup \mathcal{A}_{\mathcal{Q}}^O)^*$. It therefore follows that $t' \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$, meaning that $t' \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{R}}^I}$. Moreover, as $\mathcal{A}_{\mathcal{Q}} \cup \mathcal{A}_{\mathcal{R}}^I \subseteq \mathcal{A}_{\mathcal{P}} \cup \mathcal{A}_{\mathcal{R}}^I$, we know $t'' \in (\mathcal{A}_{\mathcal{P}} \cup \mathcal{A}_{\mathcal{R}}^I)^*$. Thus $t \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{R}}^I}$ and so $T_{\mathcal{R}} \subseteq T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{R}}^I}$. The same argument can be used to show that $F_{\mathcal{E}(\mathcal{R})} \subseteq F_{\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{R}}^I}$, and so $\mathcal{R} \sqsubseteq_{dec} \mathcal{P}$.

A.2 Parallel Composition

Proof 2 (Theorem 1) Let $t \in T_{\mathcal{Q} \parallel \mathcal{R}}$. Then by the definition of \parallel it is known that $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{Q}}$ and $t \upharpoonright \mathcal{A}_{\mathcal{R}} \in T_{\mathcal{R}}$. Now because $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$, it follows that $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$. Moreover, by Lemma 5 we see $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$, and it can easily be shown that $t \upharpoonright \mathcal{A}_{\mathcal{R}} \in T_{\mathcal{R} \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$.

By mutual exclusion, either $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$ or $t \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$. Assume $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$. Then by Lemma 6, we see that $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$. From Lemma 7 it follows that $t \in T_{\mathcal{P} \parallel \mathcal{R} \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$, as required.

Instead suppose $t \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$. Then there exists $t'a$ a prefix of t such that $t' \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t'a \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$. As $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$, it follows that $a \notin \mathcal{A}_{\mathcal{Q}}$. Therefore $a \in \mathcal{A}_{\mathcal{P}}^O$, and moreover, as a occurs in t , it follows that $a \in \mathcal{A}_{\mathcal{R}}^I$. Combining $t' \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ with $t' \upharpoonright \mathcal{A}_{\mathcal{R}} \in T_{\mathcal{R}}$ allows us to see that $t' \in T_{\mathcal{P} \parallel \mathcal{R}}$. Now $a \in \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I \setminus \mathcal{A}_{\mathcal{P} \parallel \mathcal{R}}^I$, and so $t'a \in T_{(\mathcal{P} \parallel \mathcal{R}) \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$. The remaining extension of $t'a$ to t is a word over $(\mathcal{A}_{\mathcal{Q}} \cup \mathcal{A}_{\mathcal{R}})^*$, which is also a word over $(\mathcal{A}_{\mathcal{P}} \cup \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I \cup \mathcal{A}_{\mathcal{R}})^*$. Hence $t \in T_{(\mathcal{P} \parallel \mathcal{R}) \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$.

Showing that $F_{\mathcal{E}(\mathcal{Q} \parallel \mathcal{R})} \subseteq F_{\mathcal{E}((\mathcal{P} \parallel \mathcal{R}) \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I)}$ is similar.

Lemma 5. Let \mathcal{P} , \mathcal{Q} and \mathcal{R} be declarative specifications such that $\mathcal{A}_{\mathcal{Q}}^I \cap \mathcal{A}_{\mathcal{R}}^O \subseteq \mathcal{A}_{\mathcal{P}}^I \cap \mathcal{A}_{\mathcal{R}}^O$. If $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$, then $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$.

Proof. Assume $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$ and suppose for a contradiction that $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \notin T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$. So there exists $t'a$ a prefix of t such that $t' \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$, but $t'a \upharpoonright \mathcal{A}_{\mathcal{Q}} \notin T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$. It must be the case that $a \in \mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I$, which in turn implies $a \in \mathcal{A}_{\mathcal{R}}^O$. But $\mathcal{A}_{\mathcal{Q}}^I \cap \mathcal{A}_{\mathcal{R}}^O \subseteq \mathcal{A}_{\mathcal{P}}^I \cap \mathcal{A}_{\mathcal{R}}^O$, so $a \in \mathcal{A}_{\mathcal{P}}^I$. But this means that $t'a \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P}}$, which is contradictory. Therefore $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q} \parallel \mathcal{R}}^I}$.

Lemma 6. *Let \mathcal{P} , \mathcal{Q} and \mathcal{R} be declarative specifications such that $\mathcal{A}_{\mathcal{Q}}^I \cap \mathcal{A}_{\mathcal{R}}^O \subseteq \mathcal{A}_{\mathcal{P}}^I \cap \mathcal{A}_{\mathcal{R}}^O$. If $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$ and $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$, then $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$.*

Proof. Assume $t \upharpoonright \mathcal{A}_{\mathcal{Q}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$ and $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$, and suppose for a contradiction that $t \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$. So there exists $t'a$ a prefix of t such that $t' \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$, but $t'a \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$. It must be the case that $a \in \mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{Q}}^I$, which in turn implies $a \in \mathcal{A}_{\mathcal{R}}^O$. But $\mathcal{A}_{\mathcal{Q}}^I \cap \mathcal{A}_{\mathcal{R}}^O \subseteq \mathcal{A}_{\mathcal{P}}^I \cap \mathcal{A}_{\mathcal{R}}^O$, so $a \in \mathcal{A}_{\mathcal{P}}^I$. But this means that $t'a \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$, which is contradictory. Therefore $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$.

Lemma 7. *Let \mathcal{P} , \mathcal{Q} and \mathcal{R} be declarative specifications such that $\mathcal{A}_{\mathcal{Q}}^I \cap \mathcal{A}_{\mathcal{R}}^O \subseteq \mathcal{A}_{\mathcal{P}}^I \cap \mathcal{A}_{\mathcal{R}}^O$. If $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$ and $t \upharpoonright \mathcal{A}_{\mathcal{R}} \in T_{\mathcal{R} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$, then $t \in T_{\mathcal{P} \parallel \mathcal{R} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$.*

Proof. Assume $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$ and $t \upharpoonright \mathcal{A}_{\mathcal{R}} \in T_{\mathcal{R} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$. Then one of the following cases must hold:

1. $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t \upharpoonright \mathcal{A}_{\mathcal{R}} \in T_{\mathcal{R}}$
2. $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}(\mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{P}}^I)(\mathcal{A}_{\mathcal{Q}}^I \cup \mathcal{A}_{\mathcal{P}})^*$
3. $t \upharpoonright \mathcal{A}_{\mathcal{R}} \in T_{\mathcal{R}}(\mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{R}}^I)(\mathcal{A}_{\mathcal{Q}}^I \cup \mathcal{A}_{\mathcal{R}})^*$.

Should case 1 hold, we know that $t \in T_{\mathcal{P} \parallel \mathcal{R}}$, hence $t \in T_{\mathcal{P} \parallel \mathcal{R} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$. For case 2, let $t \equiv t'at''$ such that $t' \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$, $t' \upharpoonright \mathcal{A}_{\mathcal{R}} \in T_{\mathcal{R}}$ and $t'a \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$. Therefore, $t' \in T_{\mathcal{P} \parallel \mathcal{R}}$, $a \in \mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{P}}^I$ and $t'' \in (\mathcal{A}_{\mathcal{Q}}^I \cup \mathcal{A}_{\mathcal{P}})^*$. Claim that $a \in \mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{P} \parallel \mathcal{R}}^I$. Now, $t'' \in (\mathcal{A}_{\mathcal{Q}}^I \cup \mathcal{A}_{\mathcal{P}})^*$ implies $t'' \in (\mathcal{A}_{\mathcal{Q}}^I \cup \mathcal{A}_{\mathcal{P} \parallel \mathcal{R}})^*$. Hence $t'at'' \equiv t \in T_{\mathcal{P} \parallel \mathcal{R} \uparrow \mathcal{A}_{\mathcal{Q}}^I}$. Case 3 is similar.

Proof of claim that $a \in \mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{P} \parallel \mathcal{R}}^I$. Suppose it's not the case, then $a \in (\mathcal{A}_{\mathcal{Q}}^I \setminus \mathcal{A}_{\mathcal{P}}^I) \cap (\mathcal{A}_{\mathcal{P} \parallel \mathcal{R}}^I \setminus \mathcal{A}_{\mathcal{P}}^I)$. Hence $a \in [(\mathcal{A}_{\mathcal{Q}}^I \cup \mathcal{A}_{\mathcal{R}}^I) \setminus (\mathcal{A}_{\mathcal{P}}^I \cup \mathcal{A}_{\mathcal{Q}}^O \cup \mathcal{A}_{\mathcal{R}}^O)] \cap [\mathcal{A}_{\mathcal{R}}^I \setminus (\mathcal{A}_{\mathcal{P}} \cup \mathcal{A}_{\mathcal{R}}^O)]$. But $a \in \mathcal{A}_{\mathcal{P}}$, hence $a \in \emptyset$, which is contradictory.

