UNIVERSITY OF TWENTE.

Formal Methods and Tools
Faculty of EEMCS

Master of Science Thesis

Quiescent Transition Systems

Author:
Gerjan Stokkink, B.Sc.

Examination Committee:
Mw.dr. Mariëlle Stoelinga
Mark Timmer, M.Sc.
Prof.dr.ir. Arend Rensink

August 22, 2012

And in the naked light I saw
Ten thousand people, maybe more
People talking without speaking
People hearing without listening
People writing songs that voices never share
And no one dared
Disturb the sound of silence

Simon & Garfunkel - *The Sound of Silence*

Abstract

Quiescence is a fundamental concept in modelling system behaviour, as it explicitly represents the fact that, in certain system states, no output is provided by the system. The notion of quiescence is also essential to model-based testing: if a particular implementation under test does not provide any output, then the test evaluation algorithm must decide whether to allow this behaviour, or not. A Suspension Automaton (SA) is a kind of labelled transition system in which observations of quiescence are explicitly represented with special δ-transitions. SAs form the basic building block on which the well-known `ioco` model-based testing framework is based.

The SA model, however, has a number of flaws. First of all, a SA is not defined as an entity in itself, and cannot be built from scratch. Secondly, its properties have not been fully investigated yet. Thirdly, and most importantly, the SA model does not allow nondeterminism or divergence to occur, thereby limiting the number of systems that can be naturally modelled. To address these limitations, we introduce in this thesis the so-called Quiescent Transition Systems (QTSs), which form a fully formalised alternative to the existing SAs. We also introduce Divergent Quiescent Transition Systems (DQTSs), a more complex variant on QTSs which allow (state-finite) divergence to occur.

We show how QTSs and DQTSs can be created from existing generic models by an operation called deltafication. Furthermore, we define the three familiar automata-theoretical operations of determinisation, parallel composition and action hiding for these models, and show that (D)QTSs are closed under these operations. Additionally, we prove that the operation of deltafication is commutative with all of these operations. Finally, we provide an evaluation in which we compare QTS, DQTSs and SAs. We illustrate that in the context of test-based modelling, the use of (D)QTSs offers several advantages over SAs, and recommend that the `ioco` theory be reformulated in terms of the (D)QTS model.

Acknowledgements

I would like to extend my sincere gratitude to my supervisors Mariëlle, Mark and Arend, who have guided me patiently throughout the long process of writing this thesis with their sound advice, useful comments, and in particular their motivational support. Without their efforts I would never have been able to finish this thesis in time. In particular I would like to thank Mark for his outstanding work in reviewing every chapter of this thesis, some many times over, and his invaluable help with finishing some of the more complex proofs.

I would also like to thank my parents and sisters, for always being there for me. When the going got tough they were always there to support me. I am also very grateful to my student counsellor, Renate, who helped me tirelessly with many DUO-related issues. Many thanks also to Gijs for proofreading my thesis one last time.

Finally, the process of writing this thesis would have been far less enjoyable without the camaraderie of my colleagues in the *Labruimte FMT* and the many friends I made within the FMT group. In particular, I would like to thank (in no particular order) Tom, Marina, Lesley, Freark, Harold, Ronald, Vincent, Maarten, Jaco, Stefano, Stefan, and Axel, amongst many others. The fun times and lively discussions I have had with you all certainly made me feel at home at all times.

Enschede, August 2012
Gerjan Stokkink

Table of Contents

Chapter 1

Introduction

Quiescence is a fundamental concept in modelling system behaviour. Quiescence explicitly represents the fact that, in certain system states, no output is provided by the system. The absence of outputs is often essential: an ATM, for instance, should deliver the requested amount of money only once, not twice (see Figure 1.1). This means that the ATM's state just after paying out money (s_0 in Figure 1.1) should be quiescent: it should not produce any output until further input is given. On the other hand, the state before paying out (s_3 in Figure 1.1) should clearly not be quiescent. Hence, quiescence can also sometimes be considered erroneous behaviour. Consequently, the notion of quiescence is essential in model-based testing: if a particular implementation under test does not provide any output, then the test evaluation algorithm must decide whether to produce a pass verdict (allowing quiescence at this point) or a fail verdict (prohibiting quiescence at this point).

The notion of quiescence was first introduced by Vaandrager [Vaa91] to obtain a natural extension of the notion of a terminal or blocking state: if a system is input-enabled (i.e., always ready to receive inputs), then no states are blocking, since each state has outgoing input transitions. However, quiescence can still be used to denote the fact that a state would be blocking when considering only the output actions. In the context of model-based testing, Tretmans introduced the notion of *repetitive quiescence* [Tre96a, Tre96b]. Repetitive quiescence emerged from the need to continue testing, even in a quiescent state: in the ATM example above, we need to test further behaviour that arises from the (quiescent) state after providing money. To further accommodate these needs, Tretmans introduced the *Suspension Automaton* (SA) as an auxiliary concept; an SA is a Labelled Transition System (LTS) in which the occurrence of quiescence is represented explictly using special δ-labelled transitions.

Example 1.1. Consider the ATM automaton given in Figure 1.1. The states s_0 and s_1 are quiescent, since they do not have any outgoing output transitions. To obtain the Suspension Automaton corresponding to such a system, Tretmans adds self-loops, labelled with the special quiescence label δ, to each quiescent state. □

While the papers mentioned above all convincingly argued the need for quiescence, none of them presents a comprehensive theory of quiescence. Firstly, quiescence is not treated as a first-class citizen: although the Suspension Automaton is used during testing, it is not defined as an entity in itself, and cannot be built from scratch. Therefore, quiescence cannot be used to specify systems, and neither is it clear what properties a SA satisfies or should

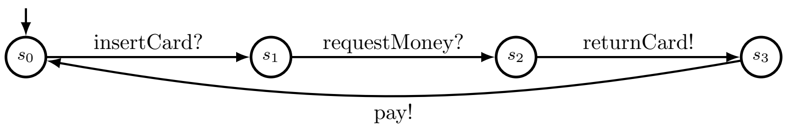


Figure 1.1: A very basic ATM.

satisfy. Since conformance relations such as `ioco` are defined based on 'suspension traces', which are the traces of a SA, it seems much more appealing to directly start from these Suspension Automata and base the theory on them. Secondly, essential compositional operators like parallel composition and action hiding have been defined for LTSs and some of their subtypes, but have not been studied for SAs at all. Therefore, it was still an open question to what extent these operators could be lifted to the setting of quiescence. Finally, the occurence of nondeterminism or divergence is explicitly disallowed for SAs, thereby essentially limiting the number of systems that can be naturally modelled using SAs.

In this thesis, we seek to remediate the shortcomings of previous work by introducing *Quiescent Transition Systems* (QTSs). QTSs form a new class of LTSs in which quiescence can be represented explicitly using special δ-transitions, and are a fully-formalised alternative to Tretmans' Suspension Automata. Whereas SAs are always constructed by adding δ-transitions to existing LTSs and subsequently determinising [Tre08], QTSs are defined in a precise manner as stand-alone entities, can be built from scratch and need not necessarily be deterministic; divergence, on the other hand, is still disallowed. The introduction of QTSs is a first step towards a comprehensive theory of quiescence, and they form a solid basis that we subsequently extend by introducing *Divergent Quiescent Transition Systems* (DQTSs). DQTSs do allow (state-recurrent) divergence to occur, and therefore allow more modelling freedom. This comes at the cost of a more complex action hiding operation, however. Together, the QTS and DQTS models form the main contribution of this thesis. An earlier, less streamlined, version of the QTS model was introduced by the author in [STS12a, STS12b].

Starting point in our theory for both QTSs and DQTSs is the observation that, when treating quiescence as a first-class citizen, certain restrictions regarding the occurrence of δ-transitions need to be put in place. For instance, it should never be the case that a δ-transition is followed by an output, as this would contradict the meaning of quiescence. As another example, we do not allow a δ-transition to enable additional behaviour; after all, it would not make much sense if our observation of the absence of outputs impacts the system. In this paper we present and discuss four such rules, that restrict the domain of all possible (D)QTSs to the sensible subclass of *well-formed* (D)QTSs. In [Wil07], four similar, but more complex, rules for *valid* deterministic SAs are discussed. We show that the classes of well-formed (D)QTSs and valid SAs are equivalent in terms of expressible behaviour.

Furthermore, we define the aforementioned well-known automata-theoretical operations on QTSs and DQTSs: determinisation, parallel composition and action hiding. These operations are very important, as they allow a modular approach to system specification. Additionally, we explain how to obtain a (D)QTS from an existing generic labelled transition system by a process called *deltafication.* We show that our four well-formedness requirements are preserved by all of these operations, hence well-formed (D)QTSs are closed under all these operations. Furthermore, we show that the operation of deltafication is commutative with the operations of determinisation, parallel composition and action hiding. These two results are very desireable, as they enable superiour modelling freedom compared to SAs.

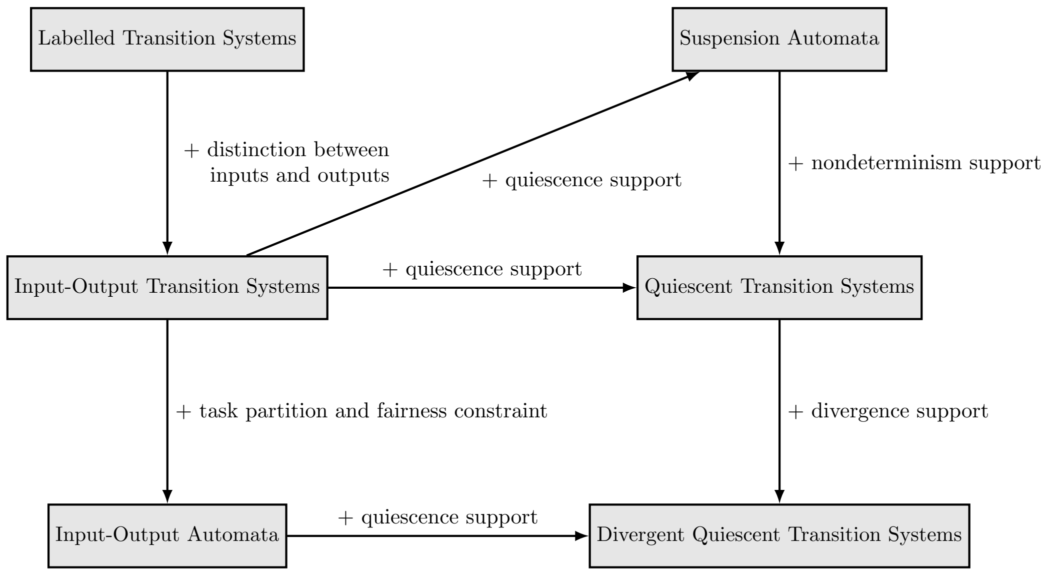

Labelled Transition Systems
Suspension Automata
+ distinction between inputs and outputs
+ quiescence support
+ nondeterminism support
Input-Output Transition Systems
+ quiescence support
Quiescent Transition Systems
+ task partition and fairness constraint
+ divergence support
Input-Output Automata
+ quiescence support
Divergent Quiescent Transition Systems

Figure 1.2: The various automata introduced in this thesis, and their main relationships.

Finally, we conclude our thesis by comparing the SA, QTS and DQTS models. Using (D)QTSs rather than SAs to model systems clearly offers several advantages. First of all, a wider range of systems can be modelled because the determinism and convergence requirements of SAs have been dropped. Furthermore, the desirable compositional properties of (D)QTSs ensure that when using a testing framework like `ioco`, the specifications of complex systems can easily be divided up, modelled as separate components, and tested more efficiently. Hence, (D)QTSs look like a promising new formalism in the context of model-based testing.

In order to succintly and accurately describe the syntax and semantics of the QTS and DQTS formalisms, a number of different automata are introduced throughout this thesis. All these automata are variants of the Labelled Transition System (LTS) formalism: an LTS is an abstract machine which describes the behaviour of a system in terms of states and transitions between them. Each transition, in turn, is associated with a label, which represents a particular action. We examine two standard LTS variants: the so-called Input-Output Transition System (IOTS) and Input-Output Automaton (IOA) models. An IOTS is simply an LTS in which the set of labels is divided into disjoint sets of inputs and outputs, and are perfectly suited to describe the behaviour of reactive systems. Therefore, IOTSs have been chosen to form the basis of both the Suspension Automaton (SA) and Quiescent Transition System (QTS) models, which extend IOTSs with support for quiescence observations using special δ-transitions. IOAs, on the other hand, are a more general type of IOTSs in which the notion of fairness is explicitly encoded using a so-called task partition. This notion of fairness is required to unambiguously capture the semantics of quiescence in the presence of divergence. Consequently, IOAs form the basis of the Divergent Quiecent Transition System (DQTS) model, which extends the QTS model with support for divergence. Figure 1.2 visualises the main relationships between the models that are discussed in this thesis.

The rest of this thesis is organised as follows. In Chapter 2 we introduce the three standard models mentioned above: Labelled Transition Systems, Input-Output Transition Systems (IOTSs) and Input-Output Automata (IOAs). Aside from these three models we also take a look at the Suspension Automaton formalism. With the aid of this background material, we are ready to introduce the IOTS-based Quiescent Transition System model in Chapter 3. We take a look at the main properties of the model, introduce the familiar compositional operations, and explore the various closure and commutative properties of the QTS model in relation to these operations. Subsequently, in Chapter 4, we similarly introduce and explore the IOA-based Divergent Quiescent Transition System model, which, as mentioned before, extends the QTS model with support for divergence. Finally, in Chapter 5, we compare the SA, QTS and DQTS models and draw various conclusions regarding their potential uses. We also offer some pointers for possible future work.

