Scientific Annals of Computer Science vol. 28 (2), 2018, pp. 289–337
doi: 10.7561/SACS.2018.2.289

Fault-Free Refinements for Interface Automata [1]

Ayleen Schinko[2], Walter Vogler[2]

Abstract

A refinement preorder for a model of concurrent systems should be compositional (i.e. a precongruence for parallel composition) and should not introduce faults into a fault-free specification. Arguably, the coarsest such precongruence is the optimal refinement preorder. For the model of interface automata, faults are communication errors in the form of unexpected inputs. The respective optimal preorder has been characterized as the inclusion of two trace sets. Here, we extend this result by regarding also quiescence (quiescence and divergence resp.) as faults. The latter preorder is coarser, i.e. better, than an earlier preorder regarding errors, quiescence and divergence. We also present conjunction operators for our settings, avoiding flaws that can be found in the literature, and a quotient operator.

Keywords: refinement, precongruence, conjunction, quotient, quiescence, divergence

1 Introduction

Interface automata (IA) [6, 7] describe how a system component performs input actions and the locally controlled output and internal actions. Arrival

[1] We dedicate this paper to Maciej Koutny on the occasion of his 60th birthday. The second author thanks Maciej for the enjoyable cooperation on writing two papers together and for all the pleasant encounters over so many years, which took place mostly in the context of our common interest in Petri nets.
This work was partially supported by the DFG-project 'Foundations of Heterogenous Specifications Using State Machines and Temporal Logic' VO 615/12-2. An extended abstract is to appear in Proc. ACSD 2018.

[2] Inst. f. Informatik, Universität Augsburg, D-86135 Augsburg
Email: {ayleen.schinko,walter.vogler}@informatik.uni-augsburg.de

of an unexpected input leads to a communication error, and such errors have to be avoided. To support the design of error-free communicating systems, interface automata are equipped with a parallel composition featuring a specific error pruning and with an alternating simulation as refinement relation. Both are somewhat arbitrary. To avoid prejudices as far as possible, we set out in [4] to find an adequate observable semantics for interface automata as follows.

To avoid preconceptions, we used a standard parallel composition; additionally, states with an error are marked, but no other modification like pruning is made without justification. Our model is called *error-IO transitions system (EIO)*. As a basic observable we took whether an interface automaton can run into an error on its own, i.e. whether an error is reachable by local actions alone.[3] Accordingly, a basic requirement for a refinement preorder is that a specification without such a locally reachable error cannot be refined by an interface automaton with such an error. Technically speaking, the preorder should refine (be contained in) the relation $\sqsubseteq_E^{\mathrm{B}}$, where $S \sqsubseteq_E^{\mathrm{B}} S'$ unless S has a locally reachable error while S' has not. Relation $\sqsubseteq_E^{\mathrm{B}}$ describes ***b***asic ***e***rror avoidance.

A second essential requirement is that the refinement preorder is compositional w.r.t. parallel composition (and possibly other operational connectives). Formally, this means that the preorder is a *precongruence* w.r.t. parallel composition, i.e. refining a component of a composed system refines the overall system. Given a basic requirement as expressed by $\sqsubseteq_E^{\mathrm{B}}$, we regard a preorder as *optimal* if it refuses an automaton as refinement only if this is necessary to achieve these two goals; technically, the preorder should be the coarsest precongruence refining $\sqsubseteq_E^{\mathrm{B}}$, called *fully abstract* w.r.t. $\sqsubseteq_E^{\mathrm{B}}$ and parallel composition. This precongruence can also be understood as being obtained from the basic observable when putting the specification and the potential refinement into all possible parallel contexts or testing environments.

For the setting above, we characterized the fully abstract precongruence as inclusion for two trace sets; this corresponds to the declarative model of [9], which was developed to study the behaviour of asynchronous circuits. One trace set is the set of *error traces*. Essentially, these are the traces leading to an error and the *continuations* of such traces, but also some pruning on traces is used. The other set is the union of the first set and the

[3] Considering only local reachability is usually called optimistic. In [4] also two other variations were studied that are called hyper-optimistic and pessimistic.

language of the automaton. Although the precongruence was designed to ensure that new errors are not introduced during refinement, it is concerned with safety in a broader sense: also error-free behaviour of a refinement must be allowed in the specification.

We found that the pruning of [6,7] leads to an equivalent EIO w.r.t. the precongruence, if the EIO is deterministic on inputs. The latter is required in most papers about interface automata, but not in [6], where the parallel composition with pruning – which is only based on intuition – fails to be associative. We also showed how pruning can be modified to work properly in this general case.

A problem in interface automata is that outputs are just seen as a cause for errors; one can refine each automaton by one that never performs any output. This is most often not desirable. To remedy it, one can complement the model by may- and must-modalities on the transitions, see e.g. [3] for an advanced extension of this kind. Alternatively, one can introduce an artificial output action δ and add a δ-loop to each state that does not enable any local action; δ indicates that the automaton cannot leave such a so-called *quiescent* state on its own. Then, the alternating simulation of [7] makes sure that, whenever a refinement cannot perform any output in some state, the same holds for each simulating state in the specification; cf. [1]. This construction is known from the ioco-approach (input/output conformance) of Tretmans [14]. One can also regard quiescent states as faulty as in [5]. Additionally, *divergent states*, which enable an infinite sequence of internal transitions, are seen as undesirable there. A motivation for this view is that the automaton may e.g. block due to a never-ending and energy consuming internal computation.

Motivated by [5], we continue our work from [4] in the present paper as follows. In a first stage, we add to $\sqsubseteq^{\mathrm{B}}_{=E}$ the requirement that locally reachable quiescence be not introduced in a refinement step, obtaining the basic preorder $\sqsubseteq^{\mathrm{B}}_{=Qui}$. This requirement enforces some local actions in the refining system, the resulting precongruence preserves liveness in some sense. In a second stage, the same requirement for divergent states is added to $\sqsubseteq^{\mathrm{B}}_{=Qui}$. For both settings of faulty states, we characterize the resp. fully abstract precongruence, adding a set of quiescent traces for the first and, additionally, adding divergence traces to the error traces considered in [4] for the second. We also show that all the precongruences are compositional for hiding.

In this paper, we follow an interleaving approach. For a treatment of failures involving so-called true concurrency and considering causality, we refer e.g. to [11].

It has become popular in recent years to consider also conjunction for operational models. At first glance, this might look surprising; but it is actually a natural concept since we consider such models as specifications that are satisfied or not by other models. A conjunction operator allows to specify different facets of an intended system separately and, then, to combine them; see [10] for a fairly early contribution and the discussion of earlier work therein. Another interesting operator is quotienting, which shows how to implement a specification as a parallel composition using an existing component. We also exhibit conjunction operators for our precongruences and a quotient operator.

In a setting that disregards errors, quiescence was already studied in [13] with a testing approach in the spirit of [8]; this kind of testing is closely related to the coarsest-precongruence idea. The quiescence semantics in [13] coincides with ours for error-free EIOs. The closest publication to the present paper is [5], where also related work is discussed more extensively. There, a setting just concerning errors and a setting concerning errors, quiescence and divergence are studied. The starting point for both is declarative, i.e. considering trace sets without an operational model. The first setting is just the same as the one in [9]. For the second, a set of quiescent traces and a set of divergence traces are added. Then, both settings are underlaid with an operational model like EIO.

Essentially, the coarsest precongruence result of [4] can also be found in [5]. Nevertheless, we will prove it again in the present paper. The first reason for this is that our EIOs are slightly more prejudiced than the ones in [4]; this makes concepts and proofs much easier and accessible here. All our results (except those concerning conjunction and quotient) are shown to hold also for the original EIO model in [12]. Furthermore, binary synchronization is considered in [4], whereas we have multicast here. The second reason is that the automata in [5] are slightly more prejudiced than ours. Refinement is defined as inclusion for the same trace sets we use, but the trace sets used to show compositionality w.r.t. parallel composition in [5] are not the same. Furthermore, the full abstractness result is for an equivalence and not for a preorder. Thus, our proofs are different, and they are more detailed. But the main reason for studying the basic setting again is that it gives an easy access to the whole approach, and the proofs are needed

anyway for our two new settings.

The authors of [5] insist that a proper treatment of quiescence requires to treat divergence as undesirable as well. Our first new setting studies quiescence ignoring divergence. Furthermore, and in contrast to [5], we have a full abstractness result for our second setting treating quiescence and divergence. This shows that the observable precongruence based on observing errors, divergence and quiescence is coarser than the precongruence in [5] and, hence, better in the sense of the above optimality. Technically, the semantics for the latter has a separate set of divergence traces, which is not closed under continuation. Our semantics has one set for error and divergence traces, i.e. error and divergence states have the same impact, and the whole set is closed under continuation.

For the two settings in [5], also conjunction (and disjunction) as well as a quotient w.r.t. parallel composition are constructed, turning them into what is often called interface theories. Our results demonstrate that our new settings can serve as interface theories as well. We also point out mistakes in the conjunctions of [5].

Section 2 gives the basic definitions. Then, we study the coarsest precongruences, extending the observable faults stepwise: Section 3 considers only errors, and we add quiescence in Section 4 and divergence in Section 5. The results on conjunctions can be found in Section 4 and 5, and a comparison of the various refinement notions (including the one from [5]) is presented in Section 5. Section 6 studies the quotient operator, and we finish with some conclusions.

2 Basic Notions

We consider labelled transition systems where each transition is labelled with an input or output action or with the invisible, *internal action* τ, which is different from all other actions. Systems communicate by performing the same action, which is an input of one and an output of the other system. If, in a state of a composed system, one component generates an output action for another component that is not ready to receive this input, then a catastrophic error arises and the state is called an *error state*. Therefore, our systems have distinguished sets of error states. If a state is not ready to receive an input i, the state would not have an i-transition in most IA-like approaches. Here, we follow [5] and give the state an i-transition to an error

state. Consequently, our systems are input-enabled; we will explain this further in Section 2.1.

Definition 1 (Error-IO Transition Systems)**.** *An* error-IO transition system (EIO) *is a tuple* $S = (Q, I, O, \delta, q_0, E)$ *with:*

- Q *– a set of* states*,*
- I, O *– disjoint sets of* input *and* output actions*, where* $\Sigma = I \cup O$ *is the* action set *and* $\mathrm{Sig}(S) = (I, O)$ *the* signature *of* S*,*
- $\delta \subseteq Q \times (\Sigma \cup \{\tau\}) \times Q$ *– the transition relation,*
- $q_0 \in Q$ *– the* initial *state,*
- $E \subseteq Q$ *– the set of* error states*.*

We require that S *is* input-enabled*, i.e. for all* $p \in Q$ *and* $i \in I$ *there exists* $q \in Q$ *with* $(p, i, q) \in \delta$*. We call an EIO* deterministic*, if it has no* τ*-transitions and, for each* $a \in \Sigma$*, each state has at most one* a*-transition.*

The idea of an EIO is that outputs and internal actions are under the control of the system, they are called *local*. In contrast, input transitions are only performed if the input is provided by an environment. If not defined otherwise, an EIO S always has the components Q, I, O, δ, q_0 and E, and similarly S_1 has components Q_1, I_1 etc. This convention also applies to the language of an EIO as defined below and for similar constructs. In pictures, we write $x?$ for an input x and $x!$ for an output x. An x without ? or ! denotes an arbitrary visible action.

For an EIO S, we derive the following notations from δ: We write $p \xrightarrow{\alpha} q$ for $(p, \alpha, q) \in \delta$ and $p \xrightarrow{\alpha}$ for $\exists q \in Q : p \xrightarrow{\alpha} q$, saying that α is *enabled* in p. A state is *stable* if it does not enable τ. Extending the notation to action sequences, $p \xrightarrow{w} q$ means that there exists a *run* $p \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} p_2 \ldots \xrightarrow{\alpha_n} q$ such that $w = \alpha_1 \alpha_2 \ldots \alpha_n$ where $\alpha_i \in (\Sigma \cup \{\tau\})$ for $i = 1, \ldots, n$; a run can also be infinite. A state q is *reachable* if $q_0 \xrightarrow{w} q$ for some w.

The projection $w|_B$ of w onto B arises from w by deleting all actions not in $B \subseteq \Sigma$. Now $p \overset{w}{\Rightarrow} q$ if $w \in \Sigma^*$ and $\exists w' \in (\Sigma \cup \{\tau\})^* : w'|_\Sigma = w \wedge p \xrightarrow{w'} q$; we say that the latter run *underlies* $p \overset{w}{\Rightarrow} q$ or just w, if the context is clear. As above, we write $p \xrightarrow{w}$ for $\exists q : p \xrightarrow{w} q$ and $p \overset{w}{\Rightarrow}$ for $\exists q : p \overset{w}{\Rightarrow} q$.

The *language* of S is $L(S) = \left\{ w \in \Sigma^* \mid q_0 \overset{w}{\Rightarrow} \right\}$; it consists of the *traces* of S.

When building specifications in a modular way, the main operators are parallel composition and some form of scoping that prevents communication on some actions. If such an action is an input, the action is blocked since it cannot be triggered by the environment anymore; this does not interest us here. In contrast, an output is locally controlled, i.e. never blocked; if communication is prevented, it is performed invisibly. Such a hiding turns some outputs into τ.

Definition 2 (Hiding)**.** *For an EIO* $S = (Q, I, O, \delta, q_0.E)$ *and some* $X \subseteq O$*,* S hiding X *is the EIO* $S/X = (Q, I, O', \delta', q_0, E)$ *where* $O' = O \setminus X$ *and* δ' *is obtained from* δ *by replacing all transition labels in* X *by* τ*.*

As in IA, component systems working in parallel synchronize on common visible actions. Since outputs are controlled by the resp. component, components cannot have an output in common. In IA, always an input and an output are synchronized (and then hidden); we, as others, allow multicast communication: an output can be received by several components, and these can consequently synchronize on each common input. Since errors are catastrophic, a state of a composition is an error if one of its component states is an error, i.e. this error is inherited from a component.

Definition 3 (Parallel Composition)**.** *EIOs* S_1 *and* S_2 *are* composable*, if* $O_1 \cap O_2 = \emptyset$*. In this case, we define their* parallel composition $S_{12} := S_1 \| S_2 = (Q, I, O, \delta, q_0, E)$ *as follows.*

- $Q = Q_1 \times Q_2$,
- $I = (I_1 \backslash O_2) \cup (I_2 \backslash O_1)$,
- $O = O_1 \cup O_2$,
- $q_0 = (q_{01}, q_{02})$,
- $\delta = \{((q_1, q_2), \alpha, (p_1, q_2)) \mid (q_1, \alpha, p_1) \in \delta_1, \alpha \in (\Sigma_1 \cup \{\tau\}) \setminus \Sigma_2\}$
  $\cup \{((q_1, q_2), \alpha, (q_1, p_2)) \mid (q_2, \alpha, p_2) \in \delta_2, \alpha \in (\Sigma_2 \cup \{\tau\}) \setminus \Sigma_1\}$
  $\cup \{((q_1, q_2), \alpha, (p_1, p_2)) \mid (q_1, \alpha, p_1) \in \delta_1, (q_2, \alpha, p_2) \in \delta_2,$
  $\alpha \in \Sigma_1 \cap \Sigma_2\}$,
- $E = (Q_1 \times E_2) \cup (E_1 \times Q_2)$

We use the above notation $S_{12} = S_1 \| S_2$ *in an analogous way for other systems, e.g.* $S_{ij} := S_i \| S_j$ *for* $i, j \in \mathbb{N}$*. We call* S_1 *a* partner *of* S_2 *if*

$I_2 \subseteq O_1$ and $O_2 = I_1$; intuitively, S_1 fully synchronises with S_2 but might have additional outputs.

Parallel composition can also be defined on the traces of two EIOs or on arbitrary words over alphabets Σ_1 and Σ_2.

Definition 4 (Parallel Composition of Traces)**.** *Let S_1 and S_2 be two EIOs.*

- *The* parallel composition *of words $w_1 \in \Sigma_1^*$ and $w_2 \in \Sigma_2^*$ is $w_1 \| w_2 := \{w \in (\Sigma_1 \cup \Sigma_2)^* \mid w|_{\Sigma_1} = w_1 \wedge w|_{\Sigma_2} = w_2\}$.*
- *The* parallel composition *of two languages (sets of words) $W_1 \subseteq \Sigma_1^*$ and $W_2 \subseteq \Sigma_2^*$ is $W_1 \| W_2 := \bigcup \{w_1 \| w_2 \mid w_1 \in W_1 \wedge w_2 \in W_2\}$.*

The following lemma is well-known.