A.3 Conjunction

Proof 3 (Theorem 2) *The proofs for the properties follow from the following results:*

1. Theorems 14 and 15.
2. Theorems 16 and 15.
3. Theorems 17 and 18.

Theorem 14. *Let \mathcal{P} and \mathcal{Q} be declarative specifications compatible for conjunction. Then $\mathcal{P} \wedge \mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$ and $\mathcal{P} \wedge \mathcal{Q} \sqsubseteq_{dec} \mathcal{Q}$.*

Proof. By symmetry, it is sufficient to show that $\mathcal{P} \wedge \mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$. We only show the more involved case of inconsistent trace inclusion.

$$\begin{aligned}
 & t \in F_{\mathcal{E}(\mathcal{P} \wedge \mathcal{Q})} \\
 \iff & t \in F_{\mathcal{E}(\mathcal{P}) \wedge \mathcal{E}(\mathcal{Q})} \\
 \iff & t \in F_{\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{Q}}^I} \text{ and } t \in F_{\mathcal{E}(\mathcal{Q}) \uparrow \mathcal{A}_{\mathcal{P}}^I} \\
 \implies & t \in F_{\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{Q}}^I} \\
 \iff & t \in F_{\mathcal{E}(\mathcal{P}) \uparrow (\mathcal{A}_{\mathcal{P}}^I \cup \mathcal{A}_{\mathcal{Q}}^I)} \\
 \iff & t \in F_{\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{P} \wedge \mathcal{Q}}^I}
 \end{aligned}$$

Theorem 15. Let \mathcal{P} , \mathcal{Q} and \mathcal{R} be declarative specifications such that \mathcal{P} is compatible with \mathcal{Q} for conjunction. If $\mathcal{R} \sqsubseteq_{dec} \mathcal{P}$ and $\mathcal{R} \sqsubseteq_{dec} \mathcal{Q}$, then $\mathcal{R} \sqsubseteq_{dec} \mathcal{P} \wedge \mathcal{Q}$.

Proof. We only show the more involved case of inconsistent trace inclusion.

$$\begin{aligned}
 & t \in F_{\mathcal{E}(\mathcal{R})} \\
 \implies & t \in F_{\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{R}}^I} \cap F_{\mathcal{E}(\mathcal{Q}) \uparrow \mathcal{A}_{\mathcal{R}}^I} && \mathcal{R} \sqsubseteq_{dec} \mathcal{P} \text{ and } \mathcal{R} \sqsubseteq_{dec} \mathcal{Q} \\
 \implies & t \in F_{(\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{R}}^I) \uparrow \mathcal{A}_{\mathcal{Q}}^I} \cap F_{(\mathcal{E}(\mathcal{Q}) \uparrow \mathcal{A}_{\mathcal{R}}^I) \uparrow \mathcal{A}_{\mathcal{P}}^I} \\
 \iff & t \in F_{(\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{R}}^I) \uparrow (\mathcal{A}_{\mathcal{Q}}^I \cup \mathcal{A}_{\mathcal{R}}^I)} \cap F_{(\mathcal{E}(\mathcal{Q}) \uparrow \mathcal{A}_{\mathcal{R}}^I) \uparrow (\mathcal{A}_{\mathcal{P}}^I \cup \mathcal{A}_{\mathcal{R}}^I)} \\
 \iff & t \in F_{(\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{R}}^I) \wedge (\mathcal{E}(\mathcal{Q}) \uparrow \mathcal{A}_{\mathcal{R}}^I)} \\
 \iff & t \in F_{(\mathcal{E}(\mathcal{P}) \wedge \mathcal{E}(\mathcal{Q})) \uparrow \mathcal{A}_{\mathcal{R}}^I} \\
 \iff & t \in F_{\mathcal{E}(\mathcal{P} \wedge \mathcal{Q}) \uparrow \mathcal{A}_{\mathcal{R}}^I}
 \end{aligned}$$

Theorem 16. Let \mathcal{P} , \mathcal{Q} and \mathcal{R} be declarative specifications such that \mathcal{P} and \mathcal{Q} are compatible for conjunction. If $\mathcal{R} \sqsubseteq_{dec} \mathcal{P} \wedge \mathcal{Q}$, then $\mathcal{R} \sqsubseteq_{dec} \mathcal{P}$ and $\mathcal{R} \sqsubseteq_{dec} \mathcal{Q}$.

Proof. By Theorem 14, we know that $\mathcal{P} \wedge \mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$ and $\mathcal{P} \wedge \mathcal{Q} \sqsubseteq_{dec} \mathcal{Q}$. Now by transitivity of refinement we see $\mathcal{R} \sqsubseteq_{dec} \mathcal{P}$ and $\mathcal{R} \sqsubseteq_{dec} \mathcal{Q}$.

Theorem 17. Let \mathcal{P} and \mathcal{Q} be declarative specifications compatible for conjunction. If $\mathcal{P} \wedge \mathcal{Q} \equiv_{dec} \mathcal{Q}$, then $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$.

Proof.

$$\begin{aligned}
 & \mathcal{P} \wedge \mathcal{Q} \equiv_{dec} \mathcal{Q} \\
 \implies & \mathcal{Q} \sqsubseteq_{dec} \mathcal{P} \wedge \mathcal{Q} && \text{equivalence} \\
 \iff & \mathcal{Q} \sqsubseteq_{dec} \mathcal{P} \wedge \mathcal{Q} \text{ and } \mathcal{P} \wedge \mathcal{Q} \sqsubseteq_{dec} \mathcal{P} && \text{Theorem 14} \\
 \implies & \mathcal{Q} \sqsubseteq_{dec} \mathcal{P} && \text{transitivity}
 \end{aligned}$$

Theorem 18. Let \mathcal{P} and \mathcal{Q} be declarative specifications compatible for conjunction. If $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$, then $\mathcal{P} \wedge \mathcal{Q} \equiv_{dec} \mathcal{Q}$.

Proof.

$$\begin{aligned}
& Q \sqsubseteq_{dec} \mathcal{P} \\
\iff & Q \sqsubseteq_{dec} \mathcal{P} \text{ and } \mathcal{P} \wedge Q \sqsubseteq_{dec} Q && \text{Theorem 14} \\
\implies & Q \wedge Q \sqsubseteq_{dec} \mathcal{P} \wedge Q \text{ and } \mathcal{P} \wedge Q \sqsubseteq_{dec} Q && \text{Theorem 3} \\
\iff & Q \sqsubseteq_{dec} \mathcal{P} \wedge Q \text{ and } \mathcal{P} \wedge Q \sqsubseteq_{dec} Q && \text{idempotence of } \wedge \\
\iff & \mathcal{P} \wedge Q \equiv_{dec} Q && \text{equivalence}
\end{aligned}$$

Proof 4 (Theorem 3) *As usual, inconsistent trace containment is the most involved part.*

$$\begin{aligned}
& t \in F_{\mathcal{E}(\mathcal{Q} \wedge \mathcal{R})} \\
\iff & t \in F_{\mathcal{E}(\mathcal{Q}) \wedge \mathcal{E}(\mathcal{R})} \\
\iff & t \in F_{\mathcal{E}(\mathcal{Q}) \uparrow \mathcal{A}_{\mathcal{R}}^I} \cap F_{\mathcal{E}(\mathcal{R}) \uparrow \mathcal{A}_{\mathcal{Q}}^I} \\
\iff & t \in F_{\mathcal{E}(\mathcal{Q}) \uparrow \mathcal{A}_{\mathcal{R}}^I} \cap F_{(\mathcal{E}(\mathcal{R}) \uparrow \mathcal{A}_{\mathcal{P}}^I) \uparrow \mathcal{A}_{\mathcal{Q}}^I} && \mathcal{A}_{\mathcal{P}}^I \subseteq \mathcal{A}_{\mathcal{Q}}^I \\
\implies & t \in F_{(\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{Q}}^I) \uparrow \mathcal{A}_{\mathcal{R}}^I} \cap F_{(\mathcal{E}(\mathcal{R}) \uparrow \mathcal{A}_{\mathcal{P}}^I) \uparrow \mathcal{A}_{\mathcal{Q}}^I} && Q \sqsubseteq_{dec} \mathcal{P} \\
\iff & t \in F_{(\mathcal{E}(\mathcal{P}) \uparrow \mathcal{A}_{\mathcal{R}}^I) \uparrow \mathcal{A}_{\mathcal{Q}}^I} \cap F_{(\mathcal{E}(\mathcal{R}) \uparrow \mathcal{A}_{\mathcal{P}}^I) \uparrow \mathcal{A}_{\mathcal{Q}}^I} \\
\iff & t \in F_{(\mathcal{E}(\mathcal{P}) \wedge \mathcal{E}(\mathcal{R})) \uparrow \mathcal{A}_{\mathcal{Q}}^I} \\
\iff & t \in F_{\mathcal{E}(\mathcal{P} \wedge \mathcal{R}) \uparrow \mathcal{A}_{\mathcal{Q}}^I} \\
\iff & t \in F_{\mathcal{E}(\mathcal{P} \wedge \mathcal{R}) \uparrow \mathcal{A}_{\mathcal{Q} \wedge \mathcal{R}}^I}
\end{aligned}$$

A.4 Quotient

Proof 5 (Theorem 4) *Follows from Theorems 19 and 20.*

Theorem 19 (Well-formedness of quotient). *Let \mathcal{P} and \mathcal{R} be declarative specifications such that $\mathcal{A}_{\mathcal{P}}^O \subseteq \mathcal{A}_{\mathcal{R}}^O$ and $\mathcal{A}_{\mathcal{P}} \subseteq \mathcal{A}_{\mathcal{R}}$. Then $\mathcal{R} \setminus \mathcal{P}$ is a declarative specification iff $T_{\mathcal{R} \setminus \mathcal{P}} \neq \emptyset$.*

Proof. Suppose $\mathcal{R} \setminus \mathcal{P}$ is a declarative specification. By the definition of declarative specifications it is known that $T_{\mathcal{R} \setminus \mathcal{P}} \neq \emptyset$.

Instead suppose that $\mathcal{R} \setminus \mathcal{P}$ is not a declarative specification. By definition of $F_{\mathcal{R} \setminus \mathcal{P}}$ and $T_{\mathcal{R} \setminus \mathcal{P}}$ we see that $F_{\mathcal{R} \setminus \mathcal{P}} \subseteq T_{\mathcal{R} \setminus \mathcal{P}}$ and $T_{\mathcal{R} \setminus \mathcal{P}}$ is prefix closed.