Chapter 2

Background

In this chapter, we review three important system specification models from the literature: Labelled Transitions Systems (LTSs), which model systems using states, actions and transitions; Input-Output Transition Systems (IOTSs), which extend LTSs by distinguishing between input and output actions; and finally Input-Output Automata, which further extend IOTSs with multiple internal actions and action partitions to formalise the notion of fair executions. For each of these models, we will define the standard operations of determinisation, parallel composition and action hiding. The IOTS and IOA models constitute the basis for the Quiescent Transition System (QTS) and Divergent Quiescent Transition System (DQTS) models that we will introduce in Chapter 3 and Chapter 4, respectively. We will also introduce Suspension Automata (SAs), which are an extension of IOTSs that support the notion of quiescence, i.e., the observation of an absence of outputs. At the end of this chapter, a short overview of the main properties of the different models will be given.

2.1 Notational Preliminaries

Before introducing the various models, we first need to establish some standard notations.

A *sequence* $\sigma = a_1\, a_2\, \ldots\, a_n$ is a (possibly infinite) ordered list of elements from a set L. We define the length of σ, denoted $|\sigma|$, as n. The empty sequence is denoted ϵ. We use L^* to denote the set of all finite sequences over L, L^ω to denote the set of all infinite sequences over L, and L^∞ to denote the set of all sequences over L, i.e., $L^\infty = L^* \cup L^\omega$. Given two sequences $\rho \in L^*$ and $\upsilon \in L^\infty$, we denote the *concatenation* of ρ and υ as $\rho \cdot \upsilon$ or simply $\rho\,\upsilon$. Note that $\epsilon \cdot \rho = \rho \cdot \epsilon = \rho$.

The *projection of an element* $a \in L$ *on* $L' \subseteq L$, denoted $a \upharpoonright L'$, equals a if $a \in L'$ and ϵ otherwise. The projection of a sequence $\sigma = a\,\sigma'$ is defined inductively by $\sigma \upharpoonright L' = (a\,\sigma') \upharpoonright L' = (a \upharpoonright L') \cdot (\sigma' \upharpoonright L')$. The projection of a set of sequences Z is defined as $Z \upharpoonright L' = \{\, \sigma \upharpoonright L' \mid \sigma \in Z \,\}$.

We use $\wp(L)$ to denote the *power set* of the set L. A set $P \subset \wp(L)$ such that $\emptyset \notin P$ is a *partition* of L if $\bigcup P = L$ and $p \neq q$ implies $p \cap q = \emptyset$ for all $p, q \in P$.

Finally, we follow [BK08] in using the notation $\exists^\infty$ for 'there exist infinitely many'. Hence, the (valid) statement 'there exist infinitely many integers greater than zero' can be denoted as $\exists^\infty j \in \mathbb{Z} \,.\, j > 0$.

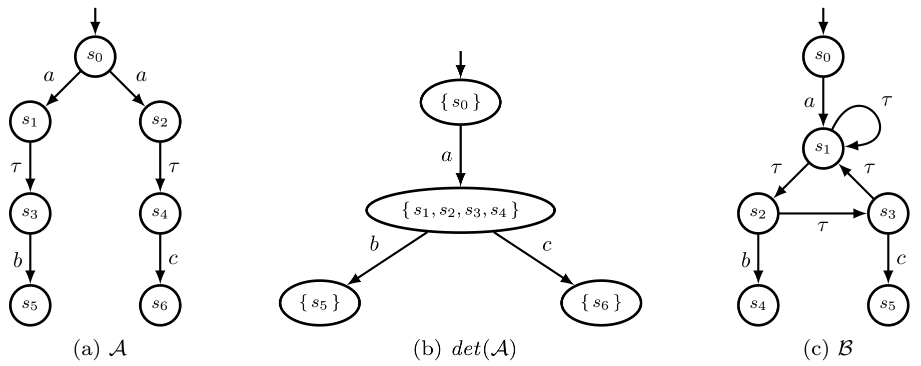


Figure 2.1: Visual representations of the LTSs $\mathcal{A}$, $det(\mathcal{A})$ and $\mathcal{B}$.

2.2 Labelled Transition Systems

2.2.1 The LTS Model

Labelled Transition Systems (LTSs) are a well-known formalism for modelling the behaviour of processes or systems. A LTS consists of a set of states, a set of transitions between these states, and a set of actions. Each state of the LTS represents a particular state which the process or system may occupy during its operation. The set of transitions define how the LTS can move from one state to the other by executing particular actions from the set of actions. With every LTS a special label τ is associated, which represents an unobservable, internal action.

Definition 2.1. A *Labelled Transition System* (LTS) is a tuple $\mathcal{A} = \langle S, S^0, L, \rightarrow \rangle$, such that:

- S is a non-empty set of states;
- $S^0 \subseteq S$ is a non-empty set of initial states;
- L is a set of labels, each representing a different action. We require $\tau \notin L$;
- $\rightarrow \ \subseteq S \times (L \cup \{ \tau \}) \times S$ is the transition relation, and defines the transitions that are possible between the states of the LTS. Each transition is marked with a label to indicate which action is responsible for the transition.

We define $L^\tau = L \cup \{ \tau \}$.

Given a LTS $\mathcal{A}$, we denote its components by $S_\mathcal{A}$, $S^0_\mathcal{A}$, $L_\mathcal{A}$ and $\rightarrow_\mathcal{A}$; we omit the subscript when it is clear from the context which LTS is referred to. We will use the terms label and action interchangeably.

Example 2.2. Figure 2.1a visualises a LTS $\mathcal{A}$ with $S_\mathcal{A} = \{ s_0, s_1, s_2, s_3, s_4, s_5, s_6 \}$, $S^0_\mathcal{A} = \{ s_0 \}$, and $L_\mathcal{A} = \{ a, b, c \}$. We represent states by circles and transitions by arrows; each arrow in turn is labelled with the associated action for that particular transition. The initial state is marked by an arrow without a source state. The state labels are left out when the identities of the states are irrelevant. □

Throughout this report, we use the following notations to describe transitions between states.

Definition 2.3 (Transitional notations)**.** Let $\mathcal{A} = \langle\, S, S^0, L, \rightarrow \,\rangle$ be an LTS with $s, s' \in S$, $a, a_i \in L^\tau$, $b, b_i \in L$, and $\sigma \in L^\infty$, then

$$\begin{array}{lll}
s \xrightarrow{a} s' & =_{\text{def}} & (s, a, s') \in \rightarrow \\
s \xrightarrow{a_1 \cdot \ldots \cdot a_n} s' & =_{\text{def}} & \exists\, s_0, \ldots, s_n \in S \;.\; s = s_0 \xrightarrow{a_1} \cdots \xrightarrow{a_n} s_n = s' \\
s \xrightarrow{a} & =_{\text{def}} & \exists\, s'' \in S \;.\; s \xrightarrow{a} s'' \\
s \not\xrightarrow{a} & =_{\text{def}} & \nexists\, s'' \in S \;.\; s \xrightarrow{a} s'' \\
s \xRightarrow{\epsilon} s' & =_{\text{def}} & s = s' \text{ or } s \xrightarrow{\tau \cdot \ldots \cdot \tau} s' \\
s \xRightarrow{b} s' & =_{\text{def}} & \exists\, s_0, s_1 \in S \;.\; s \xRightarrow{\epsilon} s_0 \xrightarrow{b} s_1 \xRightarrow{\epsilon} s' \\
s \xRightarrow{b_1 \cdot \ldots \cdot b_n} s' & =_{\text{def}} & \exists\, s_0, \ldots, s_n \in S \;.\; s = s_0 \xRightarrow{b_1} \cdots \xRightarrow{b_n} s_n = s' \\
s \xRightarrow{\sigma} & =_{\text{def}} & \exists\, s'' \in S \;.\; s \xRightarrow{\sigma} s'' \\
s \not\xRightarrow{\sigma} & =_{\text{def}} & \nexists\, s'' \in S \;.\; s \xRightarrow{\sigma} s''
\end{array}$$

If $s \xrightarrow{a}$ for a $s \in S$ and $a \in L^\tau$, we say that a is *enabled* in s. We use $L(s)$ to denote the set of all actions $a \in L^\tau$ that are enabled in the state $s \in S$, i.e., $L(s) = \{\, a \in L^\tau \mid s \xrightarrow{a} \,\}$.

Example 2.4. Consider the LTS $\mathcal{A}$ in Figure 2.1a. The following statements all apply to $\mathcal{A}$:

$$\begin{array}{llll}
s_3 \xrightarrow{b} s_5, & s_0 \xrightarrow{a\,\tau\,b} s_5, & s_4 \xrightarrow{c}, & s_3 \not\xrightarrow{c}, \\
s_1 \xRightarrow{\epsilon} s_3, & s_0 \xRightarrow{a\,b} s_5, & s_0 \xRightarrow{a\,c}, & s_0 \not\xRightarrow{a\,b\,c}
\end{array}$$

Furthermore, $L(s_0) = \{\, a \,\}$ and $L(s_5) = \emptyset$. □

We will use the following language notations to denote various aspects of LTSs and their behaviour.

Definition 2.5 (Language notations)**.** Let $\mathcal{A} = \langle\, S, S^0, L, \rightarrow \,\rangle$ be an LTS, then:

- A *path* in $\mathcal{A}$ is an alternating sequence of states and actions that can be either finite or infinite. A finite path is a finite sequence $\pi = s_0\, a_1\, s_1\, a_2\, s_2\, \ldots\, s_n$ such that for all $1 \leq i \leq n$ we have $s_{i-1} \xrightarrow{a_i} s_i$ with $s_i \in S$, $a_i \in L$. An infinite path is an infinite sequence $\pi = s_0\, a_1\, s_1\, a_2\, s_2\, \ldots$ such that for all $i \geq 1$ we have $s_{i-1} \xrightarrow{a_i} s_i$ with $s_i \in S$, $a_i \in L$. A path $\pi = s_0\, a_1\, s_1\, a_2\, s_2\, \ldots$ is called *cyclic* if $s_i = s_j$ for some $i \neq j$.

- The set of all paths in $\mathcal{A}$ is denoted *paths*$(\mathcal{A})$. The path operator *first* yields the first state on a given path, e.g., for $\pi = s_0\, a_1\, s_1$ we have *first*$(\pi) = s_0$. The path operator *states* yields the set of states that occur on a given path π, i.e., *states*$(\pi) = \pi \upharpoonright S$. For example, for $\pi = s_0\, a_1\, s_1\, \tau\, s_0\, a_2\, s_2$ we have *states*$(\pi) = \{\, s_0, s_1, s_2 \,\}$.

- For an infinite path π, ω-*states*(π) denotes the set of states that occur infinitely often on that path, i.e., for a path $\pi = s_0\, a_1\, s_1\, \ldots$, we define ω-*states*$(\pi) = \{\, s \in \textit{states}(\pi) \mid \exists^{\infty} j \;.\; s_j = s \,\}$.

- The path operator *trace* yields the sequence of actions that is obtained by erasing all states and internal actions from a given path, i.e., *trace*$(\pi) = \pi \upharpoonright L$. Such a sequence of actions is called a *trace*. For example, for $\pi = s_0\, a_1\, s_1\, \tau\, s_2\, a_2\, s_3$ we have *trace*$(\pi) = a_1\, a_2$.

- For every $s \in S$, *traces*(s) denotes the set of all traces of $\mathcal{A}$ that correspond to paths that start in s, i.e., *traces*$(s) = \{\, \textit{trace}(\pi) \mid \pi \in \textit{paths}(\mathcal{A}) \;\wedge\; \textit{first}(\pi) = s \,\}$. The set of all traces that correspond to paths that start in one of the start states of $\mathcal{A}$ is denoted *traces*$(\mathcal{A}) = \bigcup_{s \in S^0} \textit{traces}(s)$. Two LTSs $\mathcal{B}$ and $\mathcal{C}$ are *trace equivalent*, denoted $\mathcal{B} \approx_{\text{tr}} \mathcal{C}$, if *traces*$(\mathcal{B}) = $ *traces*$(\mathcal{C})$.

- For a finite trace σ and state $s \in S$, $\mathit{reach}(s, \sigma)$ denotes the set of states in $\mathcal{A}$ that can be reached from s via σ, i.e., $\mathit{reach}(s, \sigma) = \{ s' \in S \mid s \xRightarrow{\sigma} s' \}$; for a set of states $S' \subseteq S$, $\mathit{reach}(S', \sigma)$ denotes the set of states that can be reached from a state in S', i.e., $\mathit{reach}(S', \sigma) = \bigcup_{s \in S'} \mathit{reach}(s, \sigma)$.

We add subscripts to these language notations to indicate the LTS they refer to, in case this is not clear from the context.

Example 2.6. First, consider again the LTS $\mathcal{A}$ in Figure 2.1a. Clearly, $s_0\, a\, s_1\, \tau\, s_3\, b\, s_5$ and $s_0\, a\, s_2\, \tau\, s_4\, c\, s_6$ are finite paths of $\mathcal{A}$. We have $\mathit{traces}(A) = \{ \epsilon, a, a\, b, a\, c \}$. Furthermore, we find that $\mathit{reach}(s_0, a) = \{ s_1, s_2, s_3, s_4 \}$ and $\mathit{reach}(\{ s_1, s_4 \}, \epsilon) = \{ s_1, s_3, s_4 \}$. Now, consider the LTS $\mathcal{B}$ in Figure 2.1c. Both $\pi_1 = s_0\, a\, s_1\, \tau\, s_1\, \tau\, s_1\, \ldots$ and $\pi_2 = s_1\, \tau\, s_2\, \tau\, s_3\, \tau\, s_1\, \tau\, s_2\, \ldots$ are infinite paths of $\mathcal{B}$. In this case, we have $\omega\text{-}\mathit{states}(\pi_1) = \{ s_1 \}$ and $\omega\text{-}\mathit{states}(\pi_2) = \{ s_1, s_2, s_3 \}$. □

A fundamental concept in automata theory is the notion of determinism.