Lemma 5. *Let S_1 and S_2 be composable EIOs.*

1. *Let $w \in \Sigma_{12}^*$, $w_1 = w|_{\Sigma_1}$ and $w_2 = w|_{\Sigma_2}$; let $(q_1, q_2), (p_1, p_2) \in Q_{12}$. Then $(q_1, q_2) \overset{w}{\Rightarrow} (p_1, p_2)$ if and only if $q_1 \overset{w_1}{\Rightarrow} p_1$ and $q_2 \overset{w_2}{\Rightarrow} p_2$.*
2. *$L_{12} = L_1 \| L_2$*

We call the second and third (underlying) run in Part 1 (of Lemma 5) the *projections* of the first. Each visible action in one of the former two corresponds to a unique action in the latter. For prefixes v_1 of w_1 and v_2 of w_2, we say that v_1 *ends before, with* or *after* v_2 according to the positions of their last actions in w. Each prefix v of w determines prefixes v_1 of w_1 and v_2 of w_2 with $v \in v_1 \| v_2$; if the two equivalent statements in Part 1 hold, these three prefixes determine prefixes of the three runs that also make the statements true.

We will also have a quick look at the parallel composition used in [7]. There, communication is binary, i.e. an output can only synchronize with one input. Consequently, we call EIOs S_1 and S_2 *strongly composable*, if $\Sigma_1 \cap \Sigma_2 = (I_1 \cap O_2) \cup (O_1 \cap I_2)$, i.e. $(O_1 \cap O_2) = \emptyset = (I_1 \cap I_2)$.

Definition 6 (Parallel Composition With Internalization)**.** *For strongly composable EIOs S_1 and S_2, their parallel composition with hiding is defined as $S_1 | S_2 = S_{12}/(\Sigma_1 \cap \Sigma_2)$.*

2.1 Communication Errors and Basic Requirements

In our setting, the reaction to an arriving input is always specified due to input-enabledness; an input transition leading to an error state means that the system is not able to deal with this input properly at the respective state and a catastrophic error occurs. Such an error is an unavoidable problem for a system S only if it is *locally reachable*, i.e. if $q_0 \stackrel{w}{\Rightarrow} q$ with $q \in E$ for some $w \in O^*$. If S does not have such errors, we say for short that S *avoids errors*.

If $(q_1, q_2) \in Q_1 \times Q_2$ is a state of a composition, then there might be an output transition $q_1 \stackrel{a}{\to} q_1'$ for an input a of S_2 that, in the IA-setting, is not enabled in q_2. In this case, (q_1, q_2) is designated as an error state – and we called it a new error in [4]. In the present setting, we have some $q_2 \stackrel{a}{\to} q_2'$ with $q_2' \in E_2$ instead; hence, there is the output transition $(q_1, q_2) \stackrel{a}{\to} (q_1', q_2')$, and $(q_1', q_2') \in E_{12}$ is called an inherited error in [4]. Since a is an output, the error is already unavoidable in (q_1, q_2) also in the present setting. In other words, input-enabledness does not really change the setting intuitively. Strictly speaking, this is a prejudice; it is formally justified in [12], where all our results except for the ones concerning conjunction and quotient are proven without input-enabledness. The advantage of input-enabledness is that we have only one kind of error in Def. 3. Accordingly, the definition of error traces below as well as the subsequent proofs become considerably easier. The operational model in [5] is even more prejudiced since it requires each error state to have a loop-transition for each action.

In a refinement framework, one clearly does not want to introduce an error in a refinement step. To phrase this more technically, we write $S_1 \sqsubseteq_E^{\mathrm{B}} S_2$ for EIOs S_1 and S_2 with the same signature, if S_1 avoids errors provided S_2 does. Then one essential requirement for a refinement preorder is: if S_1 refines S_2, then $S_1 \sqsubseteq_E^{\mathrm{B}} S_2$. A second requirement is that the preorder supports modular reasoning, i.e. that it be a precongruence for parallel composition. Now, a preorder is semantically optimal if it rejects a system as a refinement only if these two requirements make it necessary. In other words, one should look for the (signature-preserving) ***c***oarsest precongruence $\sqsubseteq_E^{\mathrm{C}}$ for $\|$ contained in the *basic preorder* $\sqsubseteq_E^{\mathrm{B}}$. This precongruence is called *fully abstract* w.r.t. $\sqsubseteq_E^{\mathrm{B}}$ and $\|$, and was characterized with two trace sets in [4, 5].

In the present paper, we will extend this approach to deal with further semantical issues. A state q is potentially problematic, if it is *quiescent*, i.e.

has only input-transitions. On its own, S cannot progress and, hence, is deadlocked in such a state. Furthermore, a state q is sometimes considered to be as catastrophic as an error if it is *divergent*, i.e. some infinite run of τ-transitions starts at q; the idea is that S can block any communication since it gets stuck in this internal run. Our new approaches will show how to avoid faults during the design process, where faults are either error and quiescent states or error, quiescent and divergent states. Formally:

Definition 7 (Faults, Preorders)**.** *For an EIO S, Qui (or Qui_S) is the set $\{q \in Q \mid \forall \alpha \in O \cup \{\tau\} : q \stackrel{\alpha}{\not\to}\}$ of its* quiescent *states and Div (or Div_S) is the set $\{q \in Q \mid q$ has an infinite run of τ transitions$\}$ of its* divergent *states.*

We say that S avoids quiescence *(*divergence *resp.) if no quiescent (divergent resp.) state is locally reachable.*

*For EIOs S_1 and S_2 with the same signature, we write $S_1 \sqsubseteq^{\mathrm{B}}_{Qui} S_2$, ($S_1 \sqsubseteq^{\mathrm{B}}_{Div} S_2$) if S_1 avoids errors and **qui**escence (errors, quiescence and **div**ergence), provided S_2 does. Analogously to the above, $\sqsubseteq^{\mathrm{C}}_{Qui}$ ($\sqsubseteq^{\mathrm{C}}_{Div}$) is the fully abstract preorder for $\sqsubseteq^{\mathrm{B}}_{Qui}$ ($\sqsubseteq^{\mathrm{B}}_{Div}$) and parallel composition.*

The first main aim of this paper is to characterize $\sqsubseteq^{\mathrm{C}}_{Qui}$ and $\sqsubseteq^{\mathrm{C}}_{Div}$. The corresponding trace sets appearing in this paper are based on a pruning and a continuation operator defined as follows. Pruning reflects the intuition that e.g. an error has as good as occurred if it is locally reachable. The second precongruence in [5] differs from ours in that these two operators are not applied to the divergence traces there; this makes the precongruence in [5] unnecessarily discriminating. Note that we introduce pruning on traces just because this is adequate to deal with our $\|$, which does not involve any pruning itself.

Definition 8 (Pruning and Continuation Function)**.** *Let S be an EIO; with ε being the empty word and $\mathfrak{P}(M)$ denoting the powerset of a set M, we define:*

- $\mathrm{prune} : \Sigma^* \to \Sigma^*, w \mapsto u$, *with* $w = uv, u = \varepsilon \vee u \in \Sigma^* \cdot I$ *and* $v \in O^*$,
- $\mathrm{cont} : \Sigma^* \to \mathfrak{P}(\Sigma^*), w \mapsto \{wu \mid u \in \Sigma^*\}$,
- $\mathrm{cont} : \mathfrak{P}(\Sigma^*) \to \mathfrak{P}(\Sigma^*), L \mapsto \bigcup \{\mathrm{cont}(w) \mid w \in L\}$.

3 Preserving Error-Freedom

3.1 Characterizing $\sqsubseteq_E^{\mathrm{C}}$

In this section, we demonstrate our approach for the simple case that considers errors only. This gives an easy access to our approach, and we will reuse the proofs in later sections. The trace-based characterization of $\sqsubseteq_E^{\mathrm{C}}$ is defined as follows.

Definition 9 (Error Semantics)**.** *For an EIO S, we define the sets:*

- strict error traces*:* $StET(S) := \{w \in \Sigma^* \mid q_0 \stackrel{w}{\Rightarrow} q \text{ with } q \in E\}$,
- pruned error traces*:* $PrET(S) := \{\mathrm{prune}(w) \mid w \in StET(S)\}$,
- error traces $ET(S) := \mathrm{cont}(PrET(S))$.
- *The* error-flooded language *of S is* $EL(S) := L(S) \cup ET(S)$.

We call $(ET(S), EL(S))$ the error semantics *of S. For two EIOs S_1, S_2 with the same signature, we write $S_1 \sqsubseteq_E S_2$ if $ET_1 \subseteq ET_2$ and $EL_1 \subseteq EL_2$.*

Intuitively, it is clear that strict error traces are relevant if the reachability of an error is an issue. We have already argued that an error is as good as reached after a pruned strict error trace. Since an error is deadly, no further behaviour is relevant; to blur this behaviour, all continuations are added. Such a flooding is known from the treatment of divergence in [2]. Furthermore, if we want to know whether one system can still reach an error when being composed with another system, the (blurred) language of the latter is relevant.

Theorem 10 (Error Semantics for Parallel Composition)**.** *For two composable EIOs S_1, S_2 and their composition S_{12}, we have:*

1. $ET_{12} = \mathrm{cont}\,(\mathrm{prune}\,((ET_1 \| EL_2) \cup (EL_1 \| ET_2)))$,
2. $EL_{12} = (EL_1 \| EL_2) \cup ET_{12}$.

Proof: 1. „$\subseteq$“:
Since both sides of the equation are closed under continuation, it suffices to consider a prefix-minimal word w of ET_{12}, which from the definition is contained in $PrET_{12}$. Hence, there is some $v \in O_{12}^*$ such that $(q_{01}, q_{02}) \stackrel{w}{\Longrightarrow} (q_1, q_2) \stackrel{v}{\Rightarrow} (q_1', q_2')$ with $(q_1', q_2') \in E_{12}$ and $w = \mathrm{prune}(wv)$. Projecting

the underlying run in the vein of Lemma 5.1, we get $q_{01} \stackrel{w_1}{\Rightarrow} q_1 \stackrel{v_1}{\Rightarrow} q_1'$ und $q_{02} \stackrel{w_2}{\Rightarrow} q_2 \stackrel{v_2}{\Rightarrow} q_2'$ with $w \in w_1 \| w_2$ and $v \in v_1 \| v_2$. W.l.o.g. we assume that $q_1' \in E_1$. Hence, $w_1 v_1 \in StET_1 \subseteq \mathrm{cont}(PrET_1) = ET_1$. Since $q_{02} \stackrel{w_2 v_2}{\Longrightarrow}$, we infer $w_2 v_2 \in L_2 \subseteq EL_2$ and $wv \in ET_1 \| EL_2$. By $w = \mathrm{prune}(wv)$, we are done.

1. „$\supseteq$“:
Analogously, it suffices to consider a prefix-minimal word x of the r.h.s., and this is a pruned word. Hence, there is some $y \in O_{12}^*$ with $xy \in (ET_1 \| EL_2) \cup (EL_1 \| ET_2)$, and w.l.o.g. $xy \in ET_1 \| EL_2$. Due to Lemma 5, there are $w_1 \in ET_1$ and $w_2 \in EL_2$ with $xy \in w_1 \| w_2$. We will show that xy has a prefix $v' \in PrET_{12}$; since v' cannot end on an action in y, it is a prefix of x, and we are done.

Let v_1 be the shortest prefix of w_1 in $PrET_1$. If $w_2 \in ET_2$, let v_2' be the shortest prefix of w_2 in $PrET_2$ and assume by symmetry that it does not end before v_1 in xy. Otherwise, we let $v_2' = w_2$, and in both cases $v_2' \in L_2$.

The last action of v_1 determines a prefix v of xy and a prefix v_2 of v_2': hence, $v \in v_1 \| v_2$. On the one hand, $q_{02} \stackrel{v_2}{\Rightarrow} q_2$. On the other, $\exists u_1 \in O_1^* : q_{01} \stackrel{v_1}{\Rightarrow} q_1 \stackrel{u_1}{\Rightarrow} q_1'$ with $q_1' \in E_1$. By Lemma 5, $(q_{01}, q_{02}) \stackrel{v}{\Rightarrow} (q_1, q_2)$. By input-enabledness, all actions of u_1 that are inputs of S_2 can always be performed there. Hence, we can extend the latter run by $(q_1, q_2) \stackrel{u_1}{\Rightarrow} (q_1', q_2')$, and $(q_1', q_2') \in E_{12}$.

This implies $vu_1 \in ET_{12}$; $v' = \mathrm{prune}(vu_1)$ is in $PrET_{12}$ and a prefix of v and xy. Note that, in particular, the last action of v_1 and v might be an input in S_1 and an output in S_{12}. We are done.

2.: The proof for this item is essentially the same as in [4]. From the definitions, it is clear that $L_i \subseteq EL_i$ and $ET_i \subseteq EL_i$. To understand the arguments, read the chain of equations from the right.

$$
\begin{aligned}
(EL_1 \| EL_2) \cup ET_{12} &\stackrel{9}{=} ((L_1 \cup ET_1) \,\|\, (L_2 \cup ET_2)) \cup ET_{12} \\
&= (L_1 \| L_2) \cup \underbrace{(L_1 \| ET_2)}_{\substack{\subseteq (EL_1 \| ET_2) \\ \stackrel{1.}{\subseteq} ET_{12}}} \cup \underbrace{(ET_1 \| L_2)}_{\substack{\subseteq (ET_1 \| EL_2) \\ \stackrel{1.}{\subseteq} ET_{12}}} \cup \underbrace{(ET_1 \| ET_2)}_{\substack{\subseteq (EL_1 \| ET_2) \\ \stackrel{1.}{\subseteq} ET_{12}}} \cup ET_{12} \\
&= (L_1 \| L_2) \cup ET_{12} \\
&\stackrel{5}{=} L_{12} \cup ET_{12} \\
&\stackrel{9}{=} EL_{12}.
\end{aligned}
$$

□

Since cont, prune and $\|$ are monotonic on languages, this result implies:

Corollary 11 (Error-Precongruence)**.** *The relation $\sqsubseteq_E$ is a precongruence w.r.t. $\|$.*

The following lemma is the next step in proving our characterization result.

Lemma 12. *Let S_1 and S_2 be two EIOs with the same signature. If $U\|S_1 \sqsubseteq_E^{\mathrm{B}} U\|S_2$ for all partners U, then $S_1 \sqsubseteq_E S_2$.*

Proof: We write I for $I_1 = I_2$ and O for $O_1 = O_2$. Recall that for a partner U we have $I_U = O$ and $O_U \supseteq I$; in this proof, we will always have $O_U = I$. We first show $ET_1 \subseteq ET_2$. As above, it suffices to consider a prefix-minimal $w \in ET_1$.

- $w = \varepsilon$: In this case, S_1 has a locally reachable error. Let U have just one non-error state with a loop for all $x \in I_U$. Thus, S_1 can essentially reach the same states locally as $U\|S_1$, and also $U\|S_2$ can reach an error locally. This error can only stem from S_2, and $w \in ET_2$.

- $w = x_1 \dots x_n x_{n+1} \in \Sigma^+$ with $n \geq 0$ and $x_{n+1} \in I = O_U$: We construct the following partner U (see also Fig. 1):

  - $Q_U = \{q_0, q_1, \dots, q_{n+1}\}$,
  - $q_{0U} = q_0$,
  - $E_U = \emptyset$,
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i \leq n\}$
    $\cup \{(q_i, x, q_{n+1}) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i \leq n\}$
    $\cup \{(q_{n+1}, x, q_{n+1}) \mid x \in I_U\}$.

  Since $w \in PrET(S_1)$, there is some $u \in O^*$ with $wu \in StET(S_1)$. Hence, in $U\|S_1$, we have a run $(q_0, q_{01}) \stackrel{w}{\Rightarrow} (q_{n+1}, q'') \stackrel{u}{\Rightarrow} (q_{n+1}, q')$ with $q' \in E_1$. This implies $wu \in StET(U\|S_1)$. Since each action in wu is an output in one component, $U\|S_1$ can reach an error locally.

  By assumption, also in $U\|S_2$ an error can be reached locally. In the respective run, U and S_2 each perform some $x_1 \dots x_i u'$ with some $u' \in I_U^* = O^*$. With this, S_2 reaches a state in E_2, since U does not have any errors. Thus, $\mathrm{prune}(x_1 \dots x_i u') = \mathrm{prune}(x_1 \dots x_i) \in PrET_2 \subseteq ET_2$. This implies that $x_1 \dots x_i$ and also w are in ET_2.