Moreover $F_{\mathcal{R} \setminus \mathcal{P}}$ is closed under extensions. To see this, let $t \in F_{\mathcal{R} \setminus \mathcal{P}}$. Then $t \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$ or $t \in F_{\mathcal{E}(\mathcal{R})}$. Now $t \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$ implies $tt' \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$ as $T_{\mathcal{P}}$ is prefix closed. Furthermore, $t \in F_{\mathcal{E}(\mathcal{R})}$ implies $tt' \in F_{\mathcal{E}(\mathcal{R})}$ as $F_{\mathcal{E}(\mathcal{R})}$ is closed under extensions.

It can also be seen that $T_{\mathcal{R} \setminus \mathcal{P}}$ is input-enabled. For the proof, suppose $t \in T_{\mathcal{R} \setminus \mathcal{P}}$, $i \in \mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{R}}^I$ and $ti \notin T_{\mathcal{R} \setminus \mathcal{P}}$. By the input-enabledness of $T_{\mathcal{P}}$ and $T_{\mathcal{R}}$, it follows that $ti \in T_{\mathcal{R} \setminus \mathcal{P}}$, which is contradictory. Thus $T_{\mathcal{R} \setminus \mathcal{P}}$ is input-enabled.

Given that $\mathcal{R} \setminus \mathcal{P}$ satisfies the 4 key conditions for being a declarative specification, and we're assuming that $\mathcal{R} \setminus \mathcal{P}$ is not a declarative specification, it must be the case that $T_{\mathcal{R} \setminus \mathcal{P}} = \emptyset$.

Theorem 20 (Non-existence of quotient). . If $T_{\mathcal{R} \setminus \mathcal{P}} = \emptyset$, then there is no declarative specification \mathcal{Q} over an interface consisting of inputs $\mathcal{A}_{\mathcal{R} \setminus \mathcal{P}}^I$ and outputs $\mathcal{A}_{\mathcal{R} \setminus \mathcal{P}}^O$ such that $\mathcal{P} \parallel \mathcal{Q} \sqsubseteq_{dec} \mathcal{R}$.

Proof. As $T_{\mathcal{R} \setminus \mathcal{P}} = \emptyset$, it is known that $\epsilon \notin T_{\mathcal{R} \setminus \mathcal{P}}$. By the definition of $T_{\mathcal{R} \setminus \mathcal{P}}$ it follows that there is a $t \in (\mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{R}}^I)^*$ such that: (i) $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}}$ and $t \notin F_{\mathcal{E}(\mathcal{R})}$; or (ii) $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t \notin T_{\mathcal{R}}$.

Begin by supposing (i). As $t \notin F_{\mathcal{E}(\mathcal{R})}$, we know $\epsilon \notin F_{\mathcal{E}(\mathcal{R})}$. As $\mathcal{P} \parallel \mathcal{Q} \sqsubseteq_{dec} \mathcal{R}$, we see that $\epsilon \notin F_{\mathcal{E}(\mathcal{P})}$ and $\epsilon \notin F_{\mathcal{E}(\mathcal{Q})}$. So there exists $t'a$ a prefix of t with $a \in \mathcal{A}_{\mathcal{P}}$ such that $t' \upharpoonright \mathcal{A}_{\mathcal{P}} \notin F_{\mathcal{P}}$ and $t'a \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}}$. But again, as $\mathcal{P} \parallel \mathcal{Q} \sqsubseteq_{dec} \mathcal{R}$, we know $F_{\mathcal{E}(\mathcal{P} \parallel \mathcal{Q})} \subseteq F_{\mathcal{E}(\mathcal{R})}$. As $t'a \notin F_{\mathcal{E}(\mathcal{R})}$, it follows $t'a \notin F_{\mathcal{E}(\mathcal{P} \parallel \mathcal{Q})}$. Given $t'a \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}}$, it must be the case that $t'a \notin T_{\mathcal{Q}}$, otherwise $t'a \in F_{\mathcal{P} \parallel \mathcal{Q}}$. Now, as $\epsilon \in T_{\mathcal{Q}}$ and $t'a \notin T_{\mathcal{Q}}$, there exists $t''a'$ a prefix of $t'a$ such that $t'' \in T_{\mathcal{Q}}$ and $t''a' \notin T_{\mathcal{Q}}$. Hence $a' \in \mathcal{A}_{\mathcal{Q}}^O = \mathcal{A}_{\mathcal{R}}^O \setminus \mathcal{A}_{\mathcal{P}}^O$, which is contradictory as a' occurs in t and so $a' \in \mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{R}}^I$. Therefore \mathcal{Q} cannot exist.

Instead suppose (ii). So there exists $t'a$ a prefix of t in which $t' \in T_{\mathcal{R}}$ and $t'a \notin T_{\mathcal{R}}$. Hence $a \in \mathcal{A}_{\mathcal{R}}^O$, and moreover, $a \in \mathcal{A}_{\mathcal{P}}^O$. Given $\mathcal{P} \parallel \mathcal{Q} \sqsubseteq_{dec} \mathcal{R}$, we know that $T_{\mathcal{P} \parallel \mathcal{Q}} \subseteq T_{\mathcal{R}}$. As $t'a \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t'a \notin T_{\mathcal{R}}$ it must be the case that $t'a \notin T_{\mathcal{Q}}$, otherwise $t'a \in T_{\mathcal{P} \parallel \mathcal{Q}}$, which would be contradictory. As \mathcal{Q} is a declarative specification, it is known that $\epsilon \in T_{\mathcal{Q}}$. Hence there exists $t''a'$ a prefix of $t'a$ such that $t'' \in T_{\mathcal{Q}}$ and $t''a' \notin T_{\mathcal{Q}}$. Thus $a' \in \mathcal{A}_{\mathcal{Q}}^O = \mathcal{A}_{\mathcal{R}}^O \setminus \mathcal{A}_{\mathcal{P}}^O$. But as a' occurs in t , it follows that $a' \in \mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{R}}^I$, which contradicts $a' \in \mathcal{A}_{\mathcal{Q}}^O$. So again, \mathcal{Q} cannot exist.

Proof 6 (Theorem 5) Follows from Theorems 21 and 22.

Theorem 21 (Quotient is sound). Let \mathcal{P} and \mathcal{R} be declarative specifications such that $\mathcal{A}_{\mathcal{P}}^O \subseteq \mathcal{A}_{\mathcal{R}}^O$ and $\mathcal{A}_{\mathcal{P}} \subseteq \mathcal{A}_{\mathcal{R}}$. If $T_{\mathcal{R} \setminus \mathcal{P}} \neq \emptyset$, then $\mathcal{P} \parallel (\mathcal{R} \setminus \mathcal{P}) \sqsubseteq_{dec} \mathcal{R}$.

Proof. If $T_{\mathcal{R}} = F_{\mathcal{E}(\mathcal{R})}$, then $\epsilon \in F_{\mathcal{E}(\mathcal{R})}$, so $T_{\mathcal{R}} = F_{\mathcal{E}(\mathcal{R})} = \mathcal{A}_{\mathcal{R}}^*$. Hence $\mathcal{P} \parallel (\mathcal{R} \setminus \mathcal{P}) \sqsubseteq_{dec} \mathcal{R}$ trivially. So instead suppose that $T_{\mathcal{R}} \neq F_{\mathcal{E}(\mathcal{R})}$. Then $\epsilon \notin F_{\mathcal{E}(\mathcal{R})}$.

First show that $T_{\mathcal{P} \parallel (\mathcal{R} \setminus \mathcal{P})} \subseteq T_{\mathcal{R}}$. Let $t \in T_{\mathcal{P} \parallel (\mathcal{R} \setminus \mathcal{P})}$ and suppose for a contradiction that $t \notin T_{\mathcal{R}}$. So either: (i) $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t \in T_{\mathcal{R} \setminus \mathcal{P}}$; (ii) there is a prefix t' of t such that $t' \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}}$ and $t' \in T_{\mathcal{R} \setminus \mathcal{P}}$; or (iii) there is a prefix t' of t such that $t' \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t' \in F_{\mathcal{R} \setminus \mathcal{P}}$.

Begin by supposing (i). By the definition of $T_{\mathcal{R} \setminus \mathcal{P}}$, we know $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$, ($t' \upharpoonright \mathcal{A}_{\mathcal{P}} \notin F_{\mathcal{P}}$ or $t' \in F_{\mathcal{E}(\mathcal{R})}$), and ($t' \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$ or $t' \in T_{\mathcal{R}}$). Consequently, $t \in T_{\mathcal{R}}$, but this contradicts our initial assumption. Instead suppose (ii). By the definition of $T_{\mathcal{R} \setminus \mathcal{P}}$, we know that $t' \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}}$, ($t' \upharpoonright \mathcal{A}_{\mathcal{P}} \notin F_{\mathcal{P}}$ or $t' \in F_{\mathcal{E}(\mathcal{R})}$), and ($t' \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$ or $t' \in T_{\mathcal{R}}$). Hence $t' \in F_{\mathcal{E}(\mathcal{R})}$ and $t' \in T_{\mathcal{R}}$, which in turn implies $t \in F_{\mathcal{E}(\mathcal{R})}$ and so $t \in T_{\mathcal{R}}$, which is contradictory. The case (iii) follows by the same reasoning as for (ii), but by resorting to the definition of $F_{\mathcal{R} \setminus \mathcal{P}}$ rather than $T_{\mathcal{R} \setminus \mathcal{P}}$.

Finally, show that $F_{\mathcal{E}(\mathcal{P} \parallel (\mathcal{R} \setminus \mathcal{P}))} \subseteq F_{\mathcal{E}(\mathcal{R})}$. Let $t \in F_{\mathcal{E}(\mathcal{P} \parallel (\mathcal{R} \setminus \mathcal{P}))}$ and suppose for a contradiction that $t \notin F_{\mathcal{E}(\mathcal{R})}$. Then there exists a $t' \in (\mathcal{A}_{\mathcal{P}}^O \cup \mathcal{A}_{\mathcal{R} \setminus \mathcal{P}}^O)^* = (\mathcal{A}_{\mathcal{R}}^O)^*$ such that $tt' \in F_{\mathcal{P} \parallel (\mathcal{R} \setminus \mathcal{P})}$. Then there is a prefix t'' of tt' such that either: (i) $t'' \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}}$ and $t'' \in T_{\mathcal{R} \setminus \mathcal{P}}$; or (ii) $t'' \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t'' \in F_{\mathcal{R} \setminus \mathcal{P}}$.