Definition 2.7 (Determinism)**.** An LTS $\mathcal{A}$ is *deterministic* if the following two conditions hold:

1. for all $s, s' \in S$ and $a \in L^\tau$, if $s \xrightarrow{a} s'$, then $a \neq \tau$;
2. for all $s, s', s'' \in S$ and $a \in L$, if $s \xrightarrow{a} s'$ and $s \xrightarrow{a} s''$, then $s' = s''$.

Otherwise, $\mathcal{A}$ is *nondeterministic.*

Example 2.8. The LTS $\mathcal{A}$ in Figure 2.1a is nondeterministic, since both of the transitions $s_0 \xrightarrow{a} s_1$ and $s_0 \xrightarrow{a} s_2$ are enabled in s_0. Hence, if a is observed in state s_0, we do not know beforehand whether we end up in state s_1 or s_2. Furthermore, $\mathcal{A}$ contains several τ-transitions. □

When considering infinite paths, the notion of divergence (and convergence) is important.

Definition 2.9 (Divergence)**.** Let $\mathcal{A}$ be an LTS and let $\pi \in \mathit{paths}(\mathcal{A})$ be an infinite path in $\mathcal{A}$. The path π is *divergent* if it contains only internal transitions, i.e., $a_i = \tau$ for every action a_i on π. The set of divergent paths of $\mathcal{A}$ is denoted $\mathit{dpaths}(\mathcal{A})$. If $\mathcal{A}$ contains any divergent paths, then it is called divergent; otherwise, $\mathcal{A}$ is *convergent.*

Example 2.10. Consider the LTS $\mathcal{B}$ in Figure 2.1c. The infinite path $\pi = s_1\, \tau\, s_2\, \tau\, s_3\, \tau\, s_1\, \tau\, s_2\, \ldots$ of $\mathcal{B}$ is divergent, as it contains only internal transitions. Futhermore, since $|\mathit{states}(\pi)| = |\{ s_1, s_2, s_3 \}| = 3$, π is bounded. □

2.2.2 Operations

In this section, we take a look at some of the standard operations that can be applied to LTSs: determinisation, parallel composition and action hiding. It is a well-known fact from the literature that LTSs are closed under all three operations.

Each nondeterministic LTS can be transformed into a deterministic LTS [Sud06]; the latter is called the *determinisation* of the original LTS and is trace equivalent to it [BK08]. This operation is very useful, as it allows one to model a system as an LTS without paying attention to determinism; it can always be made deterministic afterwards.

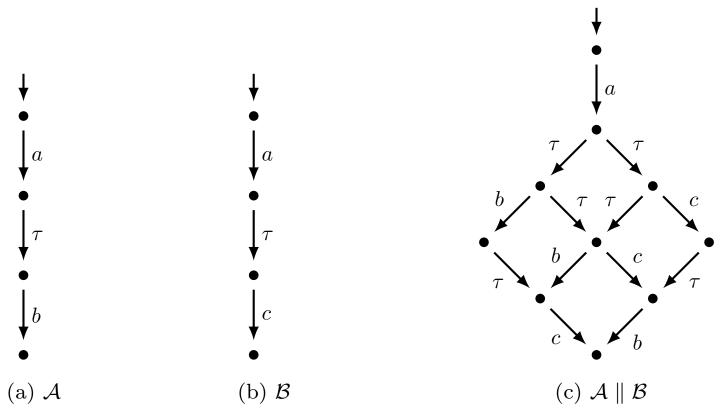


Figure 2.2: The LTSs $\mathcal{A}$ and $\mathcal{B}$, and their parallel composition $\mathcal{A} \parallel \mathcal{B}$.

Definition 2.11 (Determinisation of LTSs)**.** The *determinisation* of an LTS $\mathcal{A} = \langle\, S, S^0, L, \rightarrow_{\mathcal{A}} \rangle$ is the LTS $\mathit{det}(\mathcal{A}) = \langle\, S_{\mathrm{D}}, S^0_{\mathrm{D}}, L, \rightarrow_{\mathrm{D}} \rangle$, where S_{D}, S^0_{D} and $\rightarrow_{\mathrm{D}}$ are defined as follows:

$$\begin{array}{rcl} S_{\mathrm{D}} & = & \wp(S) \setminus \emptyset \\ S^0_{\mathrm{D}} & = & \{\, S^0 \,\} \\ \rightarrow_{\mathrm{D}} & = & \{\, (U, a, V) \in S_{\mathrm{D}} \times L \times S_{\mathrm{D}} \quad \mid \quad V = \mathit{reach}_{\mathcal{A}}(U, a) \ \wedge \ V \neq \emptyset \,\} \end{array}$$

Example 2.12. The determinisation of the nondeterministic LTS $\mathcal{A}$ in Figure 2.1a is shown in Figure 2.1b. In this case, the four individual states s_1, s_2, s_3 and s_4 of $\mathcal{A}$ are condensed into one single composite state in $\mathit{det}(\mathcal{A})$, since $\mathit{reach}(s_0, a) = \{\, s_1, s_2, s_3, s_4 \,\}$. Note also that indeed $\mathcal{A} \approx_{\mathrm{tr}} \mathit{det}(\mathcal{A})$. □

Next, we introduce the parallel composition operator. This operator is fundamental in modelling frameworks for component-based design. It allows one to build complex system models from smaller ones, thus breaking up the specification of a system into manageable pieces. Parallel composed LTSs synchronise on shared actions, and can execute the internal action τ and non-shared actions indepedently from each other.

Definition 2.13 (Parallel composition of LTSs)**.** Given two LTSs $\mathcal{A} = \langle\, S_{\mathcal{A}}, S^0_{\mathcal{A}}, L_{\mathcal{A}}, \rightarrow_{\mathcal{A}} \rangle$ and $\mathcal{B} = \langle\, S_{\mathcal{B}}, S^0_{\mathcal{B}}, L_{\mathcal{B}}, \rightarrow_{\mathcal{B}} \rangle$, the *parallel composition* of $\mathcal{A}$ and $\mathcal{B}$ is the LTS $\mathcal{A} \parallel \mathcal{B} = \langle\, S_{\mathcal{A}\parallel\mathcal{B}}, S^0_{\mathcal{A}\parallel\mathcal{B}}, L_{\mathcal{A}\parallel\mathcal{B}}, \rightarrow_{\mathcal{A}\parallel\mathcal{B}} \rangle$, where $S_{\mathcal{A}\parallel\mathcal{B}}$, $S^0_{\mathcal{A}\parallel\mathcal{B}}$, $L_{\mathcal{A}\parallel\mathcal{B}}$ and $\rightarrow_{\mathcal{A}\parallel\mathcal{B}}$ are defined as follows:

$$\begin{array}{rcl} S_{\mathcal{A}\parallel\mathcal{B}} & = & S_{\mathcal{A}} \times S_{\mathcal{B}} \\ S^0_{\mathcal{A}\parallel\mathcal{B}} & = & S^0_{\mathcal{A}} \times S^0_{\mathcal{B}} \\ L_{\mathcal{A}\parallel\mathcal{B}} & = & L_{\mathcal{A}} \cup L_{\mathcal{B}} \\ \rightarrow_{\mathcal{A}\parallel\mathcal{B}} & = & \{\, ((s,t), a, (s',t')) \in S_{\mathcal{A}\parallel\mathcal{B}} \times (L_{\mathcal{A}} \cap L_{\mathcal{B}}) \times S_{\mathcal{A}\parallel\mathcal{B}} \ \mid \ s \xrightarrow{a}_{\mathcal{A}} s' \ \wedge \ t \xrightarrow{a}_{\mathcal{B}} t' \,\} \\ & \cup & \{\, ((s,t), a, (s',t)) \in S_{\mathcal{A}\parallel\mathcal{B}} \times ((L_{\mathcal{A}} \setminus L_{\mathcal{B}}) \cup \{\, \tau \,\}) \times S_{\mathcal{A}\parallel\mathcal{B}} \ \mid \ s \xrightarrow{a}_{\mathcal{A}} s' \,\} \\ & \cup & \{\, ((s,t), a, (s,t')) \in S_{\mathcal{A}\parallel\mathcal{B}} \times ((L_{\mathcal{B}} \setminus L_{\mathcal{A}}) \cup \{\, \tau \,\}) \times S_{\mathcal{A}\parallel\mathcal{B}} \ \mid \ t \xrightarrow{a}_{\mathcal{B}} t' \,\} \end{array}$$

The first clause of the definition of $\rightarrow_{\mathcal{A}\parallel\mathcal{B}}$ ensures that parallel composed LTSs synchronise on shared actions (except the internal action τ). The next two clauses enable them to perform non-shared actions (including the internal action τ) independently from each other.

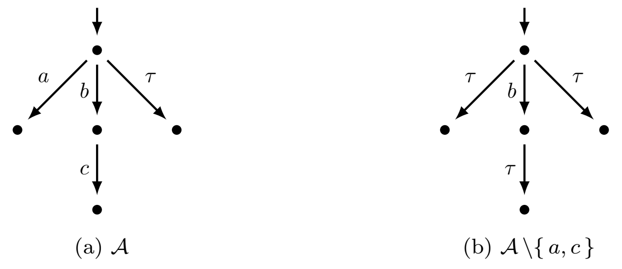


Figure 2.3: The LTSs $\mathcal{A}$ and $\mathcal{A}\backslash\{\,a,c\,\}$.

Example 2.14. Figure 2.2 shows two LTSs $\mathcal{A}$ and $\mathcal{B}$, and their parallel composition $\mathcal{A} \parallel \mathcal{B}$. We have assumed $L_{\mathcal{A}} = \{\,a,b,d\,\}$ and $L_{\mathcal{B}} = \{\,a,c,d\,\}$, hence $L_{\mathcal{A}\parallel\mathcal{B}} = \{\,a,b,c,d\,\}$. Since a and d are shared actions, the parallel composition $\mathcal{A} \parallel \mathcal{B}$ can only execute these actions when both component LTSs are able to, i.e., $\mathcal{A} \parallel \mathcal{B}$ synchronises on the a and d actions. The other actions, including the internal action τ, can be executed independently. □

Finally, it is often useful to hide certain actions of a LTS, thereby treating them as internal actions. For example, when parallel composing two LTSs, some actions are only used for synchronisation; after parallel composition, they are not needed anymore.

Definition 2.15 (Action hiding in LTSs)**.** Let $\mathcal{A} = \langle\, S, S^0, L, \rightarrow_{\mathcal{A}} \,\rangle$ be an LTS and $H \subseteq L^{\mathrm{O}}$ a set of output labels, then one can *hide* H in $\mathcal{A}$ to obtain the LTS $\mathcal{A} \setminus H = \langle\, S, S^0, L_H, \rightarrow_H \,\rangle$, where L_H and $\rightarrow_H$ are defined as follows:

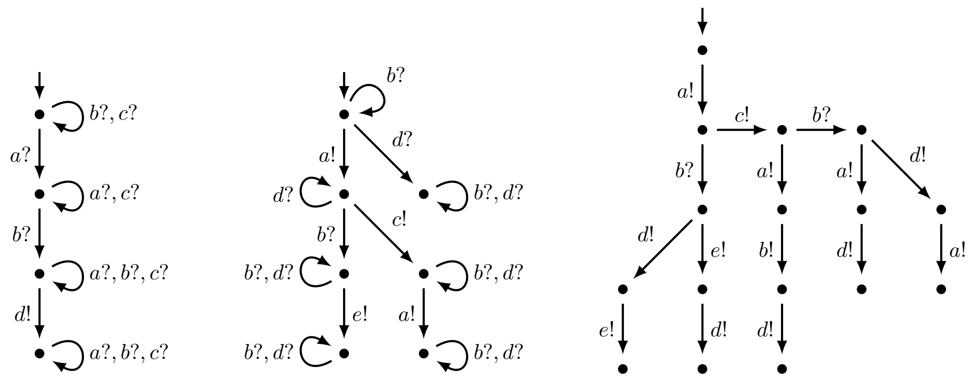
$$\begin{array}{rcll} L_H & = & L \setminus H & \\ \rightarrow_H & = & \{\,(s,a,s') \in \rightarrow_{\mathcal{A}} & \mid\ a \notin H\,\} \\ & \cup & \{\,(s,\tau,s') \in S \times \{\,\tau\,\} \times S & \mid\ \exists\, a \in H \,.\, (s,a,s') \in \rightarrow_{\mathcal{A}}\,\} \end{array}$$

Hence, the hidden actions are removed from the set of actions, and all transitions for those actions become τ-transitions.

Example 2.16. Consider the LTS $\mathcal{A}$ in Figure 2.3a and assume $L_{\mathcal{A}} = \{\,a,b,c\,\}$. After hiding the actions a and c, the resulting IOTS is $\mathcal{A}\backslash\{\,a,c\,\}$, which is shown in Figure 2.3b. We have $L_{\mathcal{A}\backslash\{\,a,c\,\}} = \{\,b\,\}$. □

2.3 Input-Output Transition Systems

Often, in particular in the context of testing, it is desirable to distinguish between actions initiated by the environment (inputs), and actions initiated by the system itself (outputs). To this end, Input-Output Transition Systems (IOTSs) [Tre96a, Tre96b] were developed, which are an extension of regular LTSs. In the context of IOTSs, we distinguish between input actions, output actions, and the internal action τ. Input actions are supplied by the environment to the system, and in response the system can generate outputs, which may also be generated autonomously by the system. All outputs can be observed by the environment, except for the internal action, which is an unobservable output. We call the set of all outputs, together with the internal action, the locally controlled actions of the system.



(a) $\mathcal{A}$ (b) $\mathcal{A}$ (c) $\mathcal{A} \parallel \mathcal{B}$

Figure 2.4: The IOTSs $\mathcal{A}$ and $\mathcal{B}$, and their parallel composition $\mathcal{A} \parallel \mathcal{B}$. Note that we have left out some of the b-labelled self-loops from the visualisation of $\mathcal{A} \parallel \mathcal{B}$ to reduce clutter.