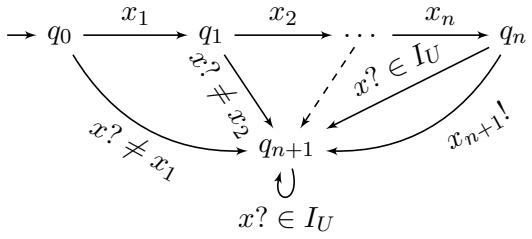

Figure 1: $x? \neq x_i$ represents all $x \in I_U \backslash \{x_i\}$

It remains to show $EL_1 \subseteq EL_2$. For this, it now suffices to consider some $w \in L_1$. Since clearly $\varepsilon \in EL_2$, we can assume $w = x_1 \dots x_n$ with $n \geq 1$. We construct the following partner U (see also Fig. 2):

- $Q_U = \{q_0, q_1, \dots, q_n, q\}$,
- $q_{0U} = q_0$,
- $E_U = \{q_n\}$,
- $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i < n\}$
  $\cup \{(q_i, x, q) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i < n\}$
  $\cup \{(q_n, x, q_n),\ (q, x, q) \mid x \in I_U\}$.

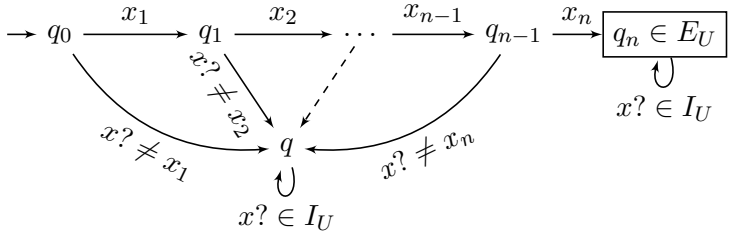

Figure 2: $x? \neq x_i$ represents all $x \in I_U \backslash \{x_i\}$, q_n is the only error state

Since $w \in L_1$, $U \| S_1$ and thus also $U \| S_2$ can reach an error locally. If this error stems from q_n, then $w \in L_2 \subseteq EL_2$. If this is not the case, S_2 has some strict error trace $x_1 \dots x_i u$ with $u \in I_U^* = O^*$. Then, some prefix of $x_1 \dots x_i$ is a pruned error trace, and again $w \in EL_2$. □

The next theorem states that $\sqsubseteq_E$ is the coarsest precongruence we have been looking for. In particular, this means that, if one is interested in a precongruence w.r.t. $\|$ that refines $\sqsubseteq_E^{\mathrm{B}}$, one needs a relation as fine as $\sqsubseteq_E$. What we prove is actually stronger: it suffices to be interested in a relation that is compositional w.r.t. $\|$ just for partners and that refines $\sqsubseteq_E^{\mathrm{B}}$ just on systems without inputs. On such systems, which result from the composition with a partner, local reachability coincides with reachability. That we do not want to introduce a reachable error in a refinement step if there was not one initially, is possibly even more convincing than $\sqsubseteq_E^{\mathrm{B}}$.

Theorem 13 (Full abstractness for Error Semantics)**.** *For two EIOs S_1 and S_2 with the same signature, we have $S_1 \sqsubseteq_E^{\mathrm{C}} S_2 \Leftrightarrow S_1 \sqsubseteq_E S_2$.*

Proof: „$\Leftarrow$“: If $S_1 \sqsubseteq_E S_2$ and S_1 can reach an error locally, we have $\varepsilon \in ET_1 \subseteq ET_2$. This implies that S_2 can also reach an error locally, thus $\sqsubseteq_E$ is contained in $\sqsubseteq_E^{\mathrm{B}}$.

As stated in Corollary 11, $\sqsubseteq_E$ is a precongruence w.r.t. $\|$. Since $\sqsubseteq_E^{\mathrm{C}}$ is the coarsest precongruence w.r.t. $\|$, $\sqsubseteq_E$ is contained in $\sqsubseteq_E^{\mathrm{C}}$.

„$\Rightarrow$“: Since $\sqsubseteq_E^{\mathrm{C}}$ is a precongruence, we have $U\|S_1 \sqsubseteq_E^{\mathrm{C}} U\|S_2$ for all partners U. Since $\sqsubseteq_E^{\mathrm{C}}$ is contained in $\sqsubseteq_E^{\mathrm{B}}$, this implies $U\|S_1 \sqsubseteq_E^{\mathrm{B}} U\|S_2$ for all partners U. With Lemma 12, we get $S_1 \sqsubseteq_E S_2$. □

3.2 Hiding and Error-Freedom

Since hiding turns outputs into τ, i.e. some local actions into another one, local reachability remains the same. This is essential for the following result to hold.

Theorem 14 (Error-Precongruence w.r.t. Internalization)**.** *Let S be an EIOs. Then:*

(i) $L(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in L(S) : w'|_{\Sigma\backslash X} = w\Big\}$,

(ii) $ET(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in ET(S) : w'|_{\Sigma\backslash X} = w\Big\}$,

(iii) $EL(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in EL(S) : w'|_{\Sigma\backslash X} = w\Big\}$.

Hence, $\sqsubseteq_E$ is a precongruence w.r.t. hiding as well as w.r.t. parallel composition with hiding.

Proof: Part (i) is obvious. It only remains to show Part (ii). Then, Part (iii) follows, which implies the first and then, with Cor. 11, the second precongruence statement.

We consider some $w \in ET(S/X)$. There are a prefix $v \in PrET(S/X)$ of w and some $u \in O^*_{S/X}$ with $vu \in StET(S/X)$. The respective underlying run of vu exists in S as well, except that some τ-transitions might be labelled by outputs from X in S. Hence, there are v' and u' such that $v'|_{\Sigma\setminus X} = v$, $u'|_{\Sigma\setminus X} = u$, $u' \in O^*$ and $v'u' \in StET(S)$, v' like v does not end with an output. (For the latter, u' has to start immediately after all actions of v have been performed.) Hence, $v' \in PrET(S)$, and the same word that extends v to w extends v' to a suitable $w' \in ET(S)$.

Vice versa, consider some $w' \in ET(S)$. There are a prefix $v' \in PrET(S)$ of w' and some $u' \in O^*$ with $v'u' \in StET(S)$. The respective underlying run of $v'u'$ exists in S/X as well, except that transition labels from X are replaced by τ. Since v' does not end with an output, the same holds for $v = v'|_{\Sigma\setminus X}$. Thus, $v \in PrET(S/X)$, and v is a prefix of $w'|_{\Sigma\setminus X}$; we are done. □

4 Preserving Freedom From Quiescence

4.1 Characterizing $\sqsubseteq^{\mathrm{C}}_{Qui}$

In this section, we will consider quiescence as an additional fault. The resulting fully abstract precongruence will also be a precongruence for hiding, even though we do not consider divergence in this approach. Recall the definition of quiescence, $\sqsubseteq^{\mathrm{B}}_{Qui}$ and $\sqsubseteq^{\mathrm{C}}_{Qui}$ in Section 2.1. To characterize $\sqsubseteq^{\mathrm{C}}_{Qui}$, we extend the error semantics by a third set of quiescent traces or qsc-traces for short. Behaviour after an error, including quiescence, does not matter; hence, qsc-traces are flooded with error traces. In contrast to an error, a quiescence in one component can be escaped by suitable behaviour of the other. Thus, qsc-traces are not closed under continuation. They are a subset of the language, so flooding EL with qsc-traces would not have an effect anyway.

Definition 15 (Quiescence Semantics)**.** *For an EIO S, we denote the set of quiescent states by Qui (or Qui_S) and define the following trace sets:*

- strict qsc-traces*:* $StQT(S) := \left\{ w \in \Sigma^* \mid q_0 \stackrel{w}{\Rightarrow} q \in Qui \right\}$,
- (error-flooded) qsc-traces*:* $QET(S) := StQT(S) \cup ET(S)$.

We call $(ET(S), QET(S), EL(S))$ the quiescence semantics *of S. For two EIOs S_1, S_2 with the same signature, we write $S_1 \sqsubseteq_{Qui} S_2$ if $S_1 \sqsubseteq_E S_2$ and $QET_1 \subseteq QET_2$.*

Due to input-enabledness, the following lemma is obvious from the definitions and Lemma 5.1.[4]

Lemma 16. *1. A state (q_1, q_2) of a parallel composition S_{12} is quiescent, if and only if the states q_1 and q_2 are so in S_1, S_2 resp.*

2. Let $w \in \Sigma_{12}^*$, $w_1 = w|_{\Sigma_1}$ and $w_2 = w|_{\Sigma_2}$. Then, $w \in StQT_{12}$ if and only if $w_1 \in StQT_1$ and $w_2 \in StQT_2$.

We list Part 1 and 3 in the following theorem to present the complete semantics; they have already been proven in Thm. 10.

Theorem 17 (Quiescence Semantics for Parallel Composition)**.** *For two composable EIOs S_1, S_2 and their composition S_{12}, we have:*

1. $ET_{12} = \mathrm{cont}\,(\mathrm{prune}\,((ET_1 \| EL_2) \cup (EL_1 \| ET_2)))$,
2. $QET_{12} = (QET_1 \| QET_2) \cup ET_{12}$,
3. $EL_{12} = (EL_1 \| EL_2) \cup ET_{12}$.

Proof: We only have to prove Part 2. For the inclusion, it suffices to consider some $w \in StQT_{12}$, and this case is settled by Lemma 16.2. For the reverse inclusion, it suffices to consider some $w_1 \in QET_1$ and $w_2 \in QET_2$. If one of these is an error trace, we are done by Part 1, otherwise again by Lemma 16.2. □

Again, we have the following consequence by monotonicity.

Corollary 18 (Quiescent Precongruence)**.** *The relation $\sqsubseteq_{Qui}$ is a precongruence w.r.t. $\|$.*

We now consider the communication with partners, taking also quiescence into account.

Lemma 19. *Let S_1 and S_2 be two EIOs with the same signature. If $U \| S_1 \sqsubseteq^{\mathrm{B}}_{Qui} U \| S_2$ for all partners U, then $S_1 \sqsubseteq_{Qui} S_2$.*

Proof: We will modify and extend the proof of Lemma 12. Here, we restrict ourselves to partners with $I_U = O$ and $O_U = I \cup \{\omega\}$ with a fresh action ω. This action allows the partner to prevent quiescence. In this but not the next section, we could replace ω by τ and still have $O_U = I$.

We first show $ET_1 \subseteq ET_2$ and consider a prefix-minimal $w \in ET_1$.

[4] In a setting without input-enabledness, the forward implication in Part 1 (of Lemma 16) only holds if (q_1, q_2) is not an error.

- $w = \varepsilon$: In this case, S_1 has a locally reachable error. Let U have just one non-error state with a loop for ω and all $x \in I_U$. Thus, S_1 can essentially reach the same states locally as $U \| S_1$, and $U \| S_2$ can reach an error or a quiescent state locally. The latter is impossible due to the ω-loop, and the error can only stem from S_2; it can be reached without involving ω, and thus $w \in ET_2$.

- $w = x_1 \dots x_n x_{n+1} \in \Sigma^+$ with $n \geq 0$ and $x_{n+1} \in I \subseteq O_U$: We construct the same partner U as for Lemma 12, but with an additional ω-transition from each state to q_{n+1}; see Fig. 3. Similarly, we derive that in $U \| S_2$ an error or a quiescent state can be reached locally; since the latter is impossible, we conclude as for Lemma 12. For u' in the resp. proof, note that it can have ωs in addition to outputs from O. But since U is input-enabled, S_2 can perform these outputs in the same way without intervening ωs; so we can assume $u' \in O^*$ and proceed as in the proof of Lemma 12.

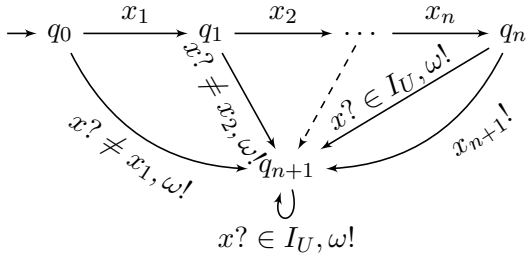

Figure 3: $x? \neq x_i$ represents all $x \in I_U \backslash \{x_i\}$

Next, we consider $EL_1 \subseteq EL_2$. As in Section 3.1, it suffices to consider some $w = x_1 \dots x_n \in L_1$ with $n \geq 1$. We modify the partner U from Lemma 12 in the same way as in the previous case; see Fig. 4. Since the ω-transitions make sure that there are no quiescent states in any of the two compositions, the proof now works as for Lemma 12.

For the remaining inclusion, we have to prove that any $w = x_1 \dots x_n \in StQT_1$ with $n \geq 0$ is also in QET_2. We construct the following partner U; see Fig. 5:

- $Q_U = \{q_0, q_1, \dots, q_n, q\}$,

- $q_{0U} = q_0$,

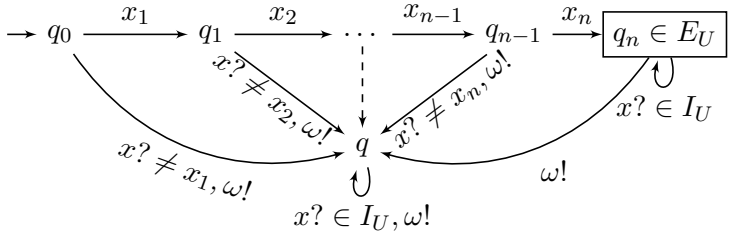

Figure 4: $x? \neq x_i$ represents all $x \in I_U \backslash \{x_i\}$, q_n is the only error state

- $E_U = \emptyset$,
- $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i < n\}$
  $\cup \{(q_i, x, q) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i < n\}$
  $\cup \{(q_i, \omega, q) \mid 0 \leq i < n\}$
  $\cup \{(q_n, x, q) \mid x \in I_U\}$
  $\cup \{(q, \alpha, q) \mid \alpha \in I_U \cup \{\omega\}\}$.

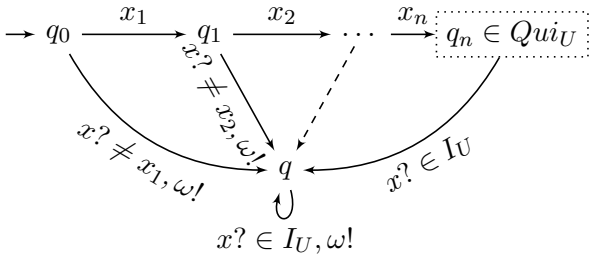

Figure 5: $x? \neq x_i$ represents all $x \in I_U \backslash \{x_i\}$, q_n is the only quiescent state

Clearly, w reaches a quiescent state in $U \| S_1$ and consists of outputs only. By assumption, also $U \| S_2$ can reach an error or quiescent state locally.

a) If an error is reached locally, it is inherited from S_2; to reach it, S_2 performs a prefix of w and possibly some more outputs. We are done as in the case for ET-inclusion.

b) If a quiescent state is reached locally, S_2 performs w and reaches a quiescent state itself. Hence, $w \in StQT_2 \subseteq QET_2$.

□

Now we can prove that $\sqsubseteq_{Qui}$ characterizes the coarsest precongruence contained in $\sqsubseteq^{\mathrm{B}}_{Qui}$.

Theorem 20 (Full Abstractness for Quiescence Semantics)**.** *For two EIOs S_1 and S_2 with the same signature, we have $S_1 \sqsubseteq^{\mathrm{C}}_{Qui} S_2 \Leftrightarrow S_1 \sqsubseteq_{Qui} S_2$.*

Proof: The proof is essentially the same as the one for Theorem 13, using Lemma 19 instead of Lemma 12. To conclude that $\sqsubseteq_{Qui}$ is contained in $\sqsubseteq^{\mathrm{B}}_{Qui}$, one only has to add: If $S_1 \sqsubseteq_{Qui} S_2$ and S_1 can reach a quiescent state locally with some w, we have $w \in QET_1 \subseteq QET_2$. Then either S_2 can reach an error locally, or $w \in QET_2 \setminus ET_2 \subseteq StQT_2$ and S_2 can also reach a quiescent state locally with w. □

4.2 Hiding, Conjunction and Quiescence

Since only outputs are hidden, hiding does not change the quiescence status of a state. Therefore, it is easy to see the following result in the light of Theorem 14:

Theorem 21 (Quiescence Precongruence w.r.t. Internalization)**.** *Let S be an EIO. Then:*

(i) $ET(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in ET(S) : w'|_{\Sigma\backslash X} = w\Big\}$,

(ii) $EL(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in EL(S) : w'|_{\Sigma\backslash X} = w\Big\}$.