Begin by assuming (i). Then by the definition of $T_{\mathcal{R} \setminus \mathcal{P}}$, we know that $t'' \in F_{\mathcal{E}(\mathcal{R})}$. If t'' is a prefix of t , then clearly $t \in F_{\mathcal{E}(\mathcal{R})}$ by upward closure of $F_{\mathcal{E}(\mathcal{R})}$. If t is a prefix of t'' , then also $t \in F_{\mathcal{E}(\mathcal{R})}$ by backward propagation of inconsistencies over t' . Instead suppose (ii). By the definition of $F_{\mathcal{R} \setminus \mathcal{P}}$ and the same reasoning as for (i), it follows that $t \in F_{\mathcal{E}(\mathcal{R})}$.

Theorem 22 (Quotient is least refined). Let \mathcal{P} and \mathcal{R} be declarative specifications such that $\mathcal{A}_{\mathcal{P}}^O \subseteq \mathcal{A}_{\mathcal{R}}^O$ and $\mathcal{A}_{\mathcal{P}} \subseteq \mathcal{A}_{\mathcal{R}}$. If \mathcal{Q} is a declarative specification over an interface consisting of inputs $\mathcal{A}_{\mathcal{R} \setminus \mathcal{P}}^I$ and outputs $\mathcal{A}_{\mathcal{R} \setminus \mathcal{P}}^O$ such that $\mathcal{P} \parallel \mathcal{Q} \sqsubseteq_{dec} \mathcal{R}$, then $\mathcal{Q} \sqsubseteq_{dec} \mathcal{R} \setminus \mathcal{P}$.

Proof. We need to show $T_{\mathcal{Q}} \subseteq T_{\mathcal{R} \setminus \mathcal{P}}$ and $F_{\mathcal{E}(\mathcal{Q})} \subseteq F_{\mathcal{E}(\mathcal{R} \setminus \mathcal{P})}$, given that $T_{\mathcal{P} \parallel \mathcal{Q}} \subseteq T_{\mathcal{R}}$ and $F_{\mathcal{E}(\mathcal{P} \parallel \mathcal{Q})} \subseteq F_{\mathcal{E}(\mathcal{R})}$. So let $t \in T_{\mathcal{Q}}$.

Begin by assuming that $t \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$. Then there exists a prefix $t'a$ of t , such that $t' \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t'a \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$. So $t' \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t' \in T_{\mathcal{Q}}$ implies that $t' \in T_{\mathcal{P} \parallel \mathcal{Q}}$. Hence $t' \in T_{\mathcal{R}}$. Now, as $t'a \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$, it follows that $t'a \upharpoonright \mathcal{A}_{\mathcal{P}} \notin F_{\mathcal{P}}$. Thus, for all t'' a strict prefix of $t'a$ it is known that $t'' \upharpoonright \mathcal{A}_{\mathcal{P}} \notin F_{\mathcal{P}}$ and $t'' \in T_{\mathcal{R}}$. Moreover, for all traces t'' with common prefix $t'a$ it is known that $t'' \upharpoonright \mathcal{A}_{\mathcal{P}} \notin F_{\mathcal{P}}$ and $t'' \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$. Hence $t'a \in T_{\mathcal{R} \setminus \mathcal{P}}$ and so by the previous reasoning $t \in T_{\mathcal{R} \setminus \mathcal{P}}$.

Instead suppose that $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$. Then $t \in T_{\mathcal{P} \parallel \mathcal{Q}}$ and so $t \in T_{\mathcal{R}}$. As $t \in T_{\mathcal{Q}}$, we know that $tt' \in T_{\mathcal{Q}}$ for all $t' \in \mathcal{A}_{\mathcal{R} \setminus \mathcal{P}}^I^*$. Now suppose for a contradiction that $t \notin T_{\mathcal{R} \setminus \mathcal{P}}$ (where this is the smallest t such that $t \in T_{\mathcal{Q}}$ and $t \notin T_{\mathcal{R} \setminus \mathcal{P}}$). Then there exists a smallest $t''a \in \mathcal{A}_{\mathcal{R} \setminus \mathcal{P}}^I^*$ such that either: (i) $tt''a \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}}$ and $tt''a \notin F_{\mathcal{E}(\mathcal{R})}$; or (ii) $tt''a \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $tt''a \notin T_{\mathcal{R}}$. But case (i) cannot hold because $tt''a \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}}$ implies that $tt''a \in F_{\mathcal{P} \parallel \mathcal{Q}}$, so $tt''a \in F_{\mathcal{E}(\mathcal{P} \parallel \mathcal{Q})}$ which in turn implies $tt''a \in F_{\mathcal{E}(\mathcal{R})}$, which is contradictory. Therefore case (ii) must hold. However, if $a \in \mathcal{A}_{\mathcal{R}}^I$, then $tt''a \in T_{\mathcal{R}}$ by input-enabledness of declarative specifications. Alternatively, if $a \in \mathcal{A}_{\mathcal{P}}^O$, then $tt''a \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$. As $tt''a \in T_{\mathcal{Q}}$, it follows that $tt''a \in T_{\mathcal{P} \parallel \mathcal{Q}}$ and so $tt''a \in T_{\mathcal{R}}$, which is contradictory. Therefore it must be the case that $t \in T_{\mathcal{R} \setminus \mathcal{P}}$.

Now to deal with the inconsistent traces, let $t \in F_{\mathcal{E}(\mathcal{Q})}$. Then there exists $t' \in \mathcal{A}_{\mathcal{R} \setminus \mathcal{P}}^O^*$ such that $tt' \in F_{\mathcal{Q}}$.

Assume that $tt' \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$. It therefore follows that $tt' \in F_{\mathcal{P} \parallel \mathcal{Q}}$ and so $t \in F_{\mathcal{E}(\mathcal{P} \parallel \mathcal{Q})}$, which in turn implies that $t \in F_{\mathcal{E}(\mathcal{R})}$. Now suppose for a contradiction that $t \notin F_{\mathcal{E}(\mathcal{R} \setminus \mathcal{P})}$, which means that $t \notin F_{\mathcal{R} \setminus \mathcal{P}}$. Then either: (i) $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t \notin F_{\mathcal{E}(\mathcal{R})}$; (ii) $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in F_{\mathcal{P}}$ and $t \notin F_{\mathcal{E}(\mathcal{R})}$; or (iii) $t \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$ and $t \notin T_{\mathcal{R}}$. But $t \in F_{\mathcal{E}(\mathcal{R})}$ implies $t \in T_{\mathcal{R}}$, so none of the conditions are satisfiable. Therefore $t \in F_{\mathcal{E}(\mathcal{R} \setminus \mathcal{P})}$.

Instead suppose $tt' \upharpoonright \mathcal{A}_{\mathcal{P}} \notin T_{\mathcal{P}}$. Then for any prefix s of tt' it is known that $s \upharpoonright \mathcal{A}_{\mathcal{P}} \notin F_{\mathcal{P}}$. Moreover, for any prefix s of tt' , if $s \upharpoonright \mathcal{A}_{\mathcal{P}} \in T_{\mathcal{P}}$, then because $s \in T_{\mathcal{Q}}$, it follows that $s \in T_{\mathcal{P} \parallel \mathcal{Q}}$ and so $s \in T_{\mathcal{R}}$. By the definition of $F_{\mathcal{R} \setminus \mathcal{P}}$, it therefore follows that $tt' \in F_{\mathcal{R} \setminus \mathcal{P}}$, and so $t \in F_{\mathcal{E}(\mathcal{R} \setminus \mathcal{P})}$.

Theorem 23 (Compositionality of quotient). *Let \mathcal{P} , \mathcal{Q} and \mathcal{R} be declarative specifications such that $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$.*

- *If $\mathcal{Q} \setminus \mathcal{R}$ and $\mathcal{P} \setminus \mathcal{R}$ are defined, then $\mathcal{Q} \setminus \mathcal{R} \sqsubseteq_{dec} \mathcal{P} \setminus \mathcal{R}$.*
- *If $\mathcal{R} \setminus \mathcal{Q}$ and $\mathcal{R} \setminus \mathcal{P}$ are defined, and $(\mathcal{A}_{\mathcal{Q}}^I \cap \mathcal{A}_{\mathcal{R}}^O) \setminus \mathcal{A}_{\mathcal{P}} = \emptyset$, then $\mathcal{R} \setminus \mathcal{Q} \sqsupseteq_{dec} \mathcal{R} \setminus \mathcal{P}$.*

Proof. For the first property, note that $\mathcal{R} \parallel (\mathcal{Q} \setminus \mathcal{R}) \sqsubseteq_{dec} \mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$. By transitivity, $\mathcal{R} \parallel (\mathcal{Q} \setminus \mathcal{R}) \sqsubseteq_{dec} \mathcal{P}$ and so $\mathcal{Q} \setminus \mathcal{R} \sqsubseteq_{dec} \mathcal{P} \setminus \mathcal{R}$. For the second property, when the constraint $(\mathcal{A}_{\mathcal{Q}}^I \cap \mathcal{A}_{\mathcal{R}}^O) \setminus \mathcal{A}_{\mathcal{P}} = \emptyset$ holds we know that $\mathcal{Q} \parallel (\mathcal{R} \setminus \mathcal{P}) \sqsubseteq_{dec} \mathcal{P} \parallel (\mathcal{R} \setminus \mathcal{P})$ by Theorem 1. Now as $\mathcal{P} \parallel (\mathcal{R} \setminus \mathcal{P}) \sqsubseteq_{dec} \mathcal{R}$, it follows that $\mathcal{R} \setminus \mathcal{P} \sqsubseteq_{dec} \mathcal{R} \setminus \mathcal{Q}$.