2.3.1 The IOTS Model

Definition 2.17 (Input-Output Transition Systems)**.** An *Input-Output Transition System* (IOTS) is a tuple $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow \rangle$, such that:

- S is a non-empty set of states;
- $S^0 \subseteq S$ is a non-empty set of initial states;
- L^{I} and L^{O} are disjoint sets of input and output labels, respectively;
- $\rightarrow \subseteq S \times (L^{\mathrm{I}} \cup L^{\mathrm{O}} \cup \{ \tau \}) \times S$ is the transition relation.

We define $L = L^{\mathrm{I}} \cup L^{\mathrm{O}}$ and $L^\tau = L \cup \{ \tau \}$, and require $\tau \notin L$.

Remark 2.18. We often suffix a question mark (?) to input labels and an exclamation mark (!) to output labels, to help distinguishing the two types. These are, however, not part of the label.

The notations introduced in Definition 2.3 and Definition 2.5 for LTSs also apply to IOTSs. Compared to regular LTSs, IOTSs partition the set of labels into disjoint sets of input labels and output labels. Furthermore, we require every IOTS to be *input-enabled*.

Definition 2.19 (Input-enabledness)**.** An IOTS $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow \rangle$ is *input-enabled* if $s \xrightarrow{a}$ for all $s \in S$ and every $a \in L^{\mathrm{I}}$, i.e., $\mathcal{A}$ can accept any input in any state.

Example 2.20. Figure 2.4a shows an IOTS $\mathcal{A}$ with $L^{\mathrm{I}}_{\mathcal{A}} = \{ a, b, c \}$ and $L^{\mathrm{O}}_{\mathcal{A}} = \{ d \}$. Note that $\mathcal{A}$ is indeed input-enabled.

From now on, we assume that all given IOTSs are input-enabled, unless explicitly stated otherwise.

By requiring IOTSs to be input-enabled, any input initiated by the environment is never refused by the system. For deterministic systems this requirement can easily be fulfilled by adding a sink state which has self-loops for all possible actions, and adding transitions for the missing inputs to that sink state (so-called *demonic completion* [DNS95, vRT04]). For nondeterministic systems a solution is provided in [BS08].

2.3.2 Operations

Similar to LTSs, IOTSs support the operations of determinisation, parallel composition and action hiding. Determinisation for IOTSs is exactly the same as for LTSs. Parallel composition of IOTSs is different, since rather than synchronising on all shared actions like LTSs, parallel composed IOTSs synchronise on shared inputs and complementary input-output pairs, and cannot have shared outputs [DNS95]. Furthermore, when two inputs synchronise the result is an input transition in the composite automaton, and when a complementary input and output synchronise, the result is an output transition.

Definition 2.21 (Parallel composition of IOTSs)**.** Given IOTSs $\mathcal{A} = \langle\, S_\mathcal{A}, S^0_\mathcal{A}, L^\mathrm{I}_\mathcal{A}, L^\mathrm{O}_\mathcal{A}, \rightarrow_\mathcal{A} \rangle$ and $\mathcal{B} = \langle\, S_\mathcal{B}, S^0_\mathcal{B}, L^\mathrm{I}_\mathcal{B}, L^\mathrm{O}_\mathcal{B}, \rightarrow_\mathcal{B} \rangle$ such that $L^\mathrm{O}_\mathcal{A} \cap L^\mathrm{O}_\mathcal{B} = \emptyset$, the *parallel composition* of $\mathcal{A}$ and $\mathcal{B}$ is the IOTS $\mathcal{A} \parallel \mathcal{B} = \langle\, S_{\mathcal{A}\parallel\mathcal{B}}, S^0_{\mathcal{A}\parallel\mathcal{B}}, L^\mathrm{I}_{\mathcal{A}\parallel\mathcal{B}}, L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}}, \rightarrow_{\mathcal{A}\parallel\mathcal{B}} \rangle$, where $S_{\mathcal{A}\parallel\mathcal{B}}$, $S^0_{\mathcal{A}\parallel\mathcal{B}}$, $L^\mathrm{I}_{\mathcal{A}\parallel\mathcal{B}}$, $L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}}$ and $\rightarrow_{\mathcal{A}\parallel\mathcal{B}}$ are defined as follows:

$$\begin{array}{rcl}
S_{\mathcal{A}\parallel\mathcal{B}} & = & S_\mathcal{A} \times S_\mathcal{B} \\
S^0_{\mathcal{A}\parallel\mathcal{B}} & = & S^0_\mathcal{A} \times S^0_\mathcal{B} \\
L^\mathrm{I}_{\mathcal{A}\parallel\mathcal{B}} & = & (L^\mathrm{I}_\mathcal{A} \cup L^\mathrm{I}_\mathcal{B}) \setminus (L^\mathrm{O}_\mathcal{A} \cup L^\mathrm{O}_\mathcal{B}) \\
L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}} & = & L^\mathrm{O}_\mathcal{A} \cup L^\mathrm{O}_\mathcal{B} \\
\rightarrow_{\mathcal{A}\parallel\mathcal{B}} & = & \{\, ((s,t),a,(s',t')) \in S_{\mathcal{A}\parallel\mathcal{B}} \times (L_\mathcal{A} \cap L_\mathcal{B}) \times S_{\mathcal{A}\parallel\mathcal{B}} \;\mid\; s \xrightarrow{a}_\mathcal{A} s' \,\wedge\, t \xrightarrow{a}_\mathcal{B} t' \,\} \\
 & \cup & \{\, ((s,t),a,(s',t)) \in S_{\mathcal{A}\parallel\mathcal{B}} \times ((L_\mathcal{A} \setminus L_\mathcal{B}) \cup \{\,\tau\,\}) \times S_{\mathcal{A}\parallel\mathcal{B}} \;\mid\; s \xrightarrow{a}_\mathcal{A} s' \,\} \\
 & \cup & \{\, ((s,t),a,(s,t')) \in S_{\mathcal{A}\parallel\mathcal{B}} \times ((L_\mathcal{B} \setminus L_\mathcal{A}) \cup \{\,\tau\,\}) \times S_{\mathcal{A}\parallel\mathcal{B}} \;\mid\; t \xrightarrow{a}_\mathcal{B} t' \,\}
\end{array}$$

We have $L_{\mathcal{A}\parallel\mathcal{B}} = L^\mathrm{I}_{\mathcal{A}\parallel\mathcal{B}} \cup L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}} = L_\mathcal{A} \cup L_\mathcal{B}$.

Example 2.22. Figure 2.4 shows two IOTSs $\mathcal{A}$ and $\mathcal{B}$, and their parallel composition $\mathcal{A} \parallel \mathcal{B}$. We have assumed that $L^\mathrm{I}_\mathcal{A} = \{\, a, b, c \,\}$, $L^\mathrm{O}_\mathcal{A} = \{\, d \,\}$, $L^\mathrm{I}_\mathcal{B} = \{\, b, d \,\}$, and $L^\mathrm{O}_\mathcal{B} = \{\, a, c, e \,\}$. Note that indeed $L^\mathrm{O}_\mathcal{A} \cap L^\mathrm{O}_\mathcal{B} = \emptyset$, as required; hence $L^\mathrm{I}_{\mathcal{A}\parallel\mathcal{B}} = \{\, b \,\}$ and $L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}} = \{\, a, c, d, e \,\}$. □

Action hiding is exactly the same for IOTSs as for LTSs, except that only output actions can be hidden.

Definition 2.23 (Action hiding in IOTSs)**.** Let $\mathcal{A} = \langle\, S, S^0, L^\mathrm{I}, L^\mathrm{O}, \rightarrow_\mathcal{A} \rangle$ be an IOTS and $H \subseteq L^\mathrm{O}$ a set of output labels, then one can *hide* H in $\mathcal{A}$ to obtain the IOTS $\mathcal{A} \setminus H = \langle\, S, S^0, L^\mathrm{I}, L^\mathrm{O}_H, \rightarrow_H \rangle$, where L^O_H and $\rightarrow_H$ are defined as follows:

$$\begin{array}{rcll}
L^\mathrm{O}_H & = & L^\mathrm{O} \setminus H & \\
\rightarrow_H & = & \{\, (s,a,s') \in \rightarrow_\mathcal{A} & \mid\; a \notin H \,\} \\
 & \cup & \{\, (s,\tau,s') \in S \times \{\,\tau\,\} \times S & \mid\; \exists\, a \in H \,.\, (s,a,s') \in \rightarrow_\mathcal{A} \,\}
\end{array}$$

Like LTSs, IOTSs are closed under the operations of determinisation, parallel composition, and action hiding.

2.4 Input-Output Automata

The IOTSs introduced in the previous section form a subclass of a more general class of automata called Input-Output Automata (IOAs) [LT87, LT89], which are another type of LTSs. Just like IOTSs, IOAs distinguish between input actions generated by the environment, and output actions generated by the system itself. However, IOAs can have multiple internal actions (rather than just τ). Furthermore, with each IOA a partition of the locally controlled actions (the combined set of output and internal actions) is associated. This partition is used to formalise the notion of fair executions.

2.4.1 The IOA Model

Definition 2.24 (Input-Output Automata)**.** An *Input-Output Automaton* (IOA) is a tuple $\mathcal{A} = \langle\, S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow \rangle$, such that:

- S is a non-empty set of states;
- $S^0 \subseteq S$ is a non-empty set of initial states;
- L^{I}, L^{O} and L^{H} are pairwise disjoint sets of input, output, and internal labels, respectively;
- P is a partition of $L^{\mathrm{O}} \cup L^{\mathrm{H}}$, i.e., a partition of the locally controlled actions, and is called the task partition;
- $\rightarrow\ \subseteq S \times (L^{\mathrm{I}} \cup L^{\mathrm{O}} \cup L^{\mathrm{H}}) \times S$ is the transition relation.

We define $L = L^{\mathrm{I}} \cup L^{\mathrm{O}} \cup L^{\mathrm{H}}$.

The intuition behind the task partition P is that each element of P represents the set of locally controlled actions under the control of a particular subcomponent of the whole system. Hence, an element of P may contain both outputs and internal transitions. The task partition P plays an important role when it comes to the notion of fairness, as will be shown later on.

Similar to IOTSs, we require every IOA to be input-enabled. Clearly, for every IOTS $\mathcal{A} = \langle\, S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow \rangle$ there exists a trace-equivalent IOA $\mathcal{A}' = \langle\, S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow \rangle$, with $L^{\mathrm{H}} = \{\, \tau \,\}$ and $P = \{\, L^{\mathrm{O}} \cup \{\, \tau \,\}\,\}$.

Example 2.25. Let $\mathcal{A}$ be an IOA with $L^{\mathrm{O}}_{\mathcal{A}} = \{\, a, b \,\}$ and $L^{\mathrm{H}}_{\mathcal{A}} = \{\, c, d \,\}$. An example of a partition of the locally controlled actions of $\mathcal{A}$ would be $P_{\mathcal{A}} = \{\, \{\, a, c \,\}, \{\, b, d \,\}\,\}$. In this case, $\mathcal{A}$ is assumed to consist of two independent subcomponents, which control the sets of actions $\{\, a, c \,\}$ and $\{\, b, d \,\}$, respectively. □

Remark 2.26. IOAs are visualised in the same manner as IOTSs, with a question mark (?) for an input, and an exclamation mark (!) for an output. A label without a suffix is assumed to be an internal label.

The notations introduced in Definition 2.3 and Definition 2.5 for LTSs also apply to IOAs. However, since IOAs can have multiple internal labels, the transitional notation $\overset{\epsilon}{\Longrightarrow}$ is more general for IOAs than for LTSs and IOTSs, which only have one internal label (τ).

Definition 2.27 (Transitional notations for IOAs)**.** Let $\mathcal{A}$ be an IOA with $s, s' \in S$, then:

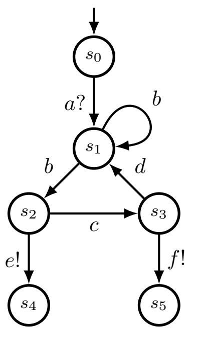
$$s \overset{\epsilon}{\Longrightarrow} s' \quad =_{\mathrm{def}} \quad s = s' \text{ or } \exists\, a_1, \ldots, a_n \in L^{\mathrm{H}}_{\mathcal{A}} \,.\, s \xrightarrow{a_1 \cdot \ldots \cdot a_n} s'$$

The other transitional notations use this more general definition of $\overset{\epsilon}{\Longrightarrow}$ where applicable.

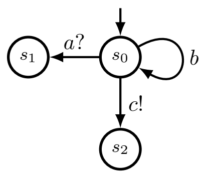
Similarly, the definitions of determinism and divergence are more general for IOAs than for LTSs and IOTSs.

Definition 2.28 (Determinism in IOAs)**.** An IOA $\mathcal{A}$ is *deterministic* if the following two conditions hold:

1. for all $s, s' \in S$ and $a \in L$, if $s \xrightarrow{a} s'$, then $a \notin L^{\mathrm{H}}_{\mathcal{A}}$;
2. for all $s, s', s'' \in S$ and $a \in L$, if $s \xrightarrow{a} s'$ and $s \xrightarrow{a} s''$, then $s' = s''$.



(a) $\mathcal{A}$



(b) $\mathcal{B}$

Figure 2.5: The IOAs $\mathcal{A}$ and $\mathcal{B}$. Note that suffixless labels indicate internal actions

Example 2.29. The IOA $\mathcal{A}$ in Figure 2.5a is nondeterministic, as it contains multiple transitions labelled with internal actions. □

Definition 2.30 (Divergent path in IOAs)**.** Let $\mathcal{A}$ be an IOA and $\pi \in \mathit{paths}(\mathcal{A})$ an infinite path in $\mathcal{A}$. The path π is *divergent* if it contains only transitions labelled with internal actions, i.e., $a_i \in L^{\mathrm{H}}_{\mathcal{A}}$ for every action a_i on π.