(iii) $StQT(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in StQT(S) : w'|_{\Sigma\backslash X} = w\Big\}$,

(iv) $QET(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in QET(S) : w'|_{\Sigma\backslash X} = w\Big\}$,

Hence, $\sqsubseteq_{Qui}$ is a precongruence w.r.t. hiding as well as w.r.t. parallel composition with hiding.

To show that the new $\sqsubseteq_{Qui}$ is a feasible basis for an interface theory, we will now define an operator $\wedge$ that satisfies the defining requirement for a conjunction, i.e. the refinements of $S_1 \wedge S_2$ (if it exists) are the common refinements of S_1 and S_2. For this to hold, S_1 and S_2 must have the same signature (I, O). Observe that conjunction is determined by the refinement – up to the equivalence contained in the refinement.

Optimally, $ET(S_1 \wedge S_2)$ will be $ET_1 \cap ET_2$, because only those traces are allowed to reach errors locally in both automata. Analogously intersection is optimal for EL and QET. Naturally, $S_1 \wedge S_2$ should be some Cartesian product with error set $E_1 \times E_2$, such that $StET_{S_1 \wedge S_2} = StET_1 \cap StET_2$.

But in view of pruning and continuation for ET, this does not work directly, and we will first normalize the EIOs.

To see why, let $i \in I$ and $o, o' \in O$, and assume that $io \in StET_1 \setminus StET_2$ and $io' \in StET_2 \setminus StET_1$. Neither io nor io' will reach an error in the Cartesian product, but i is a common error trace. To make sure that it is an error trace in the product, we will prune S_1 and S_2 according to Step i) below. This step is in essence very similar to the pruning applied during parallel composition in [6, 7]. Also, we could have $i \in StET_1 \setminus StET_2$ and $ii \in StET_2 \setminus StET_1$. In this case, neither i nor ii will reach an error in the Cartesian product, but ii is a common error trace. Consequently, we will cater for continuation in Step ii) below.

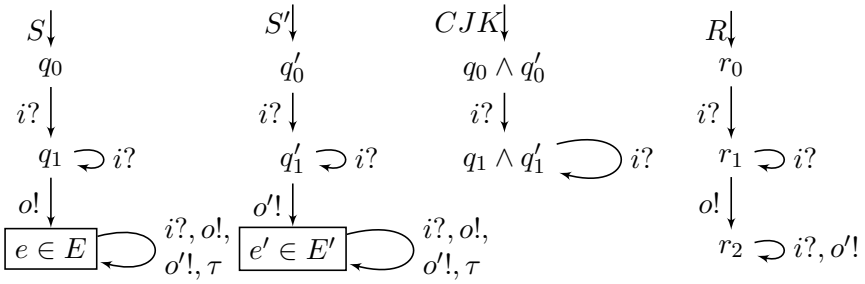

Figure 6: CJK: conjunction according to [5], R: some common refinement

In [5], a construction is used for conjunction w.r.t. $\sqsubseteq_E$ that *is* the Cartesian product with error set $E_1 \times E_2$. Since $e \xrightarrow{\alpha} e$ is required for each $e \in E$ and $\alpha \in \Sigma \cup \{\tau\}$, the second problem does not arise. But due to the first problem, the construction is wrong there e.g. for S and S' in Fig. 6, which have the common input i, outputs o, o' and error trace i. The conjunction according to [5] shown does not cover the common refinement R of S and S'. Our construction would normalize S and S' to the same automaton. Hence, also our conjunction would be the same automaton and look like the automaton CJK except that the state $q_1 \wedge q'_1$ would be an error state with a loop for all actions in Σ.

Definition 22 (Normal Form)**.** *An EIO S is in* normal form *(NF), if E has just one element, $E = \{e\}$ say, and the following hold: $e \xrightarrow{a} e$ for all $a \in \Sigma$, $q \xrightarrow{\alpha} e \wedge q \neq e$ implies $\alpha \in I$, and $e \xrightarrow{\alpha} q$ implies $q = e$.*

The normal form *of S is the EIO $NF(S)$ obtained from S with the following two steps:*

i) Define $\overline{E} := \{q \mid \exists q' \in E,\ w \in O^* :\ q \stackrel{w}{\Longrightarrow} q'\}$ *and replace E by $\overline{E}$.*

ii) If $q_0 \in \overline{E}$, *let* $NF(S) := \Big(\{q_0\}, I, O, \{q_0 \stackrel{a}{\to} q_0 \mid a \in \Sigma\}, q_0, \{q_0\}\Big)$.

Otherwise, add a new state e, which becomes the only error state. Whenever $q \stackrel{\alpha}{\to} q'$ with $q' \in \overline{E}$ and $q \notin \overline{E}$ for some $\alpha \in \Sigma \cup \{\tau\}$ (then necessarily $\alpha \in I$), remove all α-transitions from q and add a transition $q \stackrel{\alpha}{\to} e$. Finally, add all $e \stackrel{a}{\to} e$ with $a \in \Sigma$.

Clearly, $NF(S)$ is in normal form. Step i) does not change the runs, nor whether a state is quiescent or not. It does change $StET$, but neither $PrET$ nor ET. Thus, the result is equivalent to S w.r.t. $\sqsubseteq_{Qui}$ and hence $\sqsubseteq_E$. In the first case of Step ii), we have $ET(S) = \Sigma^* = ET(NF(S))$. In the second case, the step does not change whether a state is quiescent or not. We may gain runs that use some $e \stackrel{a}{\to} e$, but their traces are in $ET(S)$ and $ET(NF(S))$. We may also lose runs that use some $q \stackrel{\alpha}{\to} q'$ in S that is removed in Step ii); but their traces are again in $ET(S)$ and $ET(NF(S))$. Since $ET(S) = ET(NF(S))$ and all relevant trace sets are flooded with ET, $NF(S)$ is equivalent to S w.r.t. $\sqsubseteq_{Qui}$ and hence $\sqsubseteq_E$. So we have:

Proposition 23 (Normal Form)**.** *Each EIO S is equivalent to $NF(S)$ w.r.t. $\sqsubseteq_{Qui}$ and $\sqsubseteq_E$.*

Thus, we can assume that an EIO is in normal form if we wish. The above transformation is efficient for finite EIOs: For Step i), one performs in linear time one breadth first search (BFS) from the states in E via the reversed local transitions. For the non-trivial case of Step ii), one performs one BFS from q_0 without traversing transitions that have to be deleted. All states not visited are unreachable in $NF(S)$ and can be removed, so that $NF(S)$ might be much smaller than S. We have the following easy lemma.

Lemma 24. *If an EIO S is in normal form (with error state e), $w \in ET(S)$ if and only if w arises from a run reaching e, and $w \in EL(S)$ if and only if w arises from a run. Thus, $EL(S) = L(S)$.*

Now we are ready to define $\wedge$.

Definition 25 (Conjunction)**.** *Let S_1 and S_2 be EIOs with the same signature (I, O), which we assume to be in normal form (with error states e_1, e_2). We define $S := S_1 \times S_2$ as follows: $Q = Q_1 \times Q_2$, the signature is (I, O), $q_0 = (q_{01}, q_{02})$, $E = E_1 \times E_2 = \{(e_1, e_2)\}$, and δ consists of all*

$((q_1, q_2), a, (p_1, p_2))$ *for* $(q_1, a, p_1) \in \delta_1, (q_2, a, p_2) \in \delta_2$ *and* $a \in \Sigma$,

$((q_1, q_2), \tau, (p_1, q_2))$ *for* $(q_1, \tau, p_1) \in \delta_1$,

$((q_1, q_2), \tau, (q_1, p_2))$ *for* $(q_2, \tau, p_2) \in \delta_2$.

$S_1 \wedge S_2$ is obtained from S by adding a τ-loop to every $(q_1, q_2) \in Qui$ if one of the q_i is neither quiescent nor an error.

The addition of the τ-loops makes sure that states in the product are not wrongly quiescent. We close this section by showing that $\wedge$ is really a conjunction in the settings of this and the previous section. This implies also that $\wedge$ is commutative, associative and compositional, cf. e.g. [3].

Theorem 26 (Conjunction)**.** *For three EIOs R, S_1 and S_2 with the same signature, we have that $R \sqsubseteq_{Qui} S_1 \wedge S_2$ if and only if $R \sqsubseteq_{Qui} S_1$ and $R \sqsubseteq_{Qui} S_2$; the same holds for $\sqsubseteq_E$.*

Proof: Since $\wedge$ involves normalization, we can assume that S_1 and S_2 are in normal form. Furthermore, the Cartesian product is known from automata theory as a construction for the intersection of languages of automata with final states. Hence, we can derive from Lemma 24 that $ET(S_1 \wedge S_2) = ET_1 \cap ET_2$ and $EL(S_1 \wedge S_2) = EL_1 \cap EL_2$; we are done once we have shown the analogous statement for QET.

First, consider some $w \in QET_1 \cap QET_2$. Then, for $i \in \{1, 2\}$, w can be performed to reach some $q_i \in Q_i$ that is quiescent or the error. Thus, no τ-loop is added to (q_1, q_2) in $S_1 \wedge S_2$, and (q_1, q_2) is quiescent due to a quiescent component or equal to (e_1, e_2). Hence, $w \in QET(S_1 \wedge S_2)$.

Second, consider some $w \in QET(S_1 \wedge S_2)$. If $w \in ET(S_1 \wedge S_2)$, then $w \in QET_1$ and $w \in QET_2$. Otherwise, $w \in StQT(S_1 \wedge S_2)$ due to some $(q_1, q_2) \in Qui$. Since (q_1, q_2) has no τ-loop, we conclude that each q_i is quiescent or the error, implying $w \in QET_1 \cap QET_2$. □

5 Preserving Freedom From Divergence

5.1 Characterizing $\sqsubseteq^{\mathrm{C}}_{Div}$

In this section, we will additionally treat divergent states as faulty. Recall the definition of divergence, Div, $\sqsubseteq^{\mathrm{B}}_{Div}$ and $\sqsubseteq^{\mathrm{C}}_{Div}$ in Section 2.1. To characterize $\sqsubseteq^{\mathrm{C}}_{Div}$, we modify the quiescence semantics further to cater for divergence. Since divergence cannot be prevented by another component in a parallel

composition, it is as catastrophic as an error. Consequently, we will define divergence traces, or div-traces for short, with pruning and continuation. We will add these div-traces to ET and use the extended set for flooding the other two trace sets of the semantics.

Definition 27 (Divergence Traces)**.** *For an EIO S, we define the following trace sets:*

- strict div-traces*:* $StDT(S) := \left\{ w \in \Sigma^* \mid q_0 \stackrel{w}{\Rightarrow} q \in Div \right\}$,
- pruned div-traces*:* $PrDT(S) := \{\text{prune}(w) \mid w \in StDT(S)\}$,
- div-traces $DT(S) := \text{cont}(PrDT(S))$.

Definition 28 (Divergence Semantics)**.** *The* divergence semantics *of an EIO S consists of:*

- *the set of* error-div-traces *of S:* $EDT(S) := ET(S) \cup DT(S)$
- *the set of* flooded qsc-traces *of S:* $QDT(S) := StQT(S) \cup EDT(S)$
- *the* flooded language *of S:* $EDL(S) := L(S) \cup EDT(S)$.

We call $(EDT(S), QDT(S), EDL(S))$ the divergence semantics of S. For two EIOs S_1, S_2 with the same signature, we write $S_1 \sqsubseteq_{Div} S_2$ if $EDT_1 \subseteq EDT_2$, $QDT_1 \subseteq QDT_2$ and $EDL_1 \subseteq EDL_2$.

Although the new refinement is closely related to the previous ones, the sets in the semantics above are all different from previous sets. In fact, $\sqsubseteq_{Div}$ is incomparable to the two earlier precongruences. For comparison, we also introduce the quiescence- and divergence-sensitive precongruence $\sqsubseteq_{CJK}$ from [5]: it is the component-wise inclusion of the trace sets ET, $ET \cup StDT$, $ET \cup StDT \cup StQT$ and EL. From the definitions, it is obvious that $\sqsubseteq_{Qui}$ and $\sqsubseteq_{CJK}$ are contained in $\sqsubseteq_E$. Furthermore, $\sqsubseteq_{CJK}$ is contained in $\sqsubseteq_{Div}$: Closing the second trace set of the former semantics under pruning and continuation gives EDT; flooding the third and the fourth set with EDT gives QDT and EDL; thus, inclusion carries over. Using the EIOs in Fig. 7, we will now show that these inclusions are strict and no other inclusions hold; these relations between the four precongruences are depicted in Fig. 8 below.

S_1 and S_2 are equivalent w.r.t. $\sqsubseteq_{Qui}$ (hence $\sqsubseteq_E$), but $\neg S_1 \sqsubseteq_{Div} S_2$ due to the immediate divergence, hence also $\neg S_1 \sqsubseteq_{CJK} S_2$. Furthermore, S_4

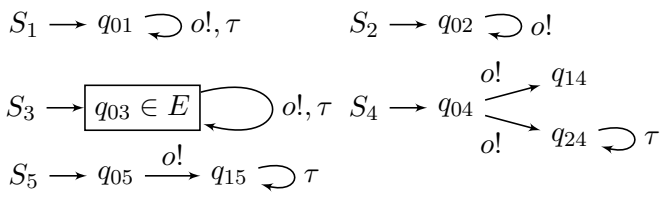

Figure 7: Examples disproving some inclusions

and S_5 are equivalent w.r.t. $\sqsubseteq_{CJK}$ (hence $\sqsubseteq_E$), but $\neg S_4 \sqsubseteq_{Qui} S_5$ due to the quiescence after o; observe that, for $\sqsubseteq_{CJK}$, this quiescence is covered up by the strict divergence trace o. This settles in which precongruences $\sqsubseteq_{Qui}$, $\sqsubseteq_E$ and $\sqsubseteq_{CJK}$ are contained.

Finally, S_1 and S_3 are equivalent w.r.t. $\sqsubseteq_{Div}$ since errors and divergences are considered equally bad. But S_3 is not a refinement of S_1 in any of the other three settings, which respect ET. We conclude:

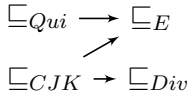

Figure 8: Inclusions between precongruences

Theorem 29 (Comparison of Precongruences)**.** *All inclusions between $\sqsubseteq_{Qui}$, $\sqsubseteq_E$, $\sqsubseteq_{Div}$ and $\sqsubseteq_{CJK}$ are depicted in Fig. 8 as arrows; these inclusions are strict.*

To shed more light on the difference between $\sqsubseteq_{Div}$ and $\sqsubseteq_{CJK}$, consider the EIOs in Fig. 9; a represents i or o. Observe that, no matter how a is chosen, the EIOs have no error or quiescent states. We have $R_i \sqsubseteq_{Div} R_i'$ due to the immediate divergence in R_i', but $\sqsubseteq_{CJK}$ fails since cont is not applied to $StDT$. On the other hand, $R_o' \sqsubseteq_{Div} R_o$ since we apply prune to $StDT$, whereas $\sqsubseteq_{CJK}$ fails.

Presumably, the authors of [5] consider it unconvincing that quiescence can be resolved by invisible actions alone; at the same time, they regard it as good enough if an output action is performed after some τs. But this does not mean that e.g. the divergence in S_1 above is problematic since an output is possible all the time.

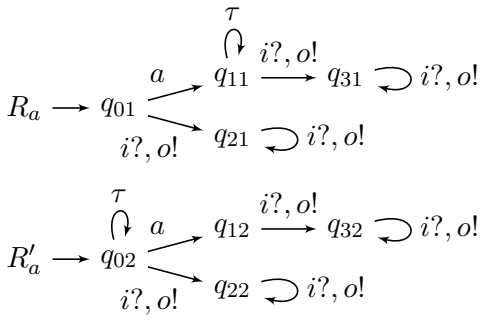

Figure 9: Difference between $\sqsubseteq_{Div}$ and $\sqsubseteq_{CJK}$

And if divergence really is a problem in any case, then it cannot be 'left behind' due to activity of the environment as in the case of a quiescence. Consequently, it should be treated like an error as in our setting. We believe the idea we ascribed to the authors of [5] would be adequately formalized by calling state q not quiescent if $q \stackrel{o}{\Rightarrow}$ for some $o \in O$. We suspect that developing the setting of the previous section with this notion of fault could be difficult.