B Proofs for Operational Specification Theory

B.1 Refinement

Proof 7 (Lemma 2) *Follows immediately from the fact that declarative refinement is a preorder (Lemma 1).*

Proof 8 (Theorem 7) *By induction on the length of t .*

- $t \equiv \epsilon$. If $\epsilon \in ST(p_{\mathcal{P}})$, then $h_{p_{\mathcal{P}}}(\epsilon) \neq \emptyset$. This means that $p_{\mathcal{P}} \xrightarrow{\epsilon}_{\mathcal{P}} p'$ implies $p' \notin F_{\mathcal{P}}$. Consequently, $\epsilon \notin F_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. Now as it is always the case that $p_{\mathcal{P}} \xrightarrow{\epsilon}_{\mathcal{P}} p_{\mathcal{P}}$, it follows that $p_{\mathcal{P}} \in T_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. Hence $\epsilon \in T_{\llbracket p_{\mathcal{P}} \rrbracket^*} \setminus F_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. The other direction follows by the same reasoning.
- $t \equiv t'o$ with $o \in \mathcal{A}_{\mathcal{P}}^O$. By the induction hypothesis it is known that $t' \in ST(p) \iff t' \in T_{\llbracket p_{\mathcal{P}} \rrbracket^*} \setminus F_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. So suppose that $t'o \in ST(p_{\mathcal{P}})$. Then: (i) $\forall p' \in h_{p_{\mathcal{P}}}(t')$, $p' \xrightarrow{!o}_{\mathcal{P}} p''$ implies $p'' \notin F_{\mathcal{P}}$; and (ii) there is some $p' \in h_{p_{\mathcal{P}}}(t')$ such that $p' \xrightarrow{!o}_{\mathcal{P}}$. Thus, taking (i) in conjunction with $t' \notin F_{\llbracket p_{\mathcal{P}} \rrbracket^*}$ it follows $t'o \notin F_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. Taking $t' \in ST(p_{\mathcal{P}})$ along with (ii) implies that $t'o \in T_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. Therefore, $t'o \in T_{\llbracket p_{\mathcal{P}} \rrbracket^*} \setminus F_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. The other direction follows by the same reasoning.
- $t \equiv t'i$ with $i \in \mathcal{A}_{\mathcal{P}}^I$. By the induction hypothesis it is known that $t' \in ST(p) \iff t' \in T_{\llbracket p_{\mathcal{P}} \rrbracket^*} \setminus F_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. So suppose that $t'i \in ST(p_{\mathcal{P}})$. Then: (i) $\forall p' \in h_{p_{\mathcal{P}}}(t')$, $p' \xrightarrow{-i}_{\mathcal{P}} p''$ implies $p'' \notin F_{\mathcal{P}}$; and (ii) for each $p' \in h_{p_{\mathcal{P}}}(t')$, $p' \xrightarrow{-i}_{\mathcal{P}}$. From taking (i) and (ii) in conjunction with $t' \notin F_{\llbracket p_{\mathcal{P}} \rrbracket^*}$ it follows $t'i \notin F_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. Taking $t' \in ST(p_{\mathcal{P}})$ along with (ii) implies that $t'i \in T_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. Therefore, $t'i \in T_{\llbracket p_{\mathcal{P}} \rrbracket^*} \setminus F_{\llbracket p_{\mathcal{P}} \rrbracket^*}$. The other direction follows by the same reasoning.

B.2 Error-completion

The proofs for Lemmas 3 and 4 are straightforward.

B.3 Parallel Composition

Proof 9 (Theorem 8) *Let $t \in T_{[[p_P]]^* || [[q_Q]]^*}$. Then $t \upharpoonright \mathcal{A}_P \in T_{[[p_P]]^*}$ and $t \upharpoonright \mathcal{A}_Q \in T_{[[q_Q]]^*}$. Therefore, by Lemma 4 there is a run over $t \upharpoonright \mathcal{A}_P$ starting in state p_P of P_\perp and a run over $t \upharpoonright \mathcal{A}_Q$ starting in state q_Q of Q_\perp . By Lemma 8 it follows that there is a run over t starting in state $p_P || q_Q$ on $P_\perp || Q_\perp$. Therefore $t \in T_{[[p_P || q_Q]]^*}$. The other direction is similar.*

Lemma 8. *Let $t \in (\mathcal{A}_P \cup \mathcal{A}_Q)^*$. Then $\{p' || q' : p_P || q_Q \xrightarrow{t}_{P_\perp || Q_\perp} p' || q'\} = \{p' || q' : p_P \xrightarrow{t \upharpoonright \mathcal{A}_P}_{P_\perp} p' \text{ and } q_Q \xrightarrow{t \upharpoonright \mathcal{A}_Q}_{Q_\perp} q'\}$.*

Proof. By induction on the length of t .

B.4 Conjunction

The proof for conjunction is even more straightforward than that for parallel composition, so is omitted.

B.5 Quotient

Proof 10 (Theorem 10) *Begin by supposing that $r_R \setminus p_P \notin F_{R \setminus P}$ or $r_R \in F_{\mathcal{E}(R)}$. Then by Theorem 11 we know that $[[r_R \setminus p_P]]^* = [[r_R]]^* \setminus [[p_P]]^*$ and moreover, $\epsilon \in T_{[[r_R \setminus p_P]]^*}$. Thus $\epsilon \in T_{[[r_R]]^* \setminus [[p_P]]^*}$ and so $[[p_P]]^* || ([[r_R]]^* \setminus [[p_P]]^*) \sqsubseteq_{dec} [[r_R]]^*$ as $T_{[[r_R]]^* \setminus [[p_P]]^*} \neq \emptyset$. Hence $[[p_P]]^* || [[r_R \setminus p_P]]^* \sqsubseteq_{dec} [[r_R]]^*$, which implies that $p_P || (r_R \setminus p_P) \sqsubseteq_{op} r_R$.*

Instead suppose that $r_R \setminus p_P \in F_{R \setminus P}$ and $r \notin F_{\mathcal{E}(R)}$. Then there exists $t \in (\mathcal{A}_{R \setminus P}^I)^$ of minimal length such that $r_R \setminus p_P \xrightarrow{t}_{R \setminus P} R \setminus P$ with $R \setminus P \in F_{R \setminus P}$ by condition F1 or F2. So condition on the length of t .*

If $t \equiv \epsilon$, then $r_R \setminus p_P = R \setminus P = \text{succ}_R^\epsilon(r_R) \setminus \text{succ}_P^\epsilon(p_P)$. Certainly $R \neq \emptyset$ and $P \neq \emptyset$ as r_R and p_P are contained within R and P respectively. Moreover, $r_R \notin F_{\mathcal{E}(R_\perp)}$, therefore it must be the case that $P \cap F_{P_\perp} \neq \emptyset$. Consequently, we know that $\epsilon \in F_{[[p_P]]^}$ and $\epsilon \notin F_{\mathcal{E}([r_R]^*)}$. As a result, $\epsilon \notin T_{[[r_R]]^* \setminus [[p_P]]^*}$ and so there is no declarative specification q' such that $[[p_P]]^* || q' \sqsubseteq_{dec} [[r_R]]^*$. Hence it holds that there is no Logic IOLTS q such that $[[p_P]]^* || [[q]]^* \sqsubseteq_{dec} [[r_R]]^*$, which in turn means there is no Logic IOLTS q such that $p_P || q \sqsubseteq_{op} r_R$.*

Now assume that $t \equiv t'i$ with $i \in \mathcal{A}_{R \setminus P}^I$. Then $r_R \setminus p_P \xrightarrow{t'}_{R \setminus P} R' \setminus P' \xrightarrow{i}_{R \setminus P} R \setminus P$. Conditioning on i , begin by assuming that $i \in \mathcal{A}_P^I \cap \mathcal{A}_R^I$. For the i -transition to exist, it must be the case that $\text{succ}_R^i(R') = \emptyset$. Now for $R \setminus P$ to be in $F_{R \setminus P}$ by F1 or F2, it must be the case that $\text{succ}_P^i(P') \cap F_{P_\perp} \neq \emptyset$. But then we see that $t'i \upharpoonright \mathcal{A}_P \in F_{[[p_P]]^} \not\Rightarrow t'i \in F_{\mathcal{E}([r_R]^*)}$. So by Definition 8, it is known that $\epsilon \notin T_{[[r_R]]^* \setminus [[p_P]]^*}$. By the same reasoning as for $t \equiv \epsilon$, the quotient cannot exist.*

When $i \in \mathcal{A}_P^O \cap \mathcal{A}_R^O$, the i -transition can only exist when $\text{succ}_R^i(R') \cap F_{\mathcal{E}(R_\perp)} = \emptyset$ and $\text{succ}_P^i(P') \neq \emptyset$. So for $R \setminus P$ to be in $F_{R \setminus P}$ it must be the case that either: (i) $\text{succ}_R^i(R') = \emptyset$; or (ii) $\text{succ}_P^i(P') \cap F_{P_\perp} \neq \emptyset$. Under (i) we know that $t'i \upharpoonright$

$\mathcal{A}_P \in T_{[[PP]]^*} \not\Rightarrow t'i \in T_{[[rR]]^*}$ and under (ii) $t'i \upharpoonright \mathcal{A}_P \in F_{[[PP]]^*} \not\Rightarrow t'i \in F_{\mathcal{E}(\llbracket rR \rrbracket^*)}$. Hence $\epsilon \notin T_{[[rR]]^* \setminus [[PP]]^*}$ and so the quotient cannot exist.