Example 2.31. Consider the IOA $\mathcal{A}$ in Figure 2.5a with $L^{\mathrm{H}}_{\mathcal{A}} = \{\, b, c, d \,\}$. Clearly, the infinite paths $s_1\, b\, s_1\, b\, s_1\, \ldots$ and $s_1\, b\, s_2\, c\, s_3\, d\, s_1\, b\, s_2\, \ldots$ are both divergent. □

The notion of fairness plays an important role in IOAs (and Divergent Quiescent Transition Systems, as will be explained in Chapter 4). The following definition of a fair path improves the notion of fair executions given in [LT87, LT89, DNS95], assuming that if an action from an element of P is infinitely often enabled in an infinite path, then an action from that same element of P must be executed infinitely often in that path.

Definition 2.32 (Fair path)**.** Let $\mathcal{A} = \langle\, S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow \,\rangle$ be an IOA and $\pi = s_0\, a_1\, s_1\, a_2\, s_2\, \ldots$ a path of $\mathcal{A}$. If π is an infinite path, then π is *fair* if, for every $A \in P$ such that $\exists^\infty j \;.\; L(s_j) \cap A \neq \emptyset$, we have $\exists^\infty j \;.\; a_j \in A$. Thus, if actions from A are infinitely often enabled in the states of path π, then actions from A are infinitely often executed in the path π. If π is a finite path, then π is fair by default. The set of all fair paths of an IOA $\mathcal{A}$ is denoted $\mathit{fpaths}(\mathcal{A})$.

Hence, under this notion of fairness, each subcomponent of the system (represented by an element of the set P) that is infinitely often given the chance to execute some of its actions, will infinitely often execute some of its actions.

Example 2.33. Consider the IOA $\mathcal{B}$ in Figure 2.5b. Clearly, $\pi = s_0\, b\, s_0\, b\, s_0\, \ldots$ is an infinite path of $\mathcal{A}$. Now assume that $P_{\mathcal{A}} = \{\, \{\, b \,\}, \{\, c \,\} \,\}$; i.e., the internal b-action and the c-output are controlled by two independent subcomponents. In this case, the path π would not be fair: the c-output, which belongs to a different partition than the internal b-action, is infinitely often enabled, but is never executed. The path π is only fair if $P_{\mathcal{A}} = \{\, \{\, b, c \,\} \,\}$. Hence, if $P_{\mathcal{A}} = \{\, \{\, b \,\}, \{\, c \,\} \,\}$, then $\mathit{fpaths}(\mathcal{A}) = \emptyset$, but if $P_{\mathcal{A}} = \{\, \{\, b, c \,\} \,\}$, then $\mathit{fpaths}(\mathcal{A}) = \{\, \pi \,\}$. □

Intuitively, unfair (infinite) paths do not correspond to realistic executions of the system, since this would imply that subcomponents of a composite IOA would be idle indefinitely, even though they are able to execute locally controlled actions. Hence, unfair paths are considered not to occur.

2.4.2 Operations

The determinisation of IOAs proceeds exactly the same as for LTSs and IOTSs. The operation of parallel compositon is also applicable to IOAs. However, since IOAs can have multiple internal actions and also have an associated partition of the locally controlled actions, we need to impose some additional constraints to ensure that the component IOAs in a parallel composition do not use internal actions to communicate, and that every locally controlled action of the parallel composition is under the control of at most one component IOA [LT87, LT89]. Two IOAs that satisfy these constraints are said to be compatible.

Definition 2.34 (IOA compatibility)**.** Two IOAs $\mathcal{A} = \langle\, S_{\mathcal{A}}, S^0_{\mathcal{A}}, L^{\mathrm{I}}_{\mathcal{A}}, L^{\mathrm{O}}_{\mathcal{A}}, L^{\mathrm{H}}_{\mathcal{A}}, P_{\mathcal{A}}, \rightarrow_{\mathcal{A}} \rangle$ and $\mathcal{B} = \langle\, S_{\mathcal{B}}, S^0_{\mathcal{B}}, L^{\mathrm{I}}_{\mathcal{B}}, L^{\mathrm{O}}_{\mathcal{B}}, L^{\mathrm{H}}_{\mathcal{B}}, P_{\mathcal{B}}, \rightarrow_{\mathcal{B}} \rangle$ are *compatible* if their sets of ouput actions are disjoint, and no internal action of either appears as an input, output or internal action of the other, i.e., $\mathcal{A}$ and $\mathcal{B}$ are compatible if $L^{\mathrm{O}}_{\mathcal{A}} \cap L^{\mathrm{O}}_{\mathcal{B}} = \emptyset$, $L^{\mathrm{H}}_{\mathcal{A}} \cap L_{\mathcal{B}} = \emptyset$, and $L^{\mathrm{H}}_{\mathcal{B}} \cap L_{\mathcal{A}} = \emptyset$.

Two compatible IOAs can be parallel composed in a similar way as IOTSs, as the next definition shows.

Definition 2.35 (Parallel composition of IOAs)**.** Let $\mathcal{A} = \langle\, S_{\mathcal{A}}, S^0_{\mathcal{A}}, L^{\mathrm{I}}_{\mathcal{A}}, L^{\mathrm{O}}_{\mathcal{A}}, L^{\mathrm{H}}_{\mathcal{A}}, P_{\mathcal{A}}, \rightarrow_{\mathcal{A}} \rangle$ and $\mathcal{B} = \langle\, S_{\mathcal{B}}, S^0_{\mathcal{B}}, L^{\mathrm{I}}_{\mathcal{B}}, L^{\mathrm{O}}_{\mathcal{B}}, L^{\mathrm{H}}_{\mathcal{B}}, P_{\mathcal{B}}, \rightarrow_{\mathcal{B}} \rangle$ be two compatible IOAs. The *parallel composition* of $\mathcal{A}$ and $\mathcal{B}$ is the IOA $\mathcal{A} \parallel \mathcal{B} = \langle\, S_{\mathcal{A}\parallel\mathcal{B}}, S^0_{\mathcal{A}\parallel\mathcal{B}}, L^{\mathrm{I}}_{\mathcal{A}\parallel\mathcal{B}}, L^{\mathrm{O}}_{\mathcal{A}\parallel\mathcal{B}}, L^{\mathrm{H}}_{\mathcal{A}\parallel\mathcal{B}}, P_{\mathcal{A}\parallel\mathcal{B}}, \rightarrow_{\mathcal{A}\parallel\mathcal{B}} \rangle$, where $S_{\mathcal{A}\parallel\mathcal{B}}$, $S^0_{\mathcal{A}\parallel\mathcal{B}}$, $L^{\mathrm{I}}_{\mathcal{A}\parallel\mathcal{B}}$, $L^{\mathrm{O}}_{\mathcal{A}\parallel\mathcal{B}}$, $L^{\mathrm{H}}_{\mathcal{A}\parallel\mathcal{B}}$, $P_{\mathcal{A}\parallel\mathcal{B}}$ and $\rightarrow_{\mathcal{A}\parallel\mathcal{B}}$ are defined as follows:

$$\begin{array}{lcll}
S_{\mathcal{A}\parallel\mathcal{B}} & = & S_{\mathcal{A}} \times S_{\mathcal{B}} & \\
S^0_{\mathcal{A}\parallel\mathcal{B}} & = & S^0_{\mathcal{A}} \times S^0_{\mathcal{B}} & \\
L^{\mathrm{I}}_{\mathcal{A}\parallel\mathcal{B}} & = & (L^{\mathrm{I}}_{\mathcal{A}} \cup L^{\mathrm{I}}_{\mathcal{B}}) \setminus (L^{\mathrm{O}}_{\mathcal{A}} \cup L^{\mathrm{O}}_{\mathcal{B}}) & \\
L^{\mathrm{O}}_{\mathcal{A}\parallel\mathcal{B}} & = & L^{\mathrm{O}}_{\mathcal{A}} \cup L^{\mathrm{O}}_{\mathcal{B}} & \\
L^{\mathrm{H}}_{\mathcal{A}\parallel\mathcal{B}} & = & L^{\mathrm{H}}_{\mathcal{A}} \cup L^{\mathrm{H}}_{\mathcal{B}} & \\
P_{\mathcal{A}\parallel\mathcal{B}} & = & P_{\mathcal{A}} \cup P_{\mathcal{B}} & \\
\rightarrow_{\mathcal{A}\parallel\mathcal{B}} & = & \{\, ((s,t),a,(s',t')) \in S_{\mathcal{A}\parallel\mathcal{B}} \times (L_{\mathcal{A}} \cap L_{\mathcal{B}}) \times S_{\mathcal{A}\parallel\mathcal{B}} & \mid\ s \xrightarrow{a}_{\mathcal{A}} s' \ \wedge\ t \xrightarrow{a}_{\mathcal{B}} t' \,\} \\
& \cup & \{\, ((s,t),a,(s',t)) \in S_{\mathcal{A}\parallel\mathcal{B}} \times (L_{\mathcal{A}} \setminus L_{\mathcal{B}}) \times S_{\mathcal{A}\parallel\mathcal{B}} & \mid\ s \xrightarrow{a}_{\mathcal{A}} s' \,\} \\
& \cup & \{\, ((s,t),a,(s,t')) \in S_{\mathcal{A}\parallel\mathcal{B}} \times (L_{\mathcal{B}} \setminus L_{\mathcal{A}}) \times S_{\mathcal{A}\parallel\mathcal{B}} & \mid\ t \xrightarrow{a}_{\mathcal{B}} t' \,\}
\end{array}$$

We have $L_{\mathcal{A}\parallel\mathcal{B}} = L^{\mathrm{I}}_{\mathcal{A}\parallel\mathcal{B}} \cup L^{\mathrm{O}}_{\mathcal{A}\parallel\mathcal{B}} \cup L^{\mathrm{H}}_{\mathcal{A}\parallel\mathcal{B}} = L_{\mathcal{A}} \cup L_{\mathcal{B}}$.

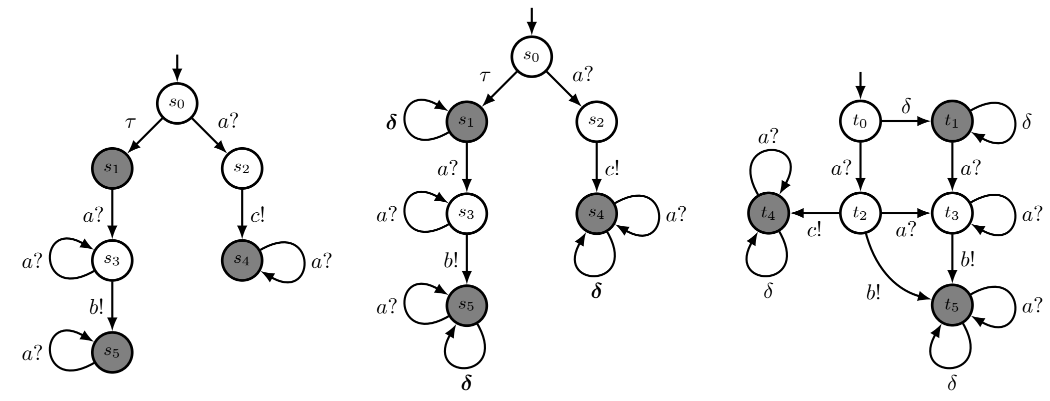
Hence, the partition of locally controlled actions for the parallel composition is simply the union of the partitions of the locally controlled actions of the component IOAs. The constraints imposed by the compatibility requirement ensure that this is indeed a valid partition.

The hiding of actions in an IOA is rather straightforward.

Definition 2.36 (Action hiding in IOAs)**.** Given an IOA $\mathcal{A} = \langle\, S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow \rangle$ and a set of outputs $H \subseteq L^{\mathrm{O}}$, *hiding H in $\mathcal{A}$* yields the IOA $\mathcal{A} \setminus H = \langle\, S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}_H, L^{\mathrm{H}}_H, P, \rightarrow \rangle$, where $L^{\mathrm{O}}_H = L^{\mathrm{O}} \setminus H$ and $L^{\mathrm{H}}_H = L^{\mathrm{H}} \cup H$.

Thus, the hiding operation simply removes the output labels that are to be hidden from the set of outputs, and adds them to the set of internal labels.

Like LTSs and IOTSs, IOAs are closed under the operations of determinisation, parallel composition, and action hiding [Tut87].



(a) $\mathcal{A}$ (b) $\mathcal{A}'$ (c) $\mathcal{S}$

Figure 2.6: Visual representation of the construction of the SA $\mathcal{S}$ that corresponds to the IOTS $\mathcal{A}$, alongside the intermediate automaton $\mathcal{A}'$.

2.5 Suspension Automata

As discussed in Section 2.3, Input-Output Transition Systems (IOTSs) can be used to model the reactive behaviour of a system in terms of inputs and outputs, but cannot explicitly express the observation of the absence of outputs, also called quiescence [Vaa91, Seg97]. For this, the so-called Suspension Automata (SAs) [Tre96a, Tre96b], an extension of regular IOTSs, are used. These automata can be used to model all possible observations for a particular system, including quiescence, and can thus be thought of as 'observation automata'. In this section, we give a short overview of SAs.

First of all, a concept that plays an important role in SAs is the so-called *quiescent* state. A quiescent state from which the system cannot proceed autonomously, without inputs from the environment; i.e., a quiescent state is a state in which no output or internal transitions can be exceuted [Tre96a, Tre96b].

Definition 2.37 (Quiescent state)**.** Let $\mathcal{A}$ be an IOTS. A state $s \in S$ is *quiescent* if no outputs or internal transitions are enabled in that state, i.e., s is called quiescent if for all $a \in L^{\mathrm{O}} \cup \{ \tau \}$ we have $s \overset{a}{\not\rightarrow}$.

Example 2.38. Consider the IOTS $\mathcal{A}$ in Figure 2.6a. The states marked in gray are all quiescent, as these states have no outgoing output or internal transitions. □

Rather than being built from scratch like regular IOTSs, a SA is constructed by taking an existing (convergent) IOTS and adding δ-labelled self-loops to its quiescent states, and determinising the result [Tre08]. The δ-label is a special kind of output label that represents the observation of quiescence, and is not part of the actual output set L^{O}.