To proceed, we show compositionality for $\sqsubseteq_{Div}$.

Theorem 30 (Divergence Semantics for Parallel Composition)**.** *For two composable EIOs S_1, S_2 and their composition S_{12}, we have:*

1. $EDT_{12} = \mathrm{cont}\,(\mathrm{prune}\,((EDT_1 \| EDL_2) \cup (EDL_1 \| EDT_2)))$,
2. $QDT_{12} = (QDT_1 \| QDT_2) \cup EDT_{12}$,
3. $EDL_{12} = (EDL_1 \| EDL_2) \cup EDT_{12}$.

Proof: The proof is the same as the combination of the proofs for Theorem 17 and 10; throughout, one has to treat divergent states and error-div-traces in the same way as error states and traces above. □

Corollary 31 (Divergence Precongruence)**.** *The relation $\sqsubseteq_{Div}$ is a precongruence w.r.t. $\|$.*

We now continue to characterize $\sqsubseteq^{\mathrm{C}}_{Div}$ in the same way as in the previous sections.

Lemma 32. *Let S_1 and S_2 be two EIOs with the same signature. If $U \| S_1 \sqsubseteq^{\mathrm{B}}_{Div} U \| S_2$ for all partners U, then $S_1 \sqsubseteq_{Div} S_2$.*

Proof: For Lemma 19, we modified and extended the proof of Lemma 12. Here, we point out that the modified version works for the present lemma as well. The only change is that, instead of an error state, an error or divergent state has to be considered – and similarly, EDT instead of ET etc. and $StET \cup StDT$ instead of $StET$ etc. □

Now we can conclude that $\sqsubseteq_{Div}$ characterizes the coarsest precongruence contained in $\sqsubseteq^{\mathrm{B}}_{Div}$. The proof is again analogous to those of Theorem 13 and 20; in particular, we have to replace error state by error or divergent state.

Theorem 33 (Full Abstractness for Divergence Semantics)**.** *For two EIOs S_1 and S_2 with the same signature, we have $S_1 \sqsubseteq^{\mathrm{C}}_{Div} S_2 \Leftrightarrow S_1 \sqsubseteq_{Div} S_2$.*

5.2 Hiding, Conjunction and Divergence

In a divergence-sensitive setting, precongruence for hiding usually needs some finiteness condition. In our study of hiding, we restrict ourselves to finite EIOs and to the hiding of single outputs to keep concepts simple. Hiding of finite sets can be obtained by repeating such hiding. We write S/o for $S/\{o\}$ if $o \in O$. Note that, in the following result, $EDT(S/o)$ is obtained from $EDL(S)$; it is larger than just $\{w \mid \exists w' \in EDT(S) : w'|_{\Sigma\setminus\{o\}} = w\}$. Due to the latter, the other two sets need a new flooding.

Theorem 34 (Divergence Precongruence w.r.t. Internalization)**.** *Let S be a finite EIO and $o \in O$. Then:*

(i) $EDT(S/o) =$
$\mathrm{cont}\left(\mathrm{prune}\left(\left\{w \mid \exists w' : w'|_{\Sigma\setminus\{o\}} = w \wedge \forall n \geq 0 : w'o^n \in EDL(S)\right\}\right)\right)$,

(ii) $EDL(S/o) = \left\{w \mid \exists w' \in EDL(S) : w'|_{\Sigma\setminus\{o\}} = w\right\} \cup EDT(S/o)$,

(iii) $QDT(S/o) = \left\{w \mid \exists w' \in QDT(S) : w'|_{\Sigma\setminus\{o\}} = w\right\} \cup EDT(S/o)$

Hence, $\sqsubseteq_{Div}$ is a precongruence w.r.t. hiding as well as w.r.t. parallel composition with hiding on finite EIOs.

Proof: Part (ii) and (iii) should be clear from earlier considerations. It only remains to show Part (i). Then, the precongruence statements follow.

„$\subseteq$“: Since the r.h.s is closed under pruning and continuation, it suffices to consider $w \in StET(S/o) \cup StDT(S/o)$. Such a w arises from some $q_0 \stackrel{w}{\Longrightarrow} q$ and the same run in S gives some w' with $w'|_{\Sigma\setminus\{o\}} = w$.

If $w \in StET(S/o)$, the run results in $w' \in StET(S)$ so that the second condition follows from continuation closure. If $w \in StDT(S/o)$, then q is divergent in S, implying the second condition in the same way, or q enables an infinite trace o^ω. In the latter case, the second condition is immediate.

„$\supseteq$“: Consider some w' as specified. If some $w'o^n \in EDT(S)$, closure under pruning implies that some prefix v' of w' is in $PrET(S) \cup PrDT(S)$. Then $v'|_{\Sigma\setminus\{o\}}$ is in $ET(S/o)$ by Theorem 14 ii) or similarly in $DT(S/o)$. This implies $w'|_{\Sigma\setminus\{o\}} \in EDT(S/o)$ by closure under continuation.

Otherwise, we have $w'o^n \in L(S)$ for all n. For $n = |Q|$, some state occurs twice along o^n on a run underlying $w'o^n$. This state is divergent in S/o, and the run to this state shows $w'|_{\Sigma\setminus\{o\}} \in EDT(S/o)$. □

Conjunction is more difficult for the setting in this section. In particular, adding a τ-loop as in Def. 25 would add a fault, which is not allowed. If a trace is neither quiescent nor divergent and state q is reached by this trace, we can follow a (possibly empty) path of τ-transitions from q until we reach a stable state; this must enable an output. In other words, q is not quiescent in the sense that $q \stackrel{o}{\Rightarrow}$ for some $o \in O$, as discussed before Thm. 30. We note formally:

Lemma 35. *For each EIO and $w \in EDL \setminus QDT$, there is some $o \in O$ with $wo \in EDL$.*

If we consider two EIOs and the component-wise intersection of their divergence semantics (with sets EDL, QDT and EDT) we might get a trace $w \in EDL \setminus QDT$ such that $wo \in EDL$ holds for no output o, see the example after Def. 38. Such a trace demonstrates a local inconsistency, which we have to remove in the conjunction. Since this problem arises on the level of traces, we have to collect the states reached by the same trace in a standard powerset construction.

But first of all, we show how to transform an EIO into a divergence-free normal from.

Definition 36 (Divergence Normal Form)**.** The divergence normal form *of S is the EIO $DF(S)$ obtained from S as follows. First, we construct S' by adding all divergent states to E. Second, we set $DF(S) = NF(S')$ (c.f. Def. 22).*

S and S' have the same runs and quiescent states. Hence, they are equivalent w.r.t. $\sqsubseteq_{Div}$ since error and divergent states are treated the same way. We have $EDT(S') = ET(S')$ and can conclude that the divergence

and the quiescence semantics of S' coincide, i.e. we also have $EDL(S') = EL(S')$ and $QDT(S') = QET(S')$. Furthermore, the construction of $NF(S')$ removes all divergent states and does not add any new one. Thus, $DF(S)$ is divergence-free and equivalent to S w.r.t. $\sqsubseteq_{Div}$ by Prop. 23. The additional claims in the following proposition are obvious.

Proposition 37. *Each EIO S is equivalent to $DF(S)$ w.r.t. $\sqsubseteq_{Div}$. The latter is divergence-free and in normal form.*

If S is divergence-free, we have $EDT(S) = ET(S)$, $QDT(S) = QET(S)$ and $EDL(S) = EL(S)$. For such EIOs, $\sqsubseteq_{Div}$ and $\sqsubseteq_{Qui}$ coincide.

Thus, we can assume that an EIO is in normal form and divergence-free if we wish. The above transformation is efficient for finite EIOs since we can determine S' efficiently:
We consider only the τ-transitions of S and determine with a suitable variant of depth first search (DFS) the strongly connected components (SCCs). Call an SCC trivial if it consists of a single state without a loop. All states in non-trivial SCCs are divergent. The states that can be reached from these 'core-divergent' states via reversed τ-transitions form the set of divergent states; they can be determined with one DFS.

Now we will define the conjunction operator, which we denote by $\wedge$ again.

Definition 38 (Conjunction)**.** *Let S_1 and S_2 be EIOs with the same signature (I, O), which we assume to be divergence-free and in normal form (with error states e_1, e_2). Consider $S := S_1 \times S_2$ as defined in Def. 25.*

We construct the power-EIO $\mathfrak{P}(S)$ *of S as follows: the state set is the powerset $\mathfrak{P}(Q)$ denoted by $Q_{\mathfrak{P}}$, the signature is (I, O), $q_{0\mathfrak{P}} = \{(q_1, q_2) \mid (q_{01}, q_{02}) \stackrel{\varepsilon}{\Rightarrow} (q_1, q_2)\}$, $E_{\mathfrak{P}} = \{P \mid (e_1, e_2) \in P\}$, and $\delta_{\mathfrak{P}}$ consists of all transitions $P \stackrel{a}{\to} \{(q_1', q_2') \mid \exists (q_1, q_2) \in P : (q_1, q_2) \stackrel{a}{\to}\stackrel{\varepsilon}{\Rightarrow} (q_1', q_2')\}$ with $a \in \Sigma$. We restrict $\mathfrak{P}(S)$ to the reachable part and, whenever $q_{0\mathfrak{P}} \stackrel{w}{\Longrightarrow} P$, we denote P also with P_w.*

We say that $P \in Q_{\mathfrak{P}}$ allows quiescence *(modulo errors) if $\exists (q_1, q_2) \in P$: q_i is quiescent or $q_i = e_i$ for each $i = 1, 2$.*

Next, we define the inconsistency set $\mathfrak{F} \subseteq Q_{\mathfrak{P}}$ *as the least set that contains P whenever*

i) $\exists i \in I$ such that $P \stackrel{i}{\to} P'$ and $P' \in \mathfrak{F}$, or

ii) P does not allow quiescence, and $P' \in \mathfrak{F}$ whenever $P \stackrel{o}{\to} P'$ with $o \in O$.

$S_1 \wedge S_2$ is obtained from $\mathfrak{P}(S)$ by removing all states in $\mathfrak{F}$ (including the incident transitions) and performing the following quiescence correction*: If $P \notin E_{\mathfrak{P}} \cup \mathfrak{F}$ is not quiescent but allows for quiescence, we add a fresh state $\overline{P}$ that inherits all incoming transitions and all outgoing input transitions from P.*

Note that condition ii) of the inconsistency set construction in Def. 38 covers the base case of those P that have no output transition.

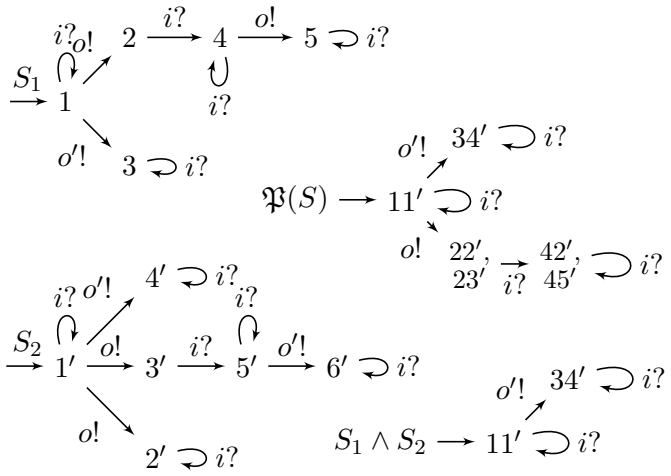

Figure 10: First example for Def. 38

Before proving the characteristic property of conjunction, we look at two examples, which are without error states and τ-transitions. Fig. 10 shows EIOs S_1 and S_2, as well as their power-EIO and conjunction. The trace $w = oi$ is quiescent in $\mathfrak{P}(S)$ since S_1 forbids o' and S_2 forbids o after w; but it is not quiescent in S_1. Thus, a common refinement of S_1 and S_2 cannot have trace w. This is the situation we referred to after Lemma 35.

Correspondingly, state $\{42', 45'\}$ does not allow quiescence due to state 4, and the second part of 38 ii) holds vacuously since $\{42', 45'\}$ has no output transition. Thus, this state is inconsistent, implying with i) that $\{22', 23'\}$ is inconsistent as well. Observe that an EIO with trace o would also have trace oi. We conclude that $\mathfrak{F} = \{\{42', 45'\}, \{22', 23'\}\}$; in particular, $\{11'\}$ has an output transition to $\{34'\} \notin \mathfrak{F}$.

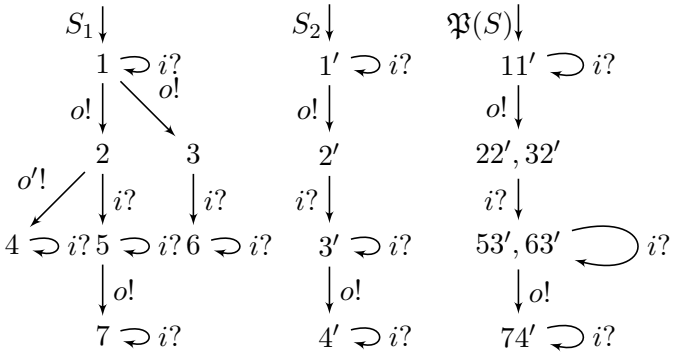

Figure 11: Second example for Def. 38

The next example in Fig. 11 shows the effect of using a powerset as state set. Here, $\mathfrak{F} = \emptyset$ since 38 ii) never applies: The first and the third state have an output; $\{22', 32'\}$ and $\{74'\}$ allow quiescence due to $32'$ and $74'$ resp.

A critical point is that $22'$ might look contradictory. It belongs to a quiescent state in $\mathfrak{P}(S)$, but 2 is not quiescent. One could have the idea to remove $22'$ from the quiescent state during the construction. But this would remove $53'$ from the next state, and the resulting $\{63'\}$ would have no output (and satisfy 38 ii)). But *oio* is possible in a common refinement, namely in $\mathfrak{P}(S) = S_1 \wedge S_2$, which is isomorphic to S_2. Note that S_1 has the quiescent traces *o* and *oio*.

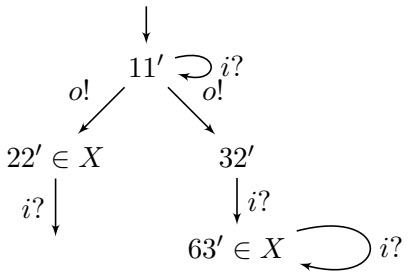

Figure 12: $\sqsubseteq_{CJK}$-conjunction according to [5]

Since S_1 and S_2 are without error states and τ-transitions, $\sqsubseteq_{Div}$ and $\sqsubseteq_{CJK}$ coincide, and hence the conjunctions should as well. It may be instructive to see how the construction in [5] fails in this case: it builds the Cartesian product and then – without any powerset construction – removes inconsistent states in the same way as we do (at least in this example).

Fig. 12 shows four initial states and all their outgoing transitions. States $22'$ and $63'$ do not allow quiescence, hence become elements of X as $\mathfrak{F}$ is denoted there. Then, $32'$ satisfies Clause i) and subsequently $11'$ satisfies Clause ii). According to this, there is no common refinement.

It is well known that $\mathfrak{P}(S)$ is deterministic and, for each $w \in \Sigma^*$ and $P \in Q_{\mathfrak{P}}$, we have $q_{0\mathfrak{P}} \stackrel{w}{\Longrightarrow} P$ if and only if $P = \{(q_1, q_2) \mid (q_{01}, q_{02}) \stackrel{w}{\Longrightarrow} (q_1, q_2)\}$. Hence, for S as well as $\mathfrak{P}(S)$, we know that $EDT = EDT_1 \cap EDT_2$ and $EDL = L = EDL_1 \cap EDL_2$; cf. Lemma 24 and the remark in the proof of Thm. 26. Hence, a common $\sqsubseteq_{Div}$-refinement R of S_1 and S_2 is at least an $\sqsubseteq_E$-refinement of $\mathfrak{P}(S)$.

Next we argue that, by removing $\mathfrak{F}$, we do not remove a trace that could appear in $EDL(R)$ for such an R, i.e. it cannot appear in any part of its divergence semantics. Note that an error state is never in $\mathfrak{F}$: if $(e_1, e_2) \in P$ and $P \stackrel{i}{\to} P'$, then $(e_1, e_2) \in P'$; thus, the first error state added to $\mathfrak{F}$ could not be added according to i). Furthermore, each error state allows quiescence, so it is surely not added to $\mathfrak{F}$ according to ii).