Finally, assume that $i \in \mathcal{A}_R^I \setminus \mathcal{A}_P$. For the i -transition to exist it must be the case that $\text{succ}_R^i(R') \cap F_{\mathcal{E}(R_\perp)} = \emptyset$. As $R \setminus P \in F_{R \setminus P}$, $R' \setminus P' \notin F_{R \setminus P}$ and $P = P'$, it follows that $R = \emptyset$, while $P \neq \emptyset$. Hence $t'i \upharpoonright \mathcal{A}_P \in T_{[[PP]]^*} \not\Rightarrow t'i \in T_{[[rR]]^*}$ and so $\epsilon \notin T_{[[rR]]^* \setminus [[PP]]^*}$, which again means that the quotient cannot exist.

Proof 11 (Theorem 11) Show that $t \in T_{[[rR]]^* \setminus [[PP]]^*} \iff t \in T_{[[rR \setminus PP]]^*}$ and $t \in F_{[[rR]]^* \setminus [[PP]]^*} \iff t \in F_{[[rR \setminus PP]]^*}$ by induction on the length of the trace t .

Case $t \equiv \epsilon$. Begin by showing $\epsilon \in F_{[[rR]]^* \setminus [[PP]]^*} \iff \epsilon \in F_{[[rR \setminus PP]]^*}$.

Suppose $\epsilon \in F_{[[rR]]^* \setminus [[PP]]^*}$. Then by Definition 8, we know that ($\epsilon \notin T_{[[PP]]^*}$ or $\epsilon \in F_{\mathcal{E}(\llbracket rR \rrbracket^*)}$), ($\epsilon \notin F_{[[PP]]^*}$ or $\epsilon \in F_{\mathcal{E}(\llbracket rR \rrbracket^*)}$) and ($\epsilon \notin T_{[[PP]]^*}$ or $\epsilon \in T_{\mathcal{E}(\llbracket rR \rrbracket^*)}$). Consequently, $\epsilon \in F_{\mathcal{E}(\llbracket rR \rrbracket^*)}$ or $\epsilon \notin T_{[[PP]]^*}$. But as pp is a process in a Logic IOLTS, it follows that $T_{[[PP]]^*}$ is non-empty. Therefore $\epsilon \in T_{[[PP]]^*}$ and so it must be the case that $\epsilon \in F_{\mathcal{E}(\llbracket rR \rrbracket^*)}$. As a result, there is some $t' \in (\mathcal{A}_R^O)^*$ such that $t' \in F_{[[rR]]^*}$. So by Definition 13 it is known that there exists $r' \in F_R$ such that $r_R \xrightarrow{t'}_R r'$. Hence $r_R \in F_{\mathcal{E}(R_\perp)}$, which means that $r_R \setminus pp \in F_{R \setminus P}$ by condition F2. As $r_R \setminus pp \xrightarrow{\epsilon}_{R \setminus P} r_R \setminus pp$ we have $\epsilon \in F_{[[rR \setminus PP]]^*}$.

Instead suppose $\epsilon \in F_{[[rR \setminus PP]]^*}$. Then there exists $R' \setminus P' \in F_{R \setminus P}$ such that $r_R \setminus pp \xrightarrow{\epsilon}_{R \setminus P} R' \setminus P'$. But $r_R \setminus pp = \text{succ}_R^{\epsilon}(r_R) \setminus \text{succ}_P^{\epsilon}(pp)$, so $r_R \setminus pp = R' \setminus P'$, which means $r_R \setminus pp \in F_{R \setminus P}$. Consequently, to satisfy the premises of the theorem, it must be the case that $r_R \in F_{\mathcal{E}(R)}$. So there is some $t \in (\mathcal{A}_R^O)^*$ and $r' \in F_R$ such that $r_R \xrightarrow{t}_R r'$. By Definition 13 it holds that $t \in F_{[[rR]]^*}$, which in turn implies $\epsilon \in F_{\mathcal{E}(\llbracket rR \rrbracket^*)}$. So by Definition 8 it must be the case that $\epsilon \in F_{[[rR]]^* \setminus [[PP]]^*}$.

Now show that $\epsilon \in T_{[[rR]]^* \setminus [[PP]]^*} \iff \epsilon \in T_{[[rR \setminus PP]]^*}$.

Under the premises of the theorem we know that $r_R \setminus pp$ is a process in a Logic IOLTS, so it must be the case that $[[rR \setminus PP]]^*$ is a well-defined declarative specification. Hence $\epsilon \in T_{[[rR \setminus PP]]^*}$.

Now suppose for a contradiction that $\epsilon \notin T_{[[rR]]^* \setminus [[PP]]^*}$. Then there exists some $t' \in (\mathcal{A}_{R \setminus P}^I)^*$ such that: (i) $t' \upharpoonright \mathcal{A}_P \in F_{[[PP]]^*}$ and $t' \notin F_{\mathcal{E}(\llbracket rR \rrbracket^*)}$; or (ii) $t' \upharpoonright \mathcal{A}_P \in T_{[[PP]]^*}$ and $t' \notin T_{[[rR]]^*}$. Considering (i) and (ii) in conjunction with the premises of the theorem, it must be the case that $r_R \setminus pp \notin F_{R \setminus P}$. This rules out case (i) from holding by condition F2 and the repeated application of the backward propagation condition F3 over t' , so it must be that case (ii) holds. Now as $t' \notin T_{[[rR]]^*}$, there is some prefix $t''a$ of t' such that $t'' \in T_{[[rR]]^*}$, but $t''a \notin T_{[[rR]]^*}$. By input-enabledness of declarative specifications it follows that $a \in \mathcal{A}_P^O \cap \mathcal{A}_R^O$. The transition system construction of Definition 20 tells us that the a -transition must exist, but its R -successor will be empty. So by condition F1 and repeated

application of F3 it follows that $r_R \setminus p_P \in F_{R \setminus P}$, which is contradictory. Therefore, $\epsilon \in T_{[[r_R]^* \setminus [p_P]^*]}$ as required.

Case $t \equiv t'o$ with $o \in \mathcal{A}_{R \setminus P}^O$. By the induction hypothesis, we know that $t' \in F_{[[r_R]^* \setminus [p_P]^*]}$ iff $t' \in F_{[r_R \setminus p_P]^*}$.

So first assume that $t'o \in F_{[[r_R]^* \setminus [p_P]^*]}$. If $t' \in F_{[[r_R]^* \setminus [p_P]^*]}$, then by the induction hypothesis it follows that $t' \in F_{[r_R \setminus p_P]^*}$. So by Definition 13 we trivially have that $t'o \in F_{[r_R \setminus p_P]^*}$. Instead suppose that $t' \notin F_{[[r_R]^* \setminus [p_P]^*]}$. But by Definition 8 applied to both t' and $t'o$ we deduce that $t' \upharpoonright \mathcal{A}_P \in T_{[[p_P]^* \setminus F_{[[p_P]^*}]}$ and $t' \in T_{[[r_R]^* \setminus F_{\mathcal{E}([r_R]^*)}]}$, while $t'o \upharpoonright \mathcal{A}_P \notin T_{[[p_P]^*]}$ or $t'o \in F_{\mathcal{E}([r_R]^*)}$. Assuming that $t'o \in F_{\mathcal{E}([r_R]^*)}$ holds, we know that $t' \in F_{\mathcal{E}([r_R]^*)}$, which is contradictory, so it must be the case that $t'o \upharpoonright \mathcal{A}_P \notin T_{[[p_P]^*]}$ and $t'o \notin F_{\mathcal{E}([r_R]^*)}$. But then, as we know that $t' \upharpoonright \mathcal{A}_P \in T_{[[p_P]^* \setminus F_{[[p_P]^*}]}$ and $t'o \upharpoonright \mathcal{A}_P \notin T_{[[p_P]^*]}$, it must be the case that $o \in \mathcal{A}_P^O$. But then $o \notin \mathcal{A}_{R \setminus P}^O$, which is contradictory, so the case of $t' \notin F_{[[r_R]^* \setminus [p_P]^*]}$ cannot occur. Therefore, $t'o \in F_{[r_R \setminus p_P]^*}$.

Now suppose that $t'o \in F_{[r_R \setminus p_P]^*}$. Then there are two possibilities: (i) there is a prefix t'' of $t'o$ such that $r_R \setminus p_P \xrightarrow{t''}_{R \setminus P} R \setminus P$ with $R \setminus P \in F_{R \setminus P}$; or (ii) there is a prefix $t''i$ of $t'o$ such that $r_R \setminus p_P \xrightarrow{t''}_{R \setminus P} R \setminus P$ and $R \setminus P \not\xrightarrow{i}_{R \setminus P}$ with $i \in \mathcal{A}_{R \setminus P}^I$. By Lemma 4 we know that $[[r_R \setminus p_P]^*] = [[(r_R \setminus p_P)_\perp]^*]$, so it is sufficient to consider (i) only.

So begin by assuming (i). If t'' is a strict prefix of $t'o$, then $t'' \in F_{[[r_R]^* \setminus [p_P]^*]}$ by the induction hypothesis. Alternatively, if $t'' = t'o$, then $r_R \setminus p_P \xrightarrow{t''}_{R \setminus P} R' \setminus P' \xrightarrow{o}_{R \setminus P} R \setminus P$. As $R \setminus P$ exists in $R \setminus P$, it must be the case that $R \cap F_{\mathcal{E}(R)} \neq \emptyset$. Now as $o \in \mathcal{A}_{R \setminus P}^O = \mathcal{A}_R^O \setminus \mathcal{A}_P^O$, it follows that $R' \cap F_{\mathcal{E}(R)} \neq \emptyset$, hence $R' \setminus P' \in F_{R \setminus P}$ by condition F2. Therefore, by the induction hypothesis $t' \in F_{[[r_R]^* \setminus [p_P]^*]}$. From all of these cases it follows that $t'o \in F_{[[r_R]^* \setminus [p_P]^*]}$ by upwards-closure, as $[[r_R]^* \setminus [p_P]^*]$ is a declarative specification.