Definition 2.39 (Suspension Automata)**.** Given an IOTS $\mathcal{A} = \langle S, S^0_{\mathcal{A}}, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow \rangle$, let $\mathcal{A}' = \langle S, S^0_{\mathcal{A}}, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow' \rangle$ where $\rightarrow' \; = \; \rightarrow \; \cup \, \{ (s, \delta, s) \in S \times \{ \delta \} \times S \mid s \text{ is quiescent in } \mathcal{A} \}$. The *Suspension Automaton* corresponding to $\mathcal{A}$ is then the IOTS $\mathcal{S} = \mathit{det}(\mathcal{A})$.

Example 2.40. Figure 2.6 visualises how a SA is constructed for the IOTS $\mathcal{A}$ shown in Figure 2.6a. First, the automaton $\mathcal{A}'$ is obtained from $\mathcal{A}$ by introducing new δ-labelled self-loops (marked in bold) for the quiescent states of $\mathcal{A}$. Next, $\mathcal{A}'$ is determinised, resulting in the SA S. □

Table 2.1: Comparison of the features of LTSs, IOTSs, IOAs and SAs.

	LTSs	IOTSs	IOAs	SAs
internal actions	τ	τ	L^{H}	τ
inputs and outputs	-	+	+	+
input-enabledness required	-	+	+	+
nondeterminism allowed	+	+	+	-
divergence allowed	+	+	+	-
stand-alone entity	+	+	+	-
closed under determinisation	+	+	+	+
closed under action hiding	+	+	+	?
closed under parallel composition	+	+	+	?

Since a SA captures all possible observations of a given IOTS, including quiescence, SAs are perfectly suited to model the behaviour of both specifications and implementations in model-based testing frameworks such as the industry-leading `ioco` [Tre96a, Tre96b] framework. The `ioco` framework, in turn, is used by well-known test generation tools like TGV [JT05], the AGEDIS TOOL SET [HN04], TESTGEN [HT99], and TORX [BFd$^+$99, TB03].

However, SAs have some major limitations [Tre08]. First of all, SAs can only be constructed for convergent IOTSs, since it is not clear how the notions of divergence and quiescence can be reconciled. Furthermore, SAs must necessarily be deterministic. Both of these requirements clearly limit the number of systems that can be effectively modelled as SAs. SAs are also implicitly defined: they can only be constructed by taking an existing IOTS and applying the transformations described in Definition 2.39; they cannot be built from scratch.

Finally, the closure properties of SAs regarding parallel composition and action hiding have not been investigated, and are therefore unknown. Hence, there is no fully formalised theory for SAs. In the next chapter, we introduce Quiescent Transition Systems (QTSs), which also extend IOTSs with the notion of quiescence, and address all these shortcomings of SAs, except the convergence requirement. This requirement will in turn be lifted by Divergent Quiescent Transition Systems (DQTSs), which will be introduced in Chapter 4.

2.6 Summary

In this chapter, we have introduced Labelled Transition Systems (LTSs), Input-Output Transition Systems (IOTSs) and Input-Output Automata (IOAs). LTSs are used to model the behaviour of systems using states, transitions, a set of actions, and a special internal action τ. IOTSs extend LTSs by distinguishing between input actions and output actions. IOAs further generalise IOTSs by allowing multiple internal actions and partitioning the set of locally controlled actions (output actions and internal actions) to formalise the notion of fair executions.

We also briefly discussed Suspension Automata (SAs), which extend IOTSs with support for quiescence observations using special δ-labelled transitions. Table 2.1 compares some of the more important features of LTSs, IOTSs, IOAs and SAs. Clearly, SAs show some shortcomings, which we will address by introducing two new types of automata in the following chapters: Quiescent Transitions Systems (QTSs), which are based on IOTSs; and Divergent Quiescent Transition Systems (DQTSs), which are based on IOAs.

Chapter 3

Quiescent Transition Systems

This chapter introduces *Quiescent Transition Systems* (QTSs). Like the Suspension Automata (SAs) described in Section 2.5, QTSs are a specialisation of regular IOTSs that support the notion of quiescence. However, a QTS need not necessarily be deterministic, and can be built from scratch, whereas SAs are implicitly defined and have to be deterministic.

Furthermore, in this chapter we also closely examine several closure and commutativity properties of QTSs with regards to the determinisation, parallel composition and action hiding operations, something which has not been done for SAs. Finally, we also define what exactly it means for a QTS to be well-formed, thus establishing the semantic validity of the model. Note that the convergence requirement of SAs also applies to QTSs. This requirement will be lifted by the Divergent Quiescent Transition Systems (DQTSs) introduced in the next chapter.

Remark 3.1. A basic variant of QTSs was already used in [TBS11] in a testing framework. An earlier version of the QTS model described here has been published previously by the author in [STS12a, STS12b].

3.1 The QTS Model

As mentioned earlier, QTSs, like SAs, are a specialisation of regular IOTSs in which the observation of quiescence is explicitly modelled with a special δ-label.

Definition 3.2 (Quiescent Transition Systems)**.** A *Quiescent Transition System* (QTS) is a tuple $\mathcal{A} = \langle\, S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow \rangle$, where:

- S is a non-empty set of states;
- $S^0 \subseteq S$ is a non-empty set of initial states;
- L^{I} and L^{O} are disjoint sets of input and output labels, respectively. As for regular IOTSs, we require $\tau \notin L^{\mathrm{I}} \cup L^{\mathrm{O}}$. Furthermore, $\delta \notin L^{\mathrm{I}} \cup L^{\mathrm{O}}$ is a special output label that is used to denote the observation of quiescence;
- $\rightarrow\ \subseteq S \times (L^{\mathrm{I}} \cup L^{\mathrm{O}} \cup \{\, \tau, \delta \,\}) \times S$ is the transition relation.

As for IOTSs, we define $L = L^{\mathrm{I}} \cup L^{\mathrm{O}}$ and $L^{\tau} = L \cup \{\, \tau \,\}$. Given a QTS $\mathcal{A}$, we denote its components by $S_{\mathcal{A}}$, $S^0_{\mathcal{A}}$, $L^{\mathrm{I}}_{\mathcal{A}}$, $L^{\mathrm{O}}_{\mathcal{A}}$ and $\rightarrow_{\mathcal{A}}$; we omit the subscript when it is clear from the context which QTS is referred to.

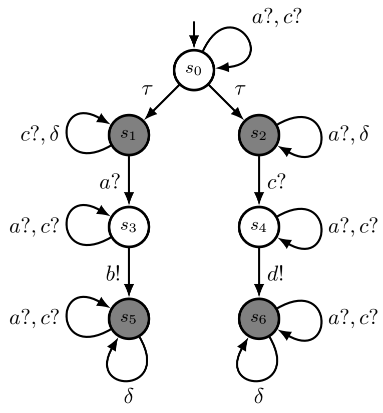


Figure 3.1: Visual representation of a QTS $\mathcal{A}$. The gray states are quiescent.

Like regular IOTSs, QTSs must be input-enabled. All other definitions introduced in Section 2.3 for IOTSs also apply to QTSs. As an important restriction, we do not allow divergent paths to occur in (regular) QTSs. As said earlier, this restriction will be lifted in DQTSs, which will be introduced in Chapter 4. Note that, similar to the δ-label of SAs, the δ-label is a special output label and is not part of L^{O}, the set of regular outputs.

Example 3.3. Figure 3.1 visualises a nondeterministic QTS $\mathcal{A}$. Note the δ-labelled transitions, which represent the observations of quiescence, i.e., the absence of enabled outputs. □

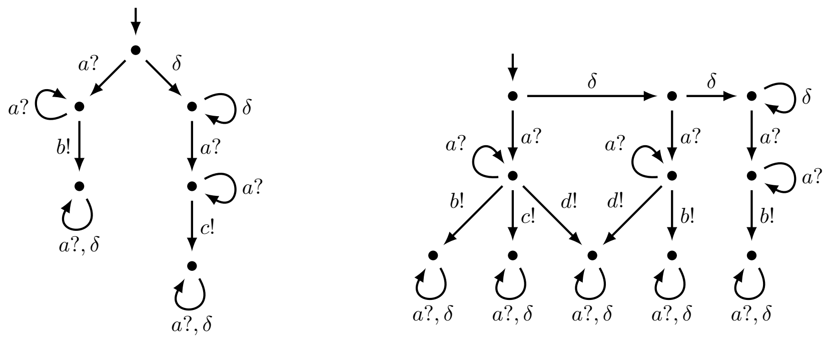
As mentioned in Definition 2.37, we distuinguish a special kind of state in IOTSs: the so-called *quiescent state*. For convenience' sake, we repeat the definition here and apply it to QTSs, and also introduce a notation for such states. Recall that a δ-labelled transition originating from such a state represents the absence of locally controlled behaviour, i.e., the observation of quiescence.

Definition 3.4 (Quiescent states of IOTSs or QTSs)**.** Let $\mathcal{A}$ be an IOTS or QTS. A state $s \in S$ is *quiescent*, denoted $q(s)$, if no locally controlled actions are enabled in that state, i.e., $q(s)$ if for all $a \in L^{\mathrm{O}} \cup \{ \tau \}$ we have $s \not\xrightarrow{a}$. The set of all quiescent states of $\mathcal{A}$ is denoted $q(\mathcal{A})$.

Example 3.5. Consider again the QTS $\mathcal{A}$ in Figure 3.1. States s_1, s_2, s_5 and s_6 are quiescent, since no locally controlled actions (in this case, the internal action τ and the outputs b and d) are enabled in those states. Note that state s_0 is not quiescent, as the internal action is enabled in this state. □

A system in a quiescent state will be idle until a new input is supplied, and hence quiescence may be observed in such a state.

An important detail to note: we consider a state s in which a τ-transition is enabled to not be quiescent, even if there is no regular output $a \in L^{\mathrm{O}}$ that is enabled in s. In this, we follow Tretmans [Tre08]. This may seem like an arbitrary restriction, but this constraint ensures that QTSs exhibit desirable closure properties (see Section 3.5). Furthermore, since internal actions are not observable, this restriction does not affect the external (observable) behaviour of the system.



(a) $\mathcal{A}$ (b) $\mathcal{B}$

Figure 3.2: The QTSs $\mathcal{A}$ and $\mathcal{B}$ that do not satisfy rule R3 and R4, respectively.

3.2 Well-formedness

In the previous sections, we have introduced QTSs, quiescent states and δ-transitions, but have not yet imposed any restrictions regarding the occurrence of δ-transitions. For instance, it is possible for a QTS to have a δ-transition that leads to a state in which outputs are enabled, but that would contradict the semantics of the δ-transition. As a consequence, the observable behaviour of a syntactically correct QTS might not correspond to any realistic specification or implementation. To exclude such situations, we define four additional rules that define exactly which QTSs exhibit correct behaviour. Such QTSs are called *well-formed.*

Definition 3.6 (Well-formedness for QTSs)**.** A QTS $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow \rangle$ is *well-formed* if it satisfies the following four rules for all states $s, s', s'' \in S$ and inputs $a \in L^{\mathrm{I}}$:

Rule R1 (Quiescence should be observable)**:** if $q(s)$, then $s \xrightarrow{\delta}$.

This rule requires that each quiescent state has an outgoing δ-transition, since in these states quiescence may be observed, as discussed earlier.

Rule R2 (Quiescent state after observation of quiescence)**:** if $s \xrightarrow{\delta} s'$, then $q(s')$.

This rule demands that a quiescent state is entered after a δ-transition, i.e., no output or internal transition may take place before a new input is provided. Note that a δ-transition may still take place. Since the notion of timing plays no role of QTSs, there is no particular observation duration associated with quiescence. Hence, a δ-transition means that the system has not produced any outputs indefinitely; therefore, enabling any outputs after a δ-transition would clearly be erroneous. Internal transitions are also not allowed directly after a δ-transition, as this simplifies the theory and ensures consistency with Definition 3.4. Again, since internal actions are not observable, this restriction does not affect the possible external (observable) behaviour of the system.

Rule R3 (Quiescence introduces no new behaviour)**:** if $s \xrightarrow{\delta} s'$, then $\mathit{traces}(s') \subseteq \mathit{traces}(s)$.

This rule prevents an observation of quiescence from enabling new behaviour, and prohibits a situation as in the QTS $\mathcal{A}$ in Figure 3.2a. Here the trace $a\,c$ can only be observed after observing quiescence. Thus, the observation of quiescence enables new behaviour.

This is counterintuitive, since there is no notion of timing associated with the QTS model. Hence, behaviour is enabled either immediately or not at all, but quiescence should never enable new behaviour.

Rule R4 (Continued quiescence preserves behaviour)**:** if $s \xrightarrow{\delta} s'$ and $s' \xrightarrow{\delta} s''$, then $\mathit{traces}(s'') = \mathit{traces}(s')$.

This rule ensures that the behaviour that can be observed after a single observation of quiescence, is the same as the behaviour that can be observed after multiple observations of quiescence in a row. This rule prohibits a situation as in the QTS $\mathcal{B}$ in Figure 3.2b. Here, the behaviour after two δ-transitions differs from the behaviour after the first. Since quiescence represents the fact that no outputs are observed and there is no notion of timing in the QTS model, there can be no difference between observing it once or multiple times in succession.

From now on, we assume that all given QTSs are well-formed.

Note that a trace of a QTS can contain a sequence of δ-actions. Although this might seem odd, it corresponds to the practical testing scenario of observing a time-out rather than an output more than once in a row.

In Section 3.6, we fully discuss the rationale behind these well-formedness rules and look at some alternative rules.

Remark 3.7. In [Wil07] a definition for *valid* deterministic SAs is given, alongside four conditions to which valid SAs should conform. These rules, although denoted differently, coincide exactly with the (less complex) rules R1, R2, R3 and R4 given above. Hence, valid SAs and well-formed QTSs form the same subclass of quiescence-enabled IOTSs.