Lemma 39. *Let $R \sqsubseteq_{Div} S_1$ and $R \sqsubseteq_{Div} S_2$ for S_1 and S_2 as in the above definition. Let $w \in \Sigma^*$ and $P_w \in Q_{\mathfrak{P}}$.*

i) P_w allows quiescence if and only if $w \in QDT_1 \cap QDT_2$.

ii) $P_w \in \mathfrak{F}$ implies that $w \notin EDL(R)$.

iii) $\mathfrak{P}(S)$ and $S_1 \wedge S_2$ are divergence-free EIOs.

Proof: i) Obvious from the above considerations and the definition of allowing quiescence. Recall that the systems under consideration are divergence-free and S_1 and S_2 are in normal form.

ii) The proof is by (possibly transfinite for infinite O) induction on the derivation. If $P_w \in \mathfrak{F}$ due to Rule i), then $P_{wi} \in \mathfrak{F}$ and, by induction, $wi \notin EDL(R)$. Due to input-enabledness, this implies $w \notin EDL(R)$.

If $P_w \in \mathfrak{F}$ due to Rule ii), then it does not allow quiescence, and we cannot have $w \in QDT(R) \subseteq QDT_1 \cap QDT_2$ by Item i). Furthermore, for

each output o, either $P_w \overset{o}{\not\to}$ or $P_w \overset{o}{\to} P'_{wo} \in \mathfrak{F}$. In either case, $wo \notin EDL(R)$. Thus, $w \in EDL(R)$ would contradict Lemma 35.

iii) The first is obvious. $S_1 \wedge S_2$ is input-enabled due to Rule i) and since the quiescence correction preserves input-enabledness. □

Theorem 40 (Conjunction)**.** *For three EIOs R, S_1 and S_2 with the same signature, we have that $R \sqsubseteq_{Div} S_1 \wedge S_2$ if and only if $R \sqsubseteq_{Div} S_1$ and $R \sqsubseteq_{Div} S_2$.*

Proof: We can assume that R, S_1 and S_2 are divergence-free and in normal form. By Lemma 39, it suffices to check the claim for $\sqsubseteq_{Qui}$ in place of $\sqsubseteq_{Div}$.

„⇐“: For the system $\mathfrak{P}(S)$ above, we have $ET = ET_1 \cap ET_2$ and $EL = L = EL_1 \cap EL_2$, hence $R \sqsubseteq_E \mathfrak{P}(S)$. Due to Lemma 39 ii) and since quiescence correction does not change the language or the (strict) error traces, we also have $R \sqsubseteq_E S_1 \wedge S_2$.

If w is a qsc- but not an error trace of R, P_w exists in $S_1 \wedge S_2$ by Lemma 39 ii) since $w \in L(R)$. By assumption, $w \in QET_1 \cap QET_2$, and P_w allows quiescence by Lemma 39 i). Hence, w is a strict qsc-trace in $S_1 \wedge S_2$ due to P_w or $\overline{P_w}$.

„⇒“: It suffices to show that $S_1 \wedge S_2 \sqsubseteq_{Qui} S_1$ and $S_1 \wedge S_2 \sqsubseteq_{Qui} S_2$, where the latter follows from the former by symmetry.

By the arguments for „⇐“, we have $S_1 \wedge S_2 \sqsubseteq_E S_1$. If w is a qsc- but not an error trace of $S_1 \wedge S_2$, this is either due to $\overline{P_w}$ and $w \in QET_1 \cap QET_2$ by Lemma 39 i), or P_w is quiescent after the removal of $\mathfrak{F}$. Then, because of Rule ii), we conclude that P_w allows quiescence and we are done as in the first subcase. □

6 Quotient

Here, we extend the divergence-sensitive approach of the previous section by a quotient operation. This is a kind of inverse or adjoined operation to parallel composition. With this operation, we can reuse components and do an incremental component-based specification. Given two EIOs S and D, the quotient is the coarsest EIO P such that $P \| D \sqsubseteq_{Div} S$ holds, and we denote it by $S /\!\!/ D$ if the quotient exists. In the following, we call S the specification, D the *divisor* and P a *solution of the quotient inequality*. One should think of D as an already implemented component, and P is a completion of D such that $P \| D$ meets the specification S.

This section has profited from studying the quotients in [3] and [5]. The setting in [3] is based on modal transition systems with an alternating simulation relation as refinement. The quotient in [5] is based on slightly different trace set inclusions for refinement as discussed above, and only deterministic EIOs are considered.

In fact, quotient operators usually need some determinism assumption. In contrast, we treat arbitrary EIOs, but we bring S and D in some suitable normal form. This normal form uses a variation of EIOs with an additional function that assigns a bit to each state. Bit 1 indicates that the state can be regarded as quiescent even if, possibly, it is not.

Definition 41 (Bit-EIO)**.** *A* bit-EIO *is a divergence-free EIO* $S = (Q, I, O, \delta, q_0, E, b)$ *with an additional function* $b : Q \to \mathbb{B}$*. Satisfying for every* $q \in Q$*: if* q *is quiescent and* $q \notin E$ *then* $b(q) = 1$*; if* $q \in E$ *then* $b(q) = 0$*.*

For a bit-EIO, the various trace sets are defined as for an EIO, except that the strict quiescent traces are $StQT(S) = \{w \mid q_0 \stackrel{w}{\Longrightarrow} q \text{ and } b(q) = 1\}$*. Consequently, QET and QDT are obtained by adding ET and EDT resp. Thus,* $\sqsubseteq_{Div}$ *is defined between arbitrary EIOs and bit-EIOs.*

To bring S and D in the intended normal form, we apply some powerset construction to S and D similar to Def. 38. The result will be a deterministic bit-EIO. During the powerset construction we 'loose' the quiescent states that have incoming traces which could be extended by some output in some other state. Hence, we mark a state of the new system by bit 1, if it contains a quiescent state of the original EIO.

Before going on, we show that each bit-EIO can be translated into an equivalent EIO. If the bit-EIO is deterministic, the equivalent EIO is 'almost' deterministic, but usually not completely so. For the translation, we add for every q with $b(q) = 1$ a fresh state $\overline{q}$ that inherits all incoming transitions and all outgoing input transitions of the original state. If q is the initial state, we also add a τ-transition from the original q_0 to $\overline{q_0}$.

Definition 42 (Quiescence Correction)**.** *For a bit-EIO* $S = (Q, I, O, \delta, q_0, E, b)$*, the* quiescent correction *is the EIO* $QC(S) = (Q \,\dot\cup\, \{\overline{q} \mid b(q) = 1\}, I, O, \delta \,\dot\cup\, \delta', q_0, E)$ *with* $\delta' = \{(q', \alpha, \overline{q}) \mid (q', \alpha, q) \in \delta \text{ and } b(q) = 1\} \,\dot\cup\, \{(\overline{q}, a, q') \mid (q, a, q') \in \delta, a \in I \text{ and } b(q) = 1\} \,\dot\cup\, \{(q_0, \tau, \overline{q_0}) \mid b(q_0) = 1\}$*.*

Proposition 43. *Every bit-EIO* S *is equivalent to its quiescent correction* $QC(S)$ *w.r.t.* $\sqsubseteq_{Div}$*.*

Proof: For the set inclusion regarding EDT, recall that there are no divergence traces. In fact, it suffices to consider the strict error traces. A run for a strict error trace $w \in StET(S)$ leads to a state in E. Since all transitions of S are also in $QC(S)$, we get $w \in StET(QC(S))$. Similarly, $L(S) \subseteq L(QC(S))$.

For $w \in StQT(S)$, a run in S can reach a state q with $b(q) = 1$ when executing w. This run also exists in $QC(S)$, and the last transition can be redirected to $\overline{q}$ because $b(q) = 1$. For $w = \varepsilon$, the run in $QC(S)$ consists of $(q_0, \tau, \overline{q_0})$. Since $\overline{q}$ only inherits the outgoing input transitions of q, it is quiescent, and $w \in StQT(QC(S))$.

For the reverse inclusions, we note that a run in $QC(S)$ for trace w could use some fresh states. All incoming and outgoing transitions of fresh states are duplicates of the incoming and outgoing transitions of the original states. So the same trace w is also executable in $QC(S)$ without using the fresh copies, and the respective run exists in S as well. This shows $L(QC(S)) \subseteq L(S)$ and, noting that error states have no fresh copy, also $EDT(QC(S)) \subseteq EDT(S)$.

A run for $w \in StQT(QC(S))$ can end in an original state q, and we can use the same arguments. If it ends in some $\overline{q}$, we have $b(q) = 1$. As above, w can also be executed in S, reaching q. This shows that $w \in StQT(S)$ in any case, and we are done. All in all, we have shown the equivalence of S and $QC(S)$ w.r.t. $\sqsubseteq_{Div}$. □

Now we define the new normal form and how it can be obtained.

Definition 44 (Qui-Div-Normal Form)**.** *A bit-EIO is in* qui-div-normal form (qui-div-NF)*, if it is deterministic (hence divergence-free) and in normal form.*
By Def. 36 and Prop. 37, we can assume that we have an EIO S in divergence normal form with only error state e. The qui-div-NF of S is the bit-EIO $QDF(S)$ obtained from S as follows. First, we construct the power-EIO $QD(S)$ of S as follows:

- $Q_{QD} = \mathfrak{P}(Q)$,
- *the signature is* (I, O),
- $q_{0QD} = \{q \mid q_0 \stackrel{\varepsilon}{\Rightarrow} q\}$,
- $E_{QD} = \{P \mid e \in P\}$,
- δ_{QD} *consists of all transitions* $P \stackrel{a}{\to} \{q' \mid \exists q \in P : q \stackrel{a}{\to}\stackrel{\varepsilon}{\Rightarrow} q'\}$ *with* $a \in \Sigma$.

We restrict $QD(S)$ to the reachable part and say that $P \in Q_{QD}$ allows quiescence *if some $q \in P$ is quiescent. $QDF(S)$ is obtained from $QD(S)$ by performing the normal form construction $NF(QD(S))$, resulting in a new system whose only error state is called e again. Finally, we add the function b that assigns 1 to a state P that allows quiescence and 0 to all other states including the new error state.*

Due to the first normal form assumption, the last normalization simply merges all error states into the new e and keeps all other states and transitions. Furthermore, $b(e) = 0$. $QDF(S)$ is in normal form due to the last construction step. Hence Lemma 24 holds for all bit-EIOs in qui-div-NF.

Proposition 45. *Each EIO S is equivalent to the bit-EIO $QDF(S)$ w.r.t. $\sqsubseteq_{Div}$. $QDF(S)$ is deterministic and in normal form.*

Proof: The powerset construction gives a deterministic system. Determinism is preserved and normal form enforced by the last step.

The powerset construction preserves all languages in the automata-theoretic sense if we define some set of final states in S (e.g. $\{e\}$) and the corresponding set of states in $QD(S)$ (E_{QD} in the example). The given example shows that the EDT-semantics is preserved in the powerset construction, and it is also in the succeeding normalization. Choosing all states in S and $QD(S)$ shows the preservation of the L-semantics.

For the treatment of quiescence, we choose the set of quiescent states in S and the P allowing quiescence in $QD(S)$; hence, the $StQT$-semantics is preserved in the powerset construction. Function b is chosen such that it is also preserved in the normalization except for traces reaching some P that allows quiescence and contains the original e. Since these traces are in $EDT(QDF(S))$, at least the QDT-semantics is preserved, and we are done. □

We now define a structure the quotient will be based on. In the following we will use s, d and p to denote a state of S, D and a prospective solution P resp.

Definition 46 (Pseudo-quotient)**.** *Let S and D be bit-EIOs in qui-div-NF with $\Sigma_D \subseteq \Sigma_S$ and $O_D \subseteq O_S$. We set $I = I_S \cup O_D$ and $O = O_S \backslash O_D$.*
The pseudo-quotient *S over D is defined as the bit-EIO $S \oslash D = (\{(e_S, e_D)\}, I, O, \{(e_S, e_D) \xrightarrow{a} (e_S, e_D) \mid a \in \Sigma\}, (e_S, e_D), \{(e_S, e_D)\}, b)$, if $s_0 = e_S$. Otherwise, $S \oslash D = (S \times D, I, O, \delta, (s_0, d_0), \{(e_S, e_D)\}, b)$ where b and the transition relation are defined by the following rules:*

(Q1) $(s,d) \xrightarrow{a} (s',d)$, *if* $s \xrightarrow{a}_S s' \wedge a \in \Sigma \backslash \Sigma_D \wedge s' \neq e_S$ $(s \neq e_S)$,

(Q2) $(s,d) \xrightarrow{a} (s',d')$, *if* $s \xrightarrow{a}_S s' \wedge d \xrightarrow{a}_D d' \wedge a \in \Sigma_D \wedge s' \neq e_S$ $(s \neq e_S)$,

(Q3) $(s,d) \xrightarrow{a} (e_S,e_D)$, *if* $s \xrightarrow{a}_S e_S \wedge s \neq e_S$ *(then:* $a \in I_S \subseteq I$*)*,

(Q4) $(e_S,e_D) \xrightarrow{a} (e_S,e_D)$, *if* $e_S \xrightarrow{a}_S e_S$ *(applies for all* $a \in \Sigma$*)*,

(Q5) $(s,d) \xrightarrow{a} (e_S,e_D)$, *if* $d \not\xrightarrow{a}_D \wedge a \in I \cap \Sigma_D \wedge s \neq e_S$ *(only for* $a \in O_D$ *possible).*

$$b(s,d) = \begin{cases} 0, & \text{if } (s,d) = (e_S,e_D) \text{ or } b(s) = 0 \wedge b(d) = 1 \\ 1, & \text{otherwise.} \end{cases}$$

For a state $(s,d) \in S \oslash D$, the intuition is that (s,d) in parallel with d has only traces that s can also execute, and that (s,d) should be the coarsest state with respect to $\sqsubseteq_{Div}$ satisfying the condition.

Rule (Q1) is necessary due to the following consideration. If S has an a-transition where a is unknown to D, this can only originate from an a-transition in the quotient that we wish to construct.

Rule (Q2) is obvious in the light of the choice of alphabet in Def. 46. As $S \oslash D$ has all actions of S and D in its alphabet, it also needs an a-transition to produce such a transition at $(s,d) \| d$.

For Rule (Q1) and (Q2) we know that s cannot be e_S. Otherwise S would have a transition $e_S \xrightarrow{a}_S s' \neq e_S$. That is not possible because S is in qui-div-NF.

Rule (Q3) deals with reaching the error state in S. Obviously, (e_S,e_D) is the most general state of $S \oslash D$. Intuitively, this rule combines (Q1) and (Q2) and replaces all states (e_S,d) by (e_S,e_D). S is in qui-div-NF and hence also in normal form. Thus, all transitions leading from a non-error state to the error state are labeled with an input. All inputs of S are inputs of the pseudo-quotient. So Rule (Q3) only generates input-transitions.

Rule (Q4) generates a transition loop at the error state for all $a \in \Sigma_S$. As usual, Σ is the union of I and O. Hence, $\Sigma = I_S \cup O_D \cup O_S \backslash O_D = I_S \cup O_S = \Sigma_S$.

Rule (Q5) makes $S \oslash D$ almost input-enabled. Action $a \in I$ is blocked by d, so it can only be an output of D since D is input-enabled. The a-transition introduced here just disappears in $(S \oslash D) \| D$, since a is blocked by d.

With b we allow quiescence for most states in the quotient except for the error state and for states (s,b) where s does not allow quiescence but d

does. In the latter case, $b(s,d) = 0$ could violate a condition of bit-EIOs because (s,d) could have no outgoing output-transitions. Such an (s,d) is reached by a quiescent trace in parallel composition with d, which could be quiescent as well. The only reachable state of S with the same trace is s, which does not allow quiescence and is not the error state. Such an (s,d) will be removed in the quotient due to the next definition; cf. (F3).

The pseudo-quotient can contain pairs (s,d) where it is impossible that s has the required properties that result from the parallel composition of (s,d) and d. We call such pairs *impossible states* and remove them from the pseudo-quotient. In order to prevent the enforced reachability of impossible states, all states having an input transition to impossible states must also be removed. This pruning due to (F4) results in the quotient.