Demonstrating that $t'o \in T_{[[r_R]^* \setminus [p_P]^*]} \iff t'o \in T_{[r_R \setminus p_P]^*}$ is similar.

Case $t \equiv t'i$ with $i \in \mathcal{A}_{R \setminus P}^I$. By the induction hypothesis, we know that $t' \in F_{[[r_R]^* \setminus [p_P]^*]}$ iff $t' \in F_{[r_R \setminus p_P]^*}$ and $t' \in T_{[[r_R]^* \setminus [p_P]^*]}$ iff $t' \in T_{[r_R \setminus p_P]^*}$.

Begin by supposing that $t'i \in F_{[[r_R]^* \setminus [p_P]^*]}$. If $t' \in F_{[[r_R]^* \setminus [p_P]^*]}$, then by the induction hypothesis we know that $t' \in F_{[r_R \setminus p_P]^*}$ and so $t'i \in F_{[r_R \setminus p_P]^*}$ by Definition 13. So instead suppose that $t' \notin F_{[[r_R]^* \setminus [p_P]^*]}$. Then by the induction hypothesis, $t' \notin F_{[r_R \setminus p_P]^*}$. As $t'i \in F_{[[r_R]^* \setminus [p_P]^*]}$ it follows that $t' \in T_{[[r_R]^* \setminus [p_P]^*]}$, which again by the induction hypothesis means that $t' \in T_{[r_R \setminus p_P]^*}$. So by Definition 13 it must be that $r_R \setminus p_P \xrightarrow{t'}_{R \setminus P} R \setminus P$ without any intermediate state being in $F_{R \setminus P}$.

If $R \setminus P \not\rightarrow_{R \setminus P}^i$, then $t'i \in F_{[[r_R \setminus p_P]]^*}$ as required. So instead assume that $R \setminus P \xrightarrow{i}_{R \setminus P} R' \setminus P'$. Suppose for a contradiction that $t'i \notin F_{[[r_R \setminus p_P]]^*}$. Then certainly $R' \setminus P' \notin F_{R \setminus P}$. We show that this is contradictory. There are two cases: $i \in \mathcal{A}_P \cap \mathcal{A}_R$ or $i \in \mathcal{A}_R \setminus \mathcal{A}_P$.

Begin by assuming $i \in \mathcal{A}_P \cap \mathcal{A}_R$, then either $i \in \mathcal{A}_P^I \cap \mathcal{A}_R^I$ or $i \in \mathcal{A}_P^O \cap \mathcal{A}_R^O$. Suppose the former, then because $t' \in T_{[[r_R]]^* \setminus [[p_P]]^*}$, $t' \notin F_{[[r_R]]^* \setminus [[p_P]]^*}$, $t'i \in T_{[[r_R]]^* \setminus [[p_P]]^*}$ and $t' \in F_{[[r_R]]^* \setminus [[p_P]]^*}$, it follows from Definition 8 that $t' \upharpoonright \mathcal{A}_P \in T_{[[p_P]]^*}$ and $t' \notin F_{\mathcal{E}([[r_R]]^*)}$, while $t'i \upharpoonright \mathcal{A}_P \notin T_{[[p_P]]^*}$ or $t'i \in F_{\mathcal{E}([[r_R]]^*)}$. As $t' \upharpoonright \mathcal{A}_P \in T_{[[p_P]]^*}$, it must be the case that $t'i \upharpoonright \mathcal{A}_P \in T_{[[p_P]]^*}$ by input-enabledness of declarative specifications. Therefore $t'i \in F_{\mathcal{E}([[r_R]]^*)}$, but then $R' \setminus P' \in F_{R \setminus P}$ by condition F2, which is contradictory. Instead suppose that $i \in \mathcal{A}_P^O \cap \mathcal{A}_R^O$. Then from similar reasoning, $t' \notin F_{\mathcal{E}([[r_R]]^*)}$ implies $t'i \notin F_{\mathcal{E}([[r_R]]^*)}$ as $i \in \mathcal{A}_R^O$. Hence $t'i \upharpoonright \mathcal{A}_P \notin T_{[[p_P]]^*}$, which implies that $P' = \emptyset$ and so $R' \setminus P' \in F_{R \setminus P}$ by condition F1 of Definition 20.

Now assume that $i \in \mathcal{A}_R \setminus \mathcal{A}_P$. Then specifically $i \in \mathcal{A}_R^I \setminus \mathcal{A}_P$. From knowing that $t' \upharpoonright \mathcal{A}_P \in T_{[[p_P]]^*}$ and $t' \notin F_{\mathcal{E}([[r_R]]^*)}$, while $t'i \upharpoonright \mathcal{A}_P \notin T_{[[p_P]]^*}$ or $t'i \in F_{\mathcal{E}([[r_R]]^*)}$, it must be that $t'i \in F_{\mathcal{E}([[r_R]]^*)}$ as $t' \upharpoonright \mathcal{A}_P = t'i \upharpoonright \mathcal{A}_P$. But then by condition F2 it follows that $R' \cap F_{\mathcal{E}(R_\perp)} \neq \emptyset$ since $R' = \text{succ}_R^a(R)$, hence $R' \setminus P' \in F_{R \setminus P}$, which is contradictory.

Consequently, it must be the case that $t'i \in F_{[[r_R \setminus p_P]]^*}$ as required.

Now suppose that $t'i \in F_{[[r_R \setminus p_P]]^*}$. If $t' \in F_{[[r_R \setminus p_P]]^*}$, then by the induction hypothesis we know that $t' \in F_{[[r_R]]^* \setminus [[p_P]]^*}$, hence $t'i \in F_{[[r_R]]^* \setminus [[p_P]]^*}$ as $[[r_R]]^* \setminus [[p_P]]^*$ is a declarative specification. So suppose that $t' \notin F_{[[r_R \setminus p_P]]^*}$. Then $r_R \setminus p_P \xrightarrow{t'}_{R \setminus P} R \setminus P$ with $R \setminus P \notin F_{R \setminus P}$, and either: (i) $R \setminus P \xrightarrow{i}_{R \setminus P} R' \setminus P'$ with $R' \setminus P' \in F_{R \setminus P}$; or (ii) $R \setminus P \not\rightarrow_{R \setminus P}^i$.

Suppose for a contradiction that $t'i \notin F_{[[r_R]]^* \setminus [[p_P]]^*}$. We know $t'i \in T_{[[r_R]]^* \setminus [[p_P]]^*}$, hence it must be the case, by Definition 8, that $t'i \upharpoonright \mathcal{A}_P \in T_{[[p_P]]^*}$ and $t'i \notin F_{\mathcal{E}([[r_R]]^*)}$. Consequently, $t'i \upharpoonright \mathcal{A}_P \notin F_{[[p_P]]^*}$ and $t'i \in T_{[[r_R]]^*}$.

By this previous statement, under assumption (i) it follows $R' \setminus P'$ cannot be in $F_{R \setminus P}$ by conditions F1 or F2. As a result, there must be a $t'' \in (\mathcal{A}_{R \setminus P}^I)^*$ and an $i' \in \mathcal{A}_{R \setminus P}^I$ such that $R' \setminus P' \xrightarrow{t''}_{R \setminus P} R'' \setminus P'' \xrightarrow{i'}_{R \setminus P} R''' \setminus P'''$ with $R''' \setminus P''' \in F_{R \setminus P}$ by conditions F1 or F2 and no previous state in $F_{R \setminus P}$. By Definition 8, we know that $t'it''i' \upharpoonright \mathcal{A}_P \notin T_{[[p_P]]^*}$ or $t'it''i' \in T_{[[r_R]]^*}$, and $t'it''i' \upharpoonright \mathcal{A}_P \notin F_{[[p_P]]^*}$ or $t'it''i' \in F_{\mathcal{E}([[r_R]]^*)}$.

Now perform a case analysis on how the i' -transition could have arisen, as according to Definition 20.

If $i' \in \mathcal{A}_P^I \cap \mathcal{A}_R^I$, then we know $\text{succ}_R^{i'}(R'') \cap F_{\mathcal{E}(R_\perp)} = \emptyset$. Hence $R''' \cap F_{\mathcal{E}(R_\perp)} = \emptyset$, meaning $t'it''i' \notin F_{\mathcal{E}([[r_R]]^*)}$. Therefore, $t'it''i' \upharpoonright \mathcal{A}_P \notin F_{[[p_P]]^*}$, which means that $R''' \setminus P'''$ is not in $F_{R \setminus P}$ by condition F2. It therefore must be the case that $P''' = \emptyset$ or $R''' = \emptyset$. But by input-enabledness on P_\perp and R_\perp , it follows that $P''' = \emptyset$ or $R''' = \emptyset$. Consequently, $R'' \setminus P'' \in F_{R \setminus P}$ by condition F1, which is contradictory.

So instead consider the case of $i' \in \mathcal{A}_P^O \cap \mathcal{A}_R^O$. Then it is known that $\text{succ}_P^{i'}(P'') \neq \emptyset$ and $\text{succ}_R^{i'}(R'') \cap F_{\mathcal{E}(R_\perp)} = \emptyset$. Consequently, $t'it''i' \upharpoonright \mathcal{A}_P \in T_{[[pp]]^*}$, which means that $t'it''i' \in T_{[[rR]]^*}$. Therefore $P''' \neq \emptyset$ and $R''' \neq \emptyset$. So $R''' \setminus P'''$ is not inconsistent by condition F1. Now $\text{succ}_R^{i'}(R'') \cap F_{\mathcal{E}(R_\perp)} = \emptyset$ implies that $t'it''i' \notin F_{\mathcal{E}([rR]^*)}$, hence it must be the case that $t'it''i' \upharpoonright \mathcal{A}_P \notin F_{[[pp]]^*}$. But then $R''' \setminus P'''$ can also not be in $F_{R \setminus P}$ by condition F2, which is contradictory.