3.3 Deltafication: from IOTS to QTS

Usually, the specification and implementation of a system are given as IOTSs, rather than QTSs. During testing, however, we typically observe the outputs of the system generated in response to inputs from the environment; thus, it is useful to be able to refer to the absence of outputs (i.e., quiescence) explicitly. Hence, we need a way to convert an IOTS to a QTS that captures all possible observations of it, including quiescence; this conversion is called deltafication and similar to the way SAs are constructed from IOTSs.

Definition 3.8 (Deltafication of IOTSs)**.** Let $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow_{\mathcal{A}} \rangle$ be an IOTS with $\delta \notin L$. The *deltafication* of $\mathcal{A}$ is the QTS $\delta(\mathcal{A}) = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow_{\delta} \rangle$, where $\rightarrow_{\delta}$ is defined as follows:

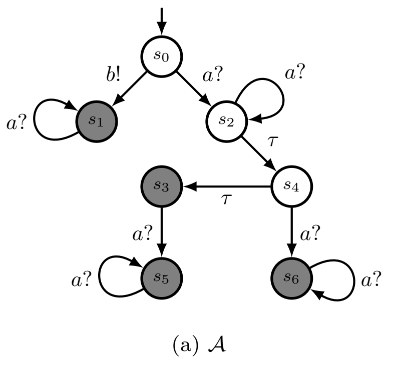
$$\rightarrow_{\delta} \;=\; \rightarrow_{\mathcal{A}} \;\cup\; \{\, (s, \delta, s) \in S \times \{\,\delta\,\} \times S \mid s \in q(\mathcal{A}) \,\}$$

Thus, the deltafication of an IOTS $\mathcal{A}$ simply adds δ-labelled self-loops to all quiescent states in $\mathcal{A}$.

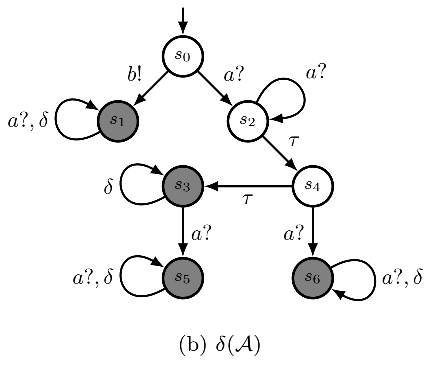
Example 3.9. Consider the IOTS $\mathcal{A}$ in Figure 3.3a. The quiescent states of $\mathcal{A}$ are s_1, s_3, s_5 and s_6, and have been marked gray. As a result, these states acquire a δ-labelled self-loop in the deltafication of $\mathcal{A}$, i.e., $\delta(\mathcal{A})$, as shown in Figure 3.3b. □

The following theorem shows that deltafication yields a well-formed QTS.

Theorem 3.10. *Given an IOTS $\mathcal{A}$ such that $\delta \notin L$, $\delta(\mathcal{A})$ is a well-formed QTS.*



(a) $\mathcal{A}$



(b) $\delta(\mathcal{A})$

Figure 3.3: Deltafication of an IOTS $\mathcal{A}$.

Proof. Let $\mathcal{A} = \langle S, S^0, L^I, L^O, \rightarrow_{\mathcal{A}} \rangle$ be an IOTS such that $\delta \notin L$, and let $\delta(\mathcal{A}) = \langle S, S^0, L^I, L^O, \rightarrow_\delta \rangle$ be its deltafication, as defined in Definition 3.8. To show that $\delta(\mathcal{A})$ is a well-formed QTS, we need to prove that $\delta(\mathcal{A})$ satisfies each of the rules R1, R2, R3 and R4. In the following, we use $traces_\delta(s)$ to denote the set of all traces of $\delta(\mathcal{A})$ starting in the state $s \in S$.

1. To prove that $\delta(\mathcal{A})$ satisfies rule R1, we must show that for all states $s \in S$:

$$\text{if } q(s), \text{ then } s \xrightarrow{\delta}_\delta$$

Let $s \in S$ be any state such that $q(s)$ holds in $\delta(\mathcal{A})$. Since deltafication doesn't change any existing transitions, $q(s)$ then also holds in $\mathcal{A}$. By Definition 3.8, we have $(s, \delta, s) \in \rightarrow_\delta$ after deltafication and therefore $s \xrightarrow{\delta}_\delta$.

2. To prove that $\delta(\mathcal{A})$ satisfies rule R2, we must show that for all states $s, s' \in S$:

$$\text{if } s \xrightarrow{\delta}_\delta s', \text{ then } q(s')$$

Consider any transition $s \xrightarrow{\delta}_\delta s'$ in $\delta(\mathcal{A})$ with $s, s' \in S$. By Definition 3.8, we have $s = s'$, and s (and therefore also s') is quiescent.

3. To prove that $\delta(\mathcal{A})$ satisfies rule R3, we must show that for all states $s, s' \in S$:

$$\text{if } s \xrightarrow{\delta}_\delta s', \text{ then } traces_\delta(s') \subseteq traces_\delta(s)$$

Consider any transition $s \xrightarrow{\delta}_\delta s'$ in $\delta(\mathcal{A})$ with $s, s' \in S$. By Definition 3.8, we have $s = s'$, and therefore $traces_\delta(s') \subseteq traces_\delta(s)$.

4. To prove that $\delta(\mathcal{A})$ satisfies rule R4, we must show that for all states $s, s', s'' \in S$:

$$\text{if } s \xrightarrow{\delta}_\delta s' \text{ and } s' \xrightarrow{\delta}_\delta s'', \text{ then } traces_\delta(s'') = traces_\delta(s')$$

Consider any pair of transitions $s \xrightarrow{\delta}_\delta s'$ and $s' \xrightarrow{\delta}_\delta s''$ with $s, s', s'' \in S$. By Definition 3.8, we have $s' = s''$, and therefore $traces_\delta(s') = traces_\delta(s'')$. □

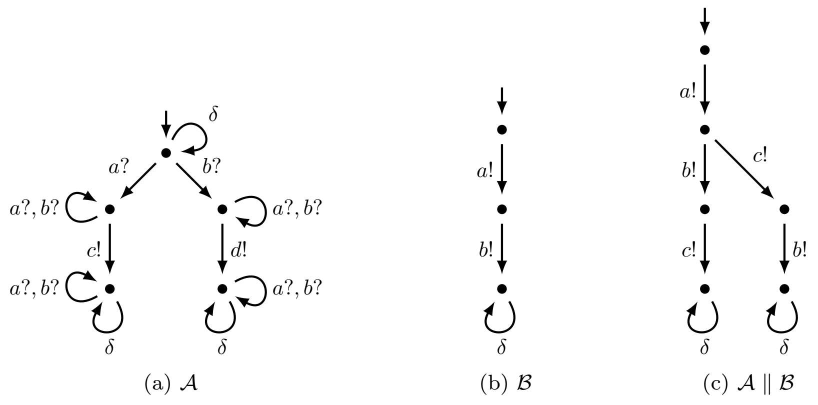


Figure 3.4: The QTSs $\mathcal{A}$, $\mathcal{B}$, and their parallel composition $\mathcal{A} \parallel \mathcal{B}$.

3.4 Operations

Since QTSs are a specialisation of IOTSs, all operations that are applicable to IOTSs (such as determinisation, parallel composition and hiding of actions) are also applicable to QTSs. Determinisation for QTSs is exactly the same as for IOTSs, but there are some minor differences for parallel composition and action hiding.

3.4.1 Parallel Composition

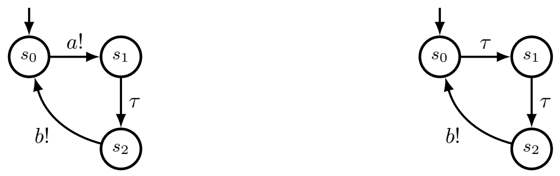
Similar to IOTSs, we require parallel composed QTSs to synchronise on shared inputs and complementary input-output pairs. However, we also require QTSs to synchronise on δ-transitions, as a parallel composition of two QTSs can only be quiescent when both component QTSs are.

Definition 3.11 (Parallel composition of QTSs)**.** Let $\mathcal{A} = \langle S_\mathcal{A}, S^0_\mathcal{A}, L^\mathrm{I}_\mathcal{A}, L^\mathrm{O}_\mathcal{A}, \rightarrow_\mathcal{A} \rangle$ and $\mathcal{B} = \langle S_\mathcal{B}, S^0_\mathcal{B}, L^\mathrm{I}_\mathcal{B}, L^\mathrm{O}_\mathcal{B}, \rightarrow_\mathcal{B} \rangle$ be two QTSs such that $L^\mathrm{O}_\mathcal{A} \cap L^\mathrm{O}_\mathcal{B} = \emptyset$. The *parallel composition* of $\mathcal{A}$ and $\mathcal{B}$ is the QTS $\mathcal{A} \parallel \mathcal{B} = \langle S_{\mathcal{A}\parallel\mathcal{B}}, S^0_{\mathcal{A}\parallel\mathcal{B}}, L^\mathrm{I}_{\mathcal{A}\parallel\mathcal{B}}, L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}}, \rightarrow_{\mathcal{A}\parallel\mathcal{B}} \rangle$, where $S_{\mathcal{A}\parallel\mathcal{B}}$, $S^0_{\mathcal{A}\parallel\mathcal{B}}$, $L^\mathrm{I}_{\mathcal{A}\parallel\mathcal{B}}$, $L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}}$ and $\rightarrow_{\mathcal{A}\parallel\mathcal{B}}$ are defined as follows:

$$\begin{array}{lcl}
S_{\mathcal{A}\parallel\mathcal{B}} & = & S_\mathcal{A} \times S_\mathcal{B} \\
S^0_{\mathcal{A}\parallel\mathcal{B}} & = & S^0_\mathcal{A} \times S^0_\mathcal{B} \\
L^\mathrm{I}_{\mathcal{A}\parallel\mathcal{B}} & = & (L^\mathrm{I}_\mathcal{A} \cup L^\mathrm{I}_\mathcal{B}) \setminus (L^\mathrm{O}_\mathcal{A} \cup L^\mathrm{O}_\mathcal{B}) \\
L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}} & = & L^\mathrm{O}_\mathcal{A} \cup L^\mathrm{O}_\mathcal{B} \\
\rightarrow_{\mathcal{A}\parallel\mathcal{B}} & = & \{\, ((s,t), a, (s',t')) \in S_{\mathcal{A}\parallel\mathcal{B}} \times ((L_\mathcal{A} \cap L_\mathcal{B}) \cup \{\,\delta\,\}) \times S_{\mathcal{A}\parallel\mathcal{B}} \mid s \xrightarrow{a}_\mathcal{A} s' \wedge t \xrightarrow{a}_\mathcal{B} t' \,\} \\
 & \cup & \{\, ((s,t), a, (s',t)) \in S_{\mathcal{A}\parallel\mathcal{B}} \times ((L_\mathcal{A} \setminus L_\mathcal{B}) \cup \{\,\tau\,\}) \times S_{\mathcal{A}\parallel\mathcal{B}} \mid s \xrightarrow{a}_\mathcal{A} s' \,\} \\
 & \cup & \{\, ((s,t), a, (s,t')) \in S_{\mathcal{A}\parallel\mathcal{B}} \times ((L_\mathcal{B} \setminus L_\mathcal{A}) \cup \{\,\tau\,\}) \times S_{\mathcal{A}\parallel\mathcal{B}} \mid t \xrightarrow{a}_\mathcal{B} t' \,\}
\end{array}$$

As with parallel composed IOTSs, we have $L_{\mathcal{A}\parallel\mathcal{B}} = L^\mathrm{I}_{\mathcal{A}\parallel\mathcal{B}} \cup L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}} = L_\mathcal{A} \cup L_\mathcal{B}$.

The first clause of $\rightarrow_{\mathcal{A}\parallel\mathcal{B}}$ ensures that parallel composed QTSs synchronise both on shared actions and the δ-label. The next two clauses enable them to perform non-shared actions independently from each other.



(a) $\mathcal{A}$ (b) $\mathcal{A} \backslash \{ a \}$

Figure 3.5: The QTSs $\mathcal{A}$ and $\mathcal{A} \backslash \{ a \}$.

Example 3.12. See Figure 3.4 for two QTSs $\mathcal{A}$ and $\mathcal{B}$, and their parallel composition $\mathcal{A} \parallel \mathcal{B}$. Note the synchronisation on the δ-transitions. □

3.4.2 Action Hiding

The hiding of outputs in QTSs is exactly the same as for IOTSs, except that we do not allow the special output label δ to be hidden, as this label doesn't represent a specific output but rather (the observation of) a lack of outputs. Furthermore, since we disallow divergent paths in QTSs, we do not allow the hiding of output labels to lead to the creation of τ-loops, i.e., cyclic divergent paths.

Definition 3.13 (Action hiding in QTSs)**.** Let $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow_{\mathcal{A}} \rangle$ be a QTS and $H \subseteq L^{\mathrm{O}}$ a set of output labels. If $\mathcal{A}$ does not contain a cyclic path $s_0\, a_1\, s_1\, a_2\, \dots$ such that for all a_i we have $a_i \in H \cup \{ \tau \}$, then one can *hide* H in $\mathcal{A}$ to obtain the QTS $\mathcal{A} \setminus H = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}_H, \rightarrow_H \rangle$, where L^{O}_H and $\rightarrow_H$ are defined as follows:

$$\begin{array}{rcll} L^{\mathrm{O}}_H & = & L^{\mathrm{O}} \setminus H & \\ \rightarrow_H & = & \{ (s, a, s') \in \rightarrow_{\mathcal{A}} & \mid \; a \notin H \} \\ & \cup & \{ (s, \tau, s') \in S \times \{ \tau \} \times S & \mid \; \exists\, a \in H \,.\, (s, a, s') \in \rightarrow_{\mathcal{A}} \} \end{array}$$

Example 3.14. Consider the QTS $\mathcal{A}$ in Figure 3.5a and assume $L_{\mathcal{A}} = \{ a, b \}$. After hiding the output action a, the resulting QTS is $\mathcal{A} \backslash \{ a \}$, which is shown in Figure 3.5b. We have $L_{\mathcal{A} \backslash \{ a \}} = \{ b \}$. Note that we cannot hide both the output actions a and b, as this would result in the τ-loop $s_0 \tau s_1 \tau s_2 \tau s_0$. □

3.5 Properties

In this section, we present several important results regarding QTSs. First, they are closed under all operations mentioned previously. Second, they have many useful commutativity properties regarding function composition of deltafication and the other operations.