Definition 47 (Quotient)**.** *Let $S \oslash D$ be the pseudo-quotient of S over D. The set $F \subseteq S \times D$ of* impossible states *is defined as the least set satisfying the following rules:*

(F1) $s \neq e_S \wedge d = e_D$ *implies* $(s,d) \in F$,

(F2) $s \overset{a}{\not\to}_S \wedge d \overset{a}{\to}_D$ *and* $a \in O_D$ *implies* $(s,d) \in F$ *(only possible for* $s \neq e_S$*),*

(F3) $(s,d) \in F$ *whenever* $b(s) = 0 \wedge b(d) = 1$ *and* $\forall a \in O$ *: if* $(s,d) \overset{a}{\to} (s',d')$ *then* $(s',d') \in F$,

(F4) $(s,d) \overset{a}{\to} (s',d') \in F$ *and* $a \in I$ *implies* $(s,d) \in F$.

The quotient $S/\!\!/D$ is obtained by deleting all states $(s,d) \in F$ and all unreachable states except (e_S, e_D) from $S \oslash D$. This also removes any transition exiting or entering the deleted state. If $(s,d) \in S/\!\!/D$, then we write $s/\!\!/d$. If $(s_0,d_0) \notin S/\!\!/D$, then the quotient S over D is not defined.

In the remainder, we will work with the quotient being represented by the bit-EIO $S/\!\!/D$. Since it should really be an EIO, one has to replace it in the end by $QC(S/\!\!/D)$.

Rule (F1) captures the division by e_D: state e_D, in parallel with any state is an error state thus there is an error trace that S (in qui-div-NF) with $s \neq e_S$ cannot match.

Rule (F2) can only be applied for $s \neq e_S$, since e_S has for each a an outgoing transition to its self. The rule captures the situation where d has an output a that is not implemented at s. Offering an a-input-transition in the quotient would lead to an a-transition in the parallel composition with d,

while not offering a would lead to an error; both would lead to a trace in the parallel composition which an S in qui-div-NF cannot match.

Rule (F3) cuts out states which would be quiescent in the parallel composition with D (possibly after removing other states) but have no quiescent counterpart in S.

Finally, Rule (F4) propagates back all impossibilities that cannot be avoided by refining, since (s, d) must have an input a-transition.

Before we go on we will show an example for the quotient construction, see Figure 13. For the example we have $I_S = \{i_1, i_2\}$, $O_S = \{o_1, o_2\}$, $I_D = \{i_1, o_2\}$, $O_D = \{o_1\}$ and, hence, $I = \{i_1, i_2, o_1\}$, $O = \{o_2\}$ for the signature of the pseudo-quotient and quotient. Σ_S, Σ_D and Σ at the loops of the error states in the figure represent all actions of the set. All states which are mapped to 1 by b are underlined in the example. If a state is quiescent in S or D it has to be marked with 1 by function b. Additionally in S, s_0 allows quiescence; this means that the only output o_2 is optional. The result is that $s_0, s_2, s_4, s_6, d_0, d_2, d_3, d_4$ are the states which are mapped to 1 by b. All other states of S and D have 0 as value of b. In particular, each of the outputs at s_1 is optional, but one of them has to be implemented.

S has a "main cycle" $i_1 o_1 i_2 o_2$, this also occurs in $S /\!\!/ D$. D contributes i_1 and o_1 to this cycle; to identify the i_1 that has to be answered by o_1, D listens in for o_2 as input. If S performs an additional o_2 from s_0 or s_1, the construction shows that they do not fit to D. After $s_0 \xrightarrow{o_2}_S s_4 \xrightarrow{i_1}_S s_5$ the state reached is not quiescent and also does not allow quiescence. In D, with the same trace only d_4 can be reached, which is quiescent. Hence the pseudo-quotient also has the trace $o_2 i_1$ and reaches (s_5, d_4) by executing this trace. In the parallel composition of the quotient and D the trace $o_2 i_1$ is not allowed to be quiescent, because it must refine the non-quiescent trace of S. This implies that (s_5, d_4) is not allowed to be quiescent; but at the same time, it does not have an output: The only output of $S \oslash D$ is o_2, which is not possible for S in s_5. Hence the state (s_5, d_4) of the pseudo-quotient is in F due to (F3). All other quiescent states of $S \oslash D$ are mapped to 1 by b. Additionally, (s_0, d_0) allows quiescence because s_0 does so. The result is, that b is 1 for all states in $\{(s_0, d_0), (s_1, d_1), (s_2, d_2), (s_4, d_3), (s_6, e_D)\}$ and 0 for all other states of $S \oslash D$. The mapping of b for $S /\!\!/ D$ is like in $S \oslash D$ for the states left in the quotient. With $(s_4, d_3) \xrightarrow{i_1}_{S \oslash D} (s_5, d_4) \in F$ and $i_1 \in I$, (F4) is fulfilled for (s_4, d_3). The conditions of (F1) and (F2) are both satisfied in (s_6, e_D), if we choose o_1 for the a in (F2). The whole set of impossible states is $F = \{(s_4, d_5), (s_5, d_4), (s_6, e_D)\}$.

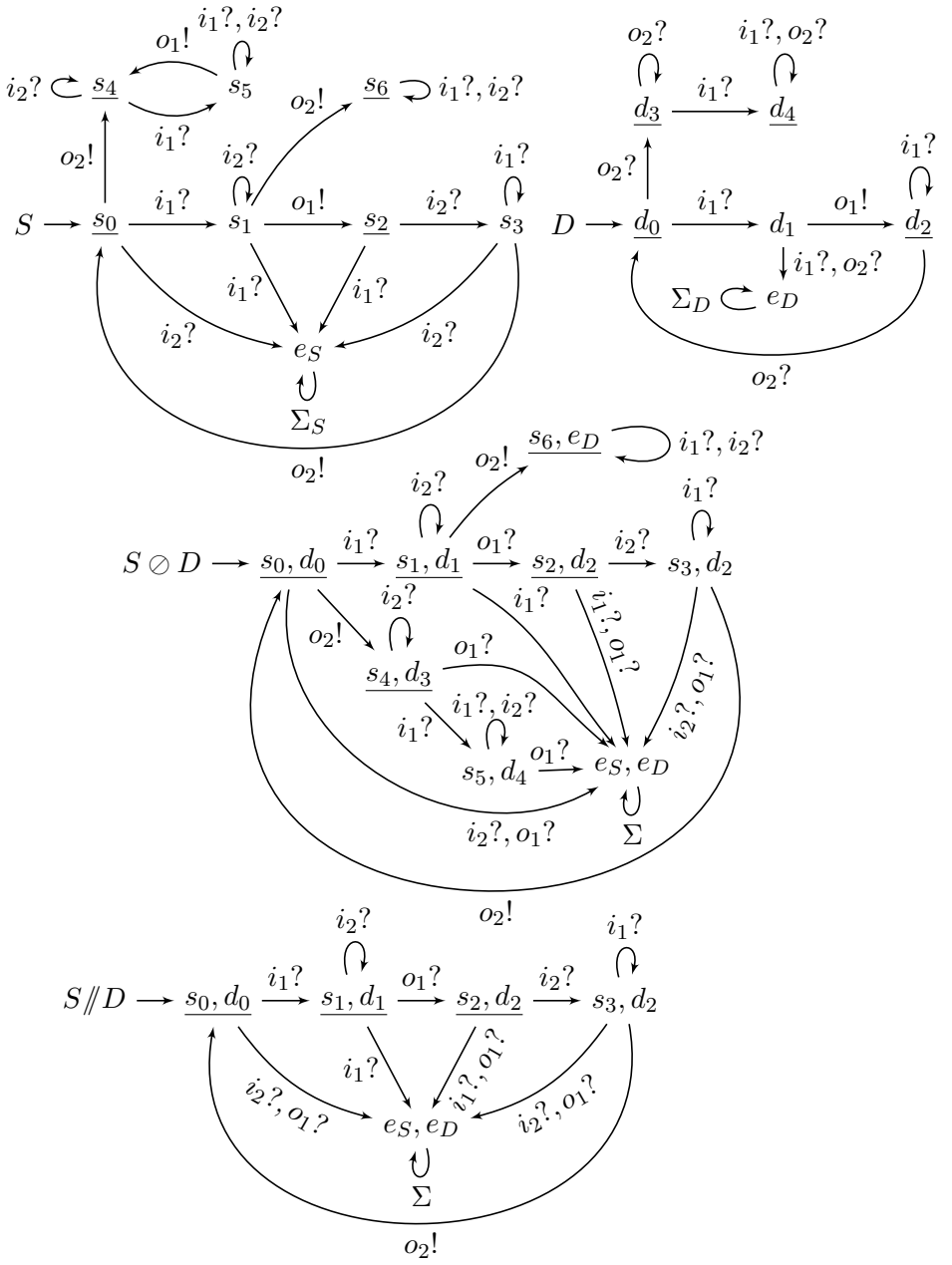

Figure 13: Example for quotient construction

Lemma 48. *In $S \oslash D$, (e_S, e_D) has no outgoing transitions to other states.*

Proof: We prove that no Q-rule can generate a violation.

(Q1)/(Q2) These rules are not applicable because $s \neq e_S$.

(Q3)/(Q4)/(Q5) All transitions resulting from these three rules have (e_S, e_D) as target state.

□

Lemma 49. *In $S \oslash D$, $(e_S, e_D) \notin F$.*

Proof: We prove this by induction on the derivation length according to the F-rules.

(F1) Since $s \neq e_S$ holds for this rule, it cannot be applied to (e_S, e_D).

(F2) To apply this rule, s must have a missing outgoing transition. If s were the error state e_S, s would have a self loop for every $a \in \Sigma_S$. Hence, (s, d) cannot be (e_S, e_D).

(F3) State (e_S, e_D) has a transition loop for all $a \in \Sigma$ and, due to Lemma 48, (e_S, e_D) has no transitions leading to other states. So $(s', d') = (e_S, e_D)$ is not in F by induction.

(F4) With the same argumentation as for Rule (F3), this rule would insert (e_S, e_D) only into F if another rule has done it before.

□

Lemma 50. *Let S and D be bit-EIOs in qui-div-NF. Then $S /\!\!/ D$ is a bit-EIO in qui-div-NF if it is defined. In particular, $S /\!\!/ D$ (if it exists) is input-enabled and deterministic.*

Proof: We assume that $S /\!\!/ D$ is defined.

In the first part, we consider the properties of a bit-EIO and, first of all, input-enabledness. Let $s /\!\!/ d \in S /\!\!/ D$ and $a \in I$. If $s = e_S$, then $d = e_D$ since otherwise (s, d) has no ingoing transitions in $S \oslash D$ and is therefore unreachable. Thus, we are done by (Q4). So let $s \neq e_S$.

We first consider $a \in I_S$. Then, we have a transition $s \xrightarrow{a}_S s'$ and get some $(s, d) \xrightarrow{a} (s', d')$ in $S \oslash D$ by (Q1) – (Q3); note that $a \in \Sigma_D$ implies $a \in I_D$ and $d \xrightarrow{a}_D d'$ for some d'. If this input-transition were deleted since

(s', d') is removed due to Def. 47, then also (s, d) would be removed due to (F4).

Second, let $a \in O_D$. If $d \not\xrightarrow{a}_D$, we are done by (Q5) and Lemma 49. Otherwise, we done by (Q2) as above, or (s, d) would be removed by (F2).

Clearly, $S /\!\!/ D$ is divergence-free since no τ-transitions are generated in Def. 46. Concerning the requirements for b we observe that, by Lemma 49, $S /\!\!/ D$ has the unique error state (e_S, e_D) and $b(e_S, e_D) = 0$ by Def. 46. For a state $s /\!\!/ d \neq (e_S, e_D)$, the only problem could be that $b(s, d) = 0$ according to Def. 46, though (s, d) is quiescent. But in this case, $b(s) = 0$ and $b(d) = 1$, so (s, d) is removed due to (F3).

In the second part, we treat the remaining properties, and start with the normal form requirements. As just noted, $S /\!\!/ D$ has the unique error state (e_S, e_D), which has a loop for all actions by (Q4) and no leaving transition by Lemma 48. All incoming transitions of the error state are generated by (Q3) and (Q5). (Q3) only generates transitions for labels in $I_S \subseteq I$ and (Q5) only generates transitions for labels in $O_D \subseteq I$.

It remains to prove determinism. S and D are already deterministic. From this, we have to show that the Q-rules generate at most one transition for each state (s, d) and action a. The F-rules only delete transitions. Therefore they cannot introduce a violation of determinism.

We note that each rule on its own can generate at most one transition for (s, d) and a. So we only have to exclude that two rules generate an a-transition for the same (s, d).

(Q1) If (Q1) generates an a-transition, a is not in Σ_D and the target state of the underlying transition in S is not the error state. This contradicts $a \in \Sigma_D$ in (Q2) and (Q5) and e_S as target state in (Q3) and (Q4).

(Q2) If (Q2) generates an a-transition, the target state of the underlying transition in S is not the error state and also D has an underlying a-transition. This contradicts e_S as target state of the first underlying transition in (Q3) and (Q4), and the non-existence of the underlying a-transition in D in (Q5).

(Q3) If (Q3) generates an a-transition from (s, d) to the error state, there must be an underlying transition in S with source state $s \neq e_S$. This contradicts e_S as source state of the underlying transition from S in (Q4). Action a is in I_S in (Q3), hence not in $O_D \subseteq O_S$. Therefore, also (Q5) cannot introduce an a-transition.

(Q4) The a-transition generated by this rule has the underlying transition $e_S \xrightarrow{a}_S e_S$. The source state of this transition makes it impossible to use Rule (Q5) for an a-transition from (s, d).

□

We note that, since $S /\!\!/ D$ is in qui-div-NF, Lemma 24 also holds for the quotient. We will now show our main result that the quotient operation above yields the coarsest bit-EIO satisfying the defining inequality. For this proof, the next lemma ensures the definiteness of $/\!\!/$.

Lemma 51. *Let S and D be bit-EIOs in qui-div-NF, and let P be an EIO with $\Sigma_D \subseteq \Sigma_S$, $O_D \subseteq O_S$, $O_P = O_S \backslash O_D$ and $I_P = I_S \cup O_D$. If $P \| QC(D) \sqsubseteq_{Div} S$, then $S /\!\!/ D$ is defined.*

Proof: We assume that P is in divergence normal form. This assumption merely simplifies the proof somewhat, but this divergence normal form has never to be computed. We will write $\rightarrow_{\|}$ and $\rightarrow_{\oslash}$ as shorthand for $\rightarrow_{p\|d}$, $\rightarrow_{s \oslash d}$. For this proof, we define the new relation $\sqsubseteq$ by: $p\|d \sqsubseteq s$ if $\exists w \in \Sigma_S^* : p_0\|d_0 \stackrel{w}{\Longrightarrow}_{\|} p\|d \wedge s_0 \stackrel{w}{\Longrightarrow}_S s$.

We will show the claim that, for all $(s, d) \in F$, there cannot exist any p with $p\|d \sqsubseteq s$, arguing that the respective trace w with its extensions would violate $P\|QC(D) \sqsubseteq_{Div} S$. Then we are done, since $p_0\|d_0 \sqsubseteq s_0$ due to $w = \varepsilon$.

We prove this claim by induction on the derivation length according to the F-rules. In each case, we assume $p\|d \sqsubseteq s$ due to w for some $p \in P$ and derive a contradiction.

(F1) $s \neq e_S \wedge d = e_D$: Here $p\|d$ is an inherited error in $P\|QC(D)$. Hence $w \in ET(P\|QC(D)) \subseteq EDT(P\|QC(D))$, but $w \notin ET(S) = EDT(S)$ since S is in qui-div-NF.

(F2) $s \stackrel{q}{\not\rightarrow}_S, d \xrightarrow{a}_D$ and $a \in O_D$: $s \neq e_S$ because e_S has transition loops for all $a \in \Sigma_S$. By $p\|d \sqsubseteq s$ and $wa \notin EL(S) = EDL(S)$, we know $p\|d \stackrel{q}{\not\rightarrow}_{\|}$. This can only happen if $p \stackrel{q}{\not\rightarrow}_P$; but a is an input of P, which is input-enabled – a contradiction.

(F3) $b(s) = 0$, $b(d) = 1$ and $\forall a \in O$: if $(s, d) \xrightarrow{a}_{\oslash} (s', d')$ then $(s', d') \in F$: Here, s cannot be e_S because then (s, d) would be (e_S, e_D), and this is not possible due to Lemma 48 and Lemma 49.