So finally consider $i' \in \mathcal{A}_R^I \setminus \mathcal{A}_P$. Then $R''' \setminus P'''$ being in $F_{R \setminus P}$ by $P''' = \emptyset$ or $P''' \cap F_{P_\perp} \neq \emptyset$ implies that $R''' \setminus P''' \in F_{R \setminus P}$, which would be contradictory. Now as $R'' \neq \emptyset$, certainly $R''' \neq \emptyset$ by input-enabledness of R_\perp . Therefore it must be the case that $R''' \cap F_{\mathcal{E}(R_\perp)} \neq \emptyset$, but this contradicts the condition for the existence of the i' -transition.

Now we consider case (ii), when $R \setminus P \not\stackrel{i}{\Rightarrow}_{R \setminus P}$. Notice by Lemma 4 that $[[rR \setminus pP]]^* = [[(rR \setminus pP)_\perp]]^*$. But in this error-completed Logic IOLTS it must be the case that $R \setminus P \stackrel{i}{\Rightarrow}_{R \setminus P}$, so the result follows by case (i).

Consequently, it must hold that $t'i \in F_{[[rR]]^* \setminus [[pP]]^*}$.

Showing that $t'i \in T_{[[rR]]^* \setminus [[pP]]^*}$ iff $t'i \in T_{[[rR \setminus pP]]^*}$ is similar.

C Proofs for Full-Abstraction Results

Proof 12 (Theorem 12) First suppose that $\mathcal{P} \equiv_{dec} \mathcal{Q}$. Then $\mathcal{A}_P^I = \mathcal{A}_Q^I$ and $\mathcal{A}_P^O = \mathcal{A}_Q^O$. So for any \mathcal{R} such that $(\mathcal{A}_P^O \cup \mathcal{A}_Q^O) \cap \mathcal{A}_R^O = \emptyset$ it follows that $\mathcal{A}_P^I \cap \mathcal{A}_R^O = \mathcal{A}_Q^I \cap \mathcal{A}_R^O$. As $\mathcal{P} \sqsubseteq_{dec} \mathcal{Q}$ and $\mathcal{Q} \sqsubseteq_{dec} \mathcal{P}$ it follows by Theorem 1 that $\mathcal{P} \parallel \mathcal{R} \sqsubseteq_{dec} \mathcal{Q} \parallel \mathcal{R}$ and $\mathcal{Q} \parallel \mathcal{R} \sqsubseteq_{dec} \mathcal{P} \parallel \mathcal{R}$. Hence $\mathcal{P} \parallel \mathcal{R} \equiv_{dec} \mathcal{Q} \parallel \mathcal{R}$, and so $F_{\mathcal{E}(\mathcal{P} \parallel \mathcal{R})} = F_{\mathcal{E}(\mathcal{Q} \parallel \mathcal{R})}$, which means that $\epsilon \in F_{\mathcal{E}(\mathcal{P} \parallel \mathcal{R})} \iff \epsilon \in F_{\mathcal{E}(\mathcal{Q} \parallel \mathcal{R})}$. Therefore, $\mathcal{E}(\mathcal{P} \parallel \mathcal{R}) \equiv_{dec}^F \mathcal{E}(\mathcal{Q} \parallel \mathcal{R})$.

Instead suppose that $\mathcal{P} \not\equiv_{dec} \mathcal{Q}$. If $\mathcal{A}_P^I \neq \mathcal{A}_Q^I$ or $\mathcal{A}_P^O \neq \mathcal{A}_Q^O$, then it is easy to formulate a process \mathcal{R} such that $\mathcal{E}(\mathcal{P} \parallel \mathcal{R})$ and $\mathcal{E}(\mathcal{Q} \parallel \mathcal{R})$ fail the \equiv_{dec}^F test on the alphabet check. So suppose that $\mathcal{A}_P^I = \mathcal{A}_Q^I$ and $\mathcal{A}_P^O = \mathcal{A}_Q^O$. Then $F_{\mathcal{E}(\mathcal{P})} \neq F_{\mathcal{E}(\mathcal{Q})}$ or $T_{\mathcal{P}} \neq T_{\mathcal{Q}}$.

Suppose the former, then without loss of generality there exists some $t \in F_{\mathcal{E}(\mathcal{P})}$ such that $t \notin F_{\mathcal{E}(\mathcal{Q})}$. So construct a process \mathcal{R} in which $\mathcal{A}_R^O = \mathcal{A}_P^I$ and $\mathcal{A}_R^I = \mathcal{A}_P^O$ in order to distinguish \mathcal{P} and \mathcal{Q} . Take $F_{\mathcal{R}} = \emptyset$ and $T_{\mathcal{R}}$ as the smallest declarative specification containing t . Then $t \in F_{\mathcal{E}(\mathcal{P} \parallel \mathcal{R})}$ and as t is a trace over $\mathcal{A}_{\mathcal{P} \parallel \mathcal{R}}^O$ it follows that $\epsilon \in F_{\mathcal{E}(\mathcal{P} \parallel \mathcal{R})}$. However, as $t \notin F_{\mathcal{E}(\mathcal{Q})}$ it follows that no prefix of t is in $F_{\mathcal{E}(\mathcal{Q})}$. In particular, $\epsilon \notin F_{\mathcal{E}(\mathcal{Q})}$. As $\epsilon \notin F_{\mathcal{R}}$ it follows that $\epsilon \notin F_{\mathcal{E}(\mathcal{Q} \parallel \mathcal{R})}$ and so $\mathcal{E}(\mathcal{P} \parallel \mathcal{R}) \not\equiv_{dec}^F \mathcal{E}(\mathcal{Q} \parallel \mathcal{R})$ as required.

Instead suppose that $F_{\mathcal{E}(\mathcal{P})} = F_{\mathcal{E}(\mathcal{Q})}$ and $T_{\mathcal{P}} \neq T_{\mathcal{Q}}$. Then without loss of generality there exists some $t \in T_{\mathcal{P}}$ such that $t \notin T_{\mathcal{Q}}$. So there exists a prefix $t'a$ of t such that $t' \in T_{\mathcal{Q}}$, but $t'a \notin T_{\mathcal{Q}}$. Moreover, it must be the case that $t \notin F_{\mathcal{E}(\mathcal{P})}$ and $t \notin F_{\mathcal{E}(\mathcal{Q})}$. So construct a process \mathcal{R} in which $\mathcal{A}_R^O = \mathcal{A}_P^I$ and $\mathcal{A}_R^I = \mathcal{A}_P^O$ in order to distinguish \mathcal{P} and \mathcal{Q} . Take $F_{\mathcal{R}} = \{t'' \in \mathcal{A}_{\mathcal{R}}^* : t'a \text{ is a prefix of } t''\}$ and $T_{\mathcal{R}}$ as the smallest declarative specification containing $F_{\mathcal{R}}$. Then clearly $t'a \in F_{\mathcal{P} \parallel \mathcal{R}}$ as $t'a \in F_{\mathcal{R}}$ and $t'a \in T_{\mathcal{P}}$. As $t'a$ is a trace over $\mathcal{A}_{\mathcal{P} \parallel \mathcal{R}}^O$, it follows

that $\epsilon \in F_{\mathcal{E}(\mathcal{P} \parallel \mathcal{R})}$. Now, as $t'a \notin F_{\mathcal{E}(\mathcal{Q})}$ it is known that no prefix of $t'a$ is in $F_{\mathcal{Q}}$. Moreover, as $t'a \notin T_{\mathcal{Q}}$, it follows that $t'a \notin F_{\mathcal{Q} \parallel \mathcal{R}}$ and so $t'a \notin F_{\mathcal{E}(\mathcal{Q} \parallel \mathcal{R})}$. Therefore, $\epsilon \notin F_{\mathcal{E}(\mathcal{Q} \parallel \mathcal{R})}$ and so $\mathcal{E}(\mathcal{P} \parallel \mathcal{R}) \not\equiv_{dec}^F \mathcal{E}(\mathcal{Q} \parallel \mathcal{R})$.

Proof 13 (Corollary 1) *By Theorems 1, 3 and 6 we know that \equiv_{dec} is a congruence (subject to compatibility) for parallel composition, conjunction and quotient. Theorem 12 tells us that \equiv_{dec} is the coarsest such relation.*

Proof 14 (Theorem 13) *Given the relationship between \equiv_{op} and \equiv_{dec} , it is sufficient to show that $\mathcal{E}(p_{\mathcal{P}} \parallel r_{\mathcal{R}}) \equiv_{op}^F \mathcal{E}(q_{\mathcal{Q}} \parallel r_{\mathcal{R}})$ iff $\llbracket \mathcal{E}(p_{\mathcal{P}} \parallel r_{\mathcal{R}}) \rrbracket^* \equiv_{dec}^F \llbracket \mathcal{E}(q_{\mathcal{Q}} \parallel r_{\mathcal{R}}) \rrbracket^*$. So without loss of generality, think of $\mathcal{E}(p_{\mathcal{P}} \parallel r_{\mathcal{R}})$ as some Logic IOLTS $s_{\mathcal{S}}$. Then we wish to show that $s_{\mathcal{S}} \in F_{\mathcal{S}}$ iff $\epsilon \in \llbracket s_{\mathcal{S}} \rrbracket^*$. But this follows immediately from Definition 13.*

Proof 15 (Corollary 2) *By Theorems 1, 3 and 6 we know that \equiv_{dec} is a congruence (subject to compatibility) for parallel composition, conjunction and quotient. Theorems 8, 9 and 11 imply that these operators are also congruences in the operational setting for \equiv_{op} . Theorem 13 tells us that \equiv_{op} is the coarsest such relation.*