3.5.1 Closure Properties

It turns out that well-formed QTSs are closed under all operations defined thus far: determinisation, parallel composition, and action hiding. Therefore, these operations are indeed well-defined for well-formed QTSs.

Theorem 3.15. *Well-formed QTSs are closed under determinisation, i.e., given a well-formed QTS $\mathcal{A}$, det($\mathcal{A}$) is also a well-formed QTS.*

Proof. Let $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow_{\mathcal{A}} \rangle$ be a well-formed QTS and let $det(\mathcal{A}) = \langle S_{\mathrm{D}}, S^0_{\mathrm{D}}, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow_{\mathrm{D}} \rangle$ be its determinisation, as defined in Definition 2.11. To prove that well-formed QTSs are closed under determinisation we must show that $det(\mathcal{A})$ is a well-formed QTS, i.e., that it satisfies each of the rules R1, R2, R3 and R4. In the following, we use $traces_{\mathrm{D}}(U)$ to denote the set of all traces of $det(\mathcal{A})$ starting in the state $U \in S_{\mathrm{D}}$.

1. To prove that $det(\mathcal{A})$ satisfies rule R1, we must show that for all states $U \in S_{\mathrm{D}}$:

$$\text{if } q(U) \text{, then } U \xrightarrow{\delta}_{\mathrm{D}}$$

   Let $U \in S_{\mathrm{D}}$ be any state such that $q(U)$ holds in $det(\mathcal{A})$. This implies that all states $s \in U$ are quiescent in $\mathcal{A}$. From rule R1 it follows that for every state $s \in U$ there exists another state $s' \in S$ such that $s \xrightarrow{\delta}_{\mathcal{A}} s'$. Therefore $reach_{\mathcal{A}}(U, \delta) \neq \emptyset$. By Definition 2.11, we then have $(U, \delta, reach_{\mathcal{A}}(U, \delta)) \in \rightarrow_{\mathrm{D}}$. Thus, $U \xrightarrow{\delta}_{\mathrm{D}}$.

2. To prove that $det(\mathcal{A})$ satisfies rule R2, we must show that for all states $U, V \in S_{\mathrm{D}}$:

$$\text{if } U \xrightarrow{\delta}_{\mathrm{D}} V \text{, then } q(V)$$

   Consider any transition $U \xrightarrow{\delta}_{\mathrm{D}} V$ with $U, V \in S_{\mathrm{D}}$. If $U \xrightarrow{\delta}_{\mathrm{D}} V$, then, by Definition 2.11, $V = reach_{\mathcal{A}}(U, \delta)$ and $V \neq \emptyset$. Hence, for every state $s' \in V$ there exists a state $s \in U$ such that $s \xrightarrow{\delta}_{\mathcal{A}} s'$. Using rule R2 we can then conclude that every $s' \in V$ is quiescent in $\mathcal{A}$, thus $q(V)$ holds in $det(\mathcal{A})$.

3. To prove that $det(\mathcal{A})$ satisfies rule R3, we must show that for all states $U, V \in S_{\mathrm{D}}$:

$$\text{if } U \xrightarrow{\delta}_{\mathrm{D}} V \text{, then } traces_{\mathrm{D}}(V) \subseteq traces_{\mathrm{D}}(U)$$

   Consider any transition $U \xrightarrow{\delta}_{\mathrm{D}} V$ with $U, V \in S_{\mathrm{D}}$. Assume $\sigma \in traces_{\mathrm{D}}(V)$. We must show that also $\sigma \in traces_{\mathrm{D}}(U)$. If $\sigma \in traces_{\mathrm{D}}(V)$, then there clearly must exist a state $s' \in V$ such that $s' \xRightarrow{\sigma}_{\mathcal{A}}$. Since $U \xrightarrow{\delta}_{\mathrm{D}} V$, it follows from Definition 2.11 that $V = reach_{\mathcal{A}}(U, \delta)$ and $V \neq \emptyset$. Hence, there must exist a state $s \in U$ such that $s \xrightarrow{\delta}_{\mathcal{A}} s'$. Using rule R3 we can then conclude that $traces_{\mathcal{A}}(s') \subseteq traces_{\mathcal{A}}(s)$, and therefore $s \xRightarrow{\sigma}_{\mathcal{A}}$. Since $s \in U$, it follows that $\sigma \in traces_{\mathrm{D}}(U)$.

4. To prove that $det(\mathcal{A})$ satisfies rule R4, we must show that for all states $U, V, W \in S_{\mathrm{D}}$:

$$\text{if } U \xrightarrow{\delta}_{\mathrm{D}} V \text{ and } V \xrightarrow{\delta}_{\mathrm{D}} W \text{, then } traces_{\mathrm{D}}(W) = traces_{\mathrm{D}}(V)$$

   Consider any pair of transitions $U \xrightarrow{\delta}_{\mathrm{D}} V$ and $V \xrightarrow{\delta}_{\mathrm{D}} W$, with $U, V, W \in S_{\mathrm{D}}$. To prove that $traces_{\mathrm{D}}(W) = traces_{\mathrm{D}}(V)$, we must show that both $traces_{\mathrm{D}}(W) \subseteq traces_{\mathrm{D}}(V)$ and $traces(V) \subseteq traces_{\mathrm{D}}(W)$. The former follows directly from rule R3, so all that's left to prove is that $traces_{\mathrm{D}}(V) \subseteq traces_{\mathrm{D}}(W)$.

   Assume $\sigma \in traces_{\mathrm{D}}(V)$. We must show that also $\sigma \in traces_{\mathrm{D}}(W)$. If $\sigma \in traces_{\mathrm{D}}(V)$, then there clearly must exist a state $s' \in V$ such that $s' \xRightarrow{\sigma}_{\mathcal{A}}$. Since $U \xrightarrow{\delta}_{\mathrm{D}} V$, it follows that there exists a $s \in U$ such that $s \xrightarrow{\delta}_{\mathcal{A}} s'$. Furthermore, it follows from rule R2 that V is quiescent, and therefore all states in V are quiescent, including s'. Since $V \xrightarrow{\delta}_{\mathrm{D}} W$, we have $W = reach(V, \delta)$ and $W \neq \emptyset$. We can then conclude, using rule R1, that there must exist a state $s'' \in W$ such that $s' \xrightarrow{\delta}_{\mathcal{A}} s''$. Thus, we have $s \xrightarrow{\delta}_{\mathcal{A}} s' \xrightarrow{\delta}_{\mathcal{A}} s''$. From rule R4 it then follows that $traces(s'') = traces(s')$ and consequently $s'' \xRightarrow{\sigma}_{\mathcal{A}}$. Since $s'' \in W$, it follows that $\sigma \in traces_{\mathrm{D}}(W)$. □

Theorem 3.16. *Well-formed QTSs are closed under parallel composition, i.e., given two well-formed QTSs $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \parallel \mathcal{B}$ is also a well-formed QTS.*

Proof. Let $\mathcal{A} = \langle S_\mathcal{A}, S^0_\mathcal{A}, L^\mathrm{I}_\mathcal{A}, L^\mathrm{O}_\mathcal{A}, \rightarrow_\mathcal{A} \rangle$ and $\mathcal{B} = \langle S_\mathcal{B}, S^0_\mathcal{B}, L^\mathrm{I}_\mathcal{B}, L^\mathrm{O}_\mathcal{B}, \rightarrow_\mathcal{B} \rangle$ be two well-formed QTSs such that $L^\mathrm{O}_\mathcal{A} \cap L^\mathrm{O}_\mathcal{B} = \emptyset$. Furthermore, let $\mathcal{A} \parallel \mathcal{B} = \langle S_{\mathcal{A}\parallel\mathcal{B}}, S^0_{\mathcal{A}\parallel\mathcal{B}}, L^\mathrm{I}_{\mathcal{A}\parallel\mathcal{B}}, L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}}, \rightarrow_{\mathcal{A}\parallel\mathcal{B}} \rangle$ be their parallel composition, as defined in Definition 3.11. To prove that well-formed QTSs are closed under parallel composition we must show that $\mathcal{A} \parallel \mathcal{B}$ is a well-formed QTS, i.e., we need to prove that $\mathcal{A} \parallel \mathcal{B}$ satisfies each of the rules R1, R2, R3 and R4.

1. To prove that $\mathcal{A} \parallel \mathcal{B}$ satisfies rule R1, we must show that for every state $(s,t) \in S_{\mathcal{A}\parallel\mathcal{B}}$:

$$\text{if } q((s,t)), \text{ then } (s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}}$$

   Let $(s,t) \in S_{\mathcal{A}\parallel\mathcal{B}}$ be any state such that $q((s,t))$ holds in $\mathcal{A} \parallel \mathcal{B}$. In this case, there is no $a \in L^\mathrm{O}_{\mathcal{A}\parallel\mathcal{B}} \cup \{\tau\}$ such that $(s,t) \xrightarrow{a}_{\mathcal{A}\parallel\mathcal{B}}$. Since both $\mathcal{A}$ and $\mathcal{B}$ are input-enabled, it follows from Definition 3.11 that there is no $a \in L^\mathrm{O}_\mathcal{A} \cup \{\tau\}$ such that $s \xrightarrow{a}_\mathcal{A}$ and no $a \in L^\mathrm{O}_\mathcal{B} \cup \{\tau\}$ such that $t \xrightarrow{a}_\mathcal{B}$. Hence, both s and t are quiescent, and by rule R1 we have $s \xrightarrow{\delta}_\mathcal{A}$ and $t \xrightarrow{\delta}_\mathcal{B}$. From Definition 3.11 it then follows that $(s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}}$.

2. To prove that $\mathcal{A} \parallel \mathcal{B}$ satisfies rule R2, we must show that for all pairs of states $(s,t),(s',t') \in S_{\mathcal{A}\parallel\mathcal{B}}$:

$$\text{if } (s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s',t'), \text{ then } q((s',t'))$$

   Consider any transition $(s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s',t')$ with $(s,t),(s',t') \in S_{\mathcal{A}\parallel\mathcal{B}}$. From Definition 3.11 it then follows that $s \xrightarrow{\delta}_\mathcal{A} s'$ and $t \xrightarrow{\delta}_\mathcal{B} t'$. By rule R2, both s' and t' are quiescent. Thus, by Definition 3.11, $q((s',t'))$ holds in $\mathcal{A} \parallel \mathcal{B}$.

3. To prove that $\mathcal{A} \parallel \mathcal{B}$ satisfies rule R3, we must show that for all pairs of states $(s,t),(s',t') \in S_{\mathcal{A}\parallel\mathcal{B}}$:

$$\text{if } (s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s',t'), \text{ then } \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s',t')) \subseteq \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s,t))$$

   Consider any transition $(s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s',t')$ with $(s,t),(s',t') \in S_{\mathcal{A}\parallel\mathcal{B}}$. Assume $\sigma \in \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s',t'))$. We have to show that also $\sigma \in \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s,t))$. Since $(s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s',t')$, it follows from Definition 3.11 that $s \xrightarrow{\delta}_\mathcal{A} s'$ and $t \xrightarrow{\delta}_\mathcal{B} t'$. By rule R3, we then have $\mathit{traces}_\mathcal{A}(s') \subseteq \mathit{traces}_\mathcal{A}(s)$ and $\mathit{traces}_\mathcal{B}(t') \subseteq \mathit{traces}_\mathcal{B}(t)$.

   Additionally, note that $\sigma \in \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s',t'))$ implies that there is a path

$$\pi = (s'_0,t'_0) \xrightarrow{a_1}_{\mathcal{A}\parallel\mathcal{B}} (s'_1,t'_1) \xrightarrow{a_2}_{\mathcal{A}\parallel\mathcal{B}} \cdots \xrightarrow{a_{n-1}}_{\mathcal{A}\parallel\mathcal{B}} (s'_{n-1},t'_{n-1}) \xrightarrow{a_n}_{\mathcal{A}\parallel\mathcal{B}} (s'_n,t'_n)$$

   for some $n \geq |\sigma|$, where $(s'_0,t'_0) = (s',t')$ and $\mathit{trace}(\pi) = \sigma$. Note that some of the actions a_i can be equal to τ, and that not all states s_i and t_i have to be distinct.

   We prove by induction on the length of the path π that (1) $s' \xRightarrow{\rho_\mathcal{A}}_\mathcal{A} s'_n$ and $t' \xRightarrow{\rho_\mathcal{B}}_\mathcal{B} t'_n$, where $\rho_\mathcal{A} = \sigma \upharpoonright (L_\mathcal{A} \cup \{\delta\})$ and $\rho_\mathcal{B} = \sigma \upharpoonright (L_\mathcal{B} \cup \{\delta\})$, that (2) $s \xRightarrow{\rho_\mathcal{A}}_\mathcal{A}$ and $t \xRightarrow{\rho_\mathcal{B}}_\mathcal{B}$, and that (3) $(s,t) \xRightarrow{\sigma}_{\mathcal{A}\parallel\mathcal{B}} (s_m,t_m)$ for every pair $(s_m,t_m) \in \mathit{reach}_\mathcal{A}(s,\rho_\mathcal{A}) \times \mathit{reach}_\mathcal{B}(t,\rho_\mathcal{B})$. Note that the last part implies that $\sigma \in \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s,t))$, which is what we needed to show (the first two parts are