By $b(d) = 1$, we also have $p_0\|d_0 \stackrel{w}{\Longrightarrow}_{\|} p\|\overline{d}$. If p is quiescent then $w \in StQT(P\|QC(D))$, but $w \notin QDT(S)$ since $b(s) = 0$, $s \neq e_S$ and s is in qui-div-NF.

Thus, p has to have an outgoing transition $p \xrightarrow{a}_P p'$ for some $a \in O$ and p'. Action a is an input for D or not in its alphabet. Hence, $p\|d$ will inherit the a-transition from p as $p\|d \xrightarrow{a}_{\|} p'\|d'$ (where d' could be d). The result is $wa \in L(P\|QC(D)) \subseteq EDL(P\|QC(D))$. Since $P\|QC(D) \sqsubseteq_{Div} S$, we know $wa \in EDL(S) = EL(S)$; since S is in qui-div-NF, this implies $s \xrightarrow{a}_S s'$ for some s'.

Given our considerations about d and a, we have $(s,d) \xrightarrow{a}_{\oslash} (s',d')$ by one of (Q1),(Q2) and (Q3). Thus, $(s',d') \in F$ by assumption of (F3) and (s',d') satisfies our claim by induction. At the same time, we have $p_0\|d_0 \overset{wa}{\Longrightarrow}_{\|} p'\|d' \wedge s_0 \overset{wa}{\Longrightarrow}_S s'$, a contradiction.

(F4) $(s,d) \xrightarrow{a}_{\oslash} (s',d') \in F$ and $a \in I$: Our claim holds for (s',d') by induction hypothesis, and the transition is due to one of the Q-rules:

(Q1)/(Q2) Action a is an input for P and P has to be input-enabled. Hence $p \xrightarrow{a}_P p'$ for some p'. In parallel with d, we get the transition $p\|d \xrightarrow{a}_{\|} p'\|d'$ because in (Q1) a is not in Σ_D ($d' = d$) and (Q2) requires d to have an a-transition to d'. For the target state $p'\|d' \sqsubseteq s'$ holds. This contradicts $(s',d') \in F$.

(Q3)/(Q4)/(Q5) Transitions that exist due to these three rules have (e_S, e_D) as target state, which cannot be in F due to Lemma 49.

□

The parallel composition of EIOs is defined if the output action sets are disjoint. The output action set of P is defined as $O_P = O_S \backslash O_D$ in Lemma 51 and the following theorem. This set and O_D are clearly disjoint.

Theorem 52. *Let S and D be bit-EIOs in qui-div-NF and P an EIO such that $\Sigma_D \subseteq \Sigma_S$, $O_D \subseteq O_S$, $O_P = O_S \backslash O_D$ and $I_P = I_S \cup O_D$. Then, $P \sqsubseteq_{Div} S/\!\!/D$ with $S/\!\!/D$ defined iff $P\|QC(D) \sqsubseteq_{Div} S$.*

Proof: Again, we can assume that P is in divergence normal form.

„⇒“: We show that the trace set inclusions for $P\|QC(D) \sqsubseteq_{Div} S$ hold, if they hold for $P \sqsubseteq_{Div} S/\!\!/D$.

First, we take a prefix minimal $w \in EDT(P\|QC(D))$ and show that w is also in $EDT(S)$. Some wv with $v \in O_S^*$ reaches an error state in $P\|QC(D)$, since divergence is not possible.

If this fault is inherited from D, $wv|_{\Sigma_D}$ is a strict error trace in D and P can execute wv. P is a refinement of the defined quotient $S/\!\!/D$, so wv is a trace in $S/\!\!/D$. This must be due to the Q-rules without (Q5) and

thus wv must also be an executable trace in S; in particular, none of the states reached in this trace are allowed to fulfill (F1). Hence wv must be in $EDT(S)$ and, by $v \in O_S^*$, also $w \in EDT(S)$.

If the error in $P \| QC(D)$ is inherited from P, we know $wv \in EDT(P) \subseteq EDT(S /\!\!/ D)$. The underlying run in $S /\!\!/ D$ reaches the error state (e_S, e_D) and the transitions of this run exist due to the Q-rules. (Q5) cannot play a rôle here because $wv|_{\Sigma_D}$ is a trace for D. Hence, S takes part in all transitions to execute wv. The transition in $(S /\!\!/ D)$ which reaches the error state results from (Q3), and the underlying transition of S reaches e_S. Thus $wv \in EDT(S)$ and, by $v \in O_S^*$, also $w \in EDT(S)$.

For the next trace set inclusion, it is enough to show for a $w \in StQT(P \| QC(D))$ that it is also contained in $QDT(S)$. Since w reaches a quiescent state in the parallel composition of P and $QC(D)$, Lemma 16 tells us that this can only happen if both states of the components are already quiescent. Thus, P reaches a quiescent state with w and $QC(D)$ with $w|_{\Sigma_D}$. The quiescent trace of P is a quiescent or an error trace of $S /\!\!/ D$. The trace in $S /\!\!/ D$ results from the Q-rules which require underlying transitions in S.

Since $S /\!\!/ D$ is in qui-div-NF, it can really execute this trace. The last state reached is either the error state and we can conclude from the Q-rules that $w \in EDT(S)$, or the last state $s /\!\!/ d$ counts as quiescent. In the latter case, all transitions on the run in $S /\!\!/ D$ are due to (Q1) and (Q2). Hence, the run in $QC(D)$ ends in $\overline{d}$ (or maybe d, if d is quiescent), i.e. $b(d) = 1$. Since $b(s, d) = 1$, this implies $b(s) = 1$ and $w \in StQT(S)$.

As the last point, we have to show $EDL(P \| QC(D)) \subseteq EDL(S)$. With the above argumentation, it is enough to consider $w \in L(P \| QC(D)) \setminus EDT(P \| QC(D))$. This w is executable in the parallel composition of P and $QC(D)$. Hence $w \in L(P) \subseteq EDL(S /\!\!/ D)$. Again, w is a trace in $S /\!\!/ D$ due to the Q-rules without (Q5). Thus, it is also a trace in S and $w \in L(S) \subseteq EDL(S)$.

„$\Leftarrow$“: We show that the trace set inclusions for $P \sqsubseteq_{Div} S /\!\!/ D$ hold, if they hold for $P \| QC(D) \sqsubseteq_{Div} S$. With Lemma 51 we know that $S /\!\!/ D$ is defined here.

First, for $w \in EDT(P)$, we have to show that w is also in $EDT(S /\!\!/ D)$. Since P is in normal form, it can execute the trace.

If D can match all actions on $w|_{\Sigma_D}$, w is in $EDT(P \| QC(D))$ and executable. With $P \| QC(D) \sqsubseteq_{Div} S$, we conclude that $w \in EDT(S)$. This can only result from a run in S which reaches e_S. With (Q1), (Q2), (Q3) and (Q4), we also get a run for w in $S \oslash D$; as in the proof of Lemma 51, all

states on this run are not in F, hence in $S/\!\!/D$. Therefore, the run reaches the error state (e_S, e_D), proving $w \in EDT(S/\!\!/D)$.

If D is not able to match all actions of $w|_{\Sigma_D}$, this is due to a missing output transition after the execution of a prefix of $w|_{\Sigma_D}$. The corresponding prefix v of w is executable in the parallel composition of P and $QC(D)$ and, by $P\|QC(D) \sqsubseteq_{Div} S$, v is also in the language of S. With the first four Q-rules, this prefix is also executable in $S/\!\!/D$ as above. After this prefix, a state is reached in $S/\!\!/D$ where the conditions of (Q5) are fulfilled for the next action on w. Hence, a prefix of w reaches the error state in $S/\!\!/D$ implying $w \in EDT(S/\!\!/D)$.

It is enough to consider some $w \in StQT(P)\backslash EDT(P)$ for the next inclusion. If D cannot match all actions for $w|_{\Sigma_D}$, w is an error trace in $S/\!\!/D$ like argued above. So we consider D to have $w|_{\Sigma_D}$ as an executable trace.

If $w|_{\Sigma_D} \in EDT(D)$, $QC(D)$ passes an error on to the parallel composition, and with $P\|QC(D) \sqsubseteq_{Div} S$ also $w \in EDT(S)$ follows. From the Q-rules we conclude that $w \in EDT(S/\!\!/D) \subseteq QDT(S/\!\!/D)$.

If $w|_{\Sigma_D} \in StQT(D)\backslash EDT(D)$, the parallel composition $P\|QC(D)$ has w as strict quiescent trace. With $P\|QC(D) \sqsubseteq_{Div} S$, we get $w \in QDT(S)$. If $w \in EDT(S)$, $w \in EDT(S/\!\!/D)$ follows with the above argumentation. If $w \in StQT(S)\backslash EDT(S)$, S reaches a state s with $b(s) = 1$ after w. Again, $S/\!\!/D$ can perform w reaching some $s/\!\!/d$. Since $b(s/\!\!/d) = 1$, we have $w \in StQT(S/\!\!/D)$, and in any case $w \in QDT(S/\!\!/D)$.

If $w|_{\Sigma_D} \in L(D)\backslash QDT(D)$, the parallel composition $P\|QC(D)$ can execute w but does not reach any faulty state along the way. With $P\|QC(D) \sqsubseteq_{Div} S$ we have $w \in EDL(S)$. If $w \in EDT(S)$, we are again done with the arguments from above. For $w \in L(S)\backslash EDT(S)$, we get $w \in L(S/\!\!/D)$ with (Q1) and (Q2). The state in $QC(D)$ which is reached by $w|_{\Sigma_D}$ cannot be quiescent, hence b maps this state to 0. Thus, with the rules for b of $S \oslash D$, the state in $S/\!\!/D$ which is reached by w is mapped to 1 and $w \in StQT(S/\!\!/D) \subseteq QDT(S/\!\!/D)$.

The last inclusion we have to show is $L(P)\backslash EDT(P) \subseteq EDL(S/\!\!/D)$. We take a $w \in L(P)\backslash EDT(P)$. If D cannot match all actions which are required for $w|_{\Sigma_D}$, w is an error trace in $S/\!\!/D$ like argued above. So we consider D to have $w|_{\Sigma_D}$ as an executable trace.

If $w|_{\Sigma_D} \in EDT(D)$, $QC(D)$ passes an error on to the parallel composition and, with $P\|QC(D) \sqsubseteq_{Div} S$, also $w \in EDT(S)$ follows. As above, we get with the Q-rules that $w \in EDT(S/\!\!/D) \subseteq EDL(S/\!\!/D)$.

If $w|_{\Sigma_D} \in L(D)\backslash EDT(D)$, we get $w \in L(P\|QC(D))$ and conclude $w \in EDL(S)$ with $P\|QC(D) \sqsubseteq_{Div} S$. If $w \in EDT(S)$, we are done with the arguments from above. Otherwise, we get $w \in L(S/\!\!/D) \subseteq EDL(S/\!\!/D)$ with (Q1) and (Q2). □

From this theorem we can also conclude that $/\!\!/$ is monotonous w.r.t. $\sqsubseteq_{Div}$ in the left argument.

Theorem 53. *Let S_1, S_2, D be bit-EIOs in qui-div-NF with $S_1 \sqsubseteq_{Div} S_2$. If $S_1/\!\!/D$ is defined, then $S_2/\!\!/D$ is defined and $S_1/\!\!/D \sqsubseteq_{Div} S_2/\!\!/D$.*

Proof: If $S_1/\!\!/D$ is defined, then $QC(S_1/\!\!/D)\|QC(D) \sqsubseteq_{Div} S_1$ by Thm. 52. Applying the assumption $S_1 \sqsubseteq_{Div} S_2$, transitivity of $\sqsubseteq_{Div}$ and Thm. 52 again, we conclude that $S_1/\!\!/D \sqsubseteq_{Div} S_2/\!\!/D$; in particular, $S_2/\!\!/D$ is also defined. □

7 Conclusion

A refinement preorder should ensure that some desired properties are preserved in a refinement step, and it should support compositional reasoning. Optimally, it rejects a prospective refinement only if the two goals make it necessary. With the coarsest-precongruence approach we followed in this paper, one can find such optimal preorders. The approach is most attractive in cases where one starts from a simple property, but gets a preorder that preserves much stronger properties. In this paper, we considered the property that a system cannot run into a fault autonomously, with three variants of fault. In the first case, a fault is a communication mismatch, called error. We characterized the coarsest precongruence with the sets ET and EL. The error traces of the first set also describe how errors can arise non-autonomously in parallel compositions, the second set restricts also the error-free behaviour of a refinement.

We obtained two further precongruences on the basis that a fault can also be a quiescence or can also be a quiescence or divergence. The last of these precongruences shows that, for an optimal preorder, divergence is as catastrophic as an error (while quiescence is less harmful). This is in contrast to the declarative semantics presented in [5]. We showed that all our precongruences are also compositional w.r.t. hiding and presented a conjunction operator for each of them. Finally, we introduced a quotient operator (being adjoint to parallel composition). While the quotient results in [5] are restricted to deterministic systems, we treated arbitrary EIOs

here. For defining the quotient and proving its characteristic property, we developed a new structure, and these new bit-EIOs can represent all EIOs in an almost deterministic way.

In particular in the context of conjunction, it would also be interesting to allow alphabet extension in a refinement step. Think of two EIOs that specify two properties concerning different sets of actions. Then, a common refinement should have all actions, i.e. more than any of the two conjuncts; cf. e.g. [3]. In [5], an alphabet change is possible, resulting in more inputs and fewer outputs; this is technically easier in a setting that regards outputs as a source for errors. But the scenario just described makes clear why we want to be able to also enlarge the set of outputs.

References

[1] F. Aarts and F. W. Vaandrager. Learning I/O Automata. In *CONCUR 2010*, volume 6269 of *LNCS*, pages 71–85. Springer, 2010. doi:10.1007/978-3-642-15375-4_6.

[2] S. D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A Theory of Communicating Sequential Processes. *J. ACM*, 31(3):560–599, 1984. doi:10.1145/828.833.

[3] F. Bujtor, S. Fendrich, G. Lüttgen, and W. Vogler. Nondeterministic Modal Interfaces. *Theor. Comput. Sci.*, 642:24–53, 2016. doi:10.1016/j.tcs.2016.06.011.

[4] F. Bujtor and W. Vogler. Error-Pruning in Interface Automata. *Theor. Comput. Sci.*, 597:18–39, 2015. doi:10.1016/j.tcs.2015.06.047.

[5] C. Chilton, B. Jonsson, and M. Z. Kwiatkowska. An Algebraic Theory of Interface Automata. *Theor. Comput. Sci.*, 549:146–174, 2014. doi:10.1016/j.tcs.2014.07.018.

[6] L. de Alfaro and T. A. Henzinger. Interface Automata. In *ESEC/FSE 2001*, pages 109–120. ACM, 2001. doi:10.1145/503209.503226.

[7] L. de Alfaro and T. A. Henzinger. Interface-Based Design. In *Engineering Theories of Software Intensive Systems*, pages 83–104. Springer, 2005. doi:10.1007/1-4020-3532-2_3.

[8] R. De Nicola and M. Hennessy. Testing Equivalences for Processes. *Theor. Comput. Sci. 34*, pages 83–133, 1984. doi:10.1016/0304-3975(84)90113-0.

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

[10] G. Lüttgen and W. Vogler. Conjunction on Processes: Full Abstraction Via Ready-Tree Semantics. *Theor. Comput. Sci.*, 373(1-2):19–40, 2007. doi:10.1016/j.tcs.2006.10.022.

[11] B. Randell and M. Koutny. Failures: Their Definition, Modelling and Analysis. In *ICTAC 2007*, volume 4711 of *LNCS*, pages 260–274. Springer, 2007. doi:10.1007/978-3-540-75292-9_18.

[12] A. Schinko. Kommunikationsfehler, Verklemmung und Divergenz bei Interface-Automaten. B.Sc. Thesis, Universität Augsburg [obtainable from the author], 2016.

[13] R. Segala. Quiescence, Fairness, Testing, and the Notion of Implementation (Extended Abstract). In *CONCUR '93*, volume 715 of *LNCS*, pages 324–338. Springer, 1993. doi:10.1007/3-540-57208-2_23.

[14] J. Tretmans. Test Generation With Inputs, Outputs, and Quiescence. In *Tools and Algorithms for Construction and Analysis of Systems, Second International Workshop, TACAS '96*, volume 1055 of *LNCS*, pages 127–146. Springer, 1996. doi:10.1007/3-540-61042-1_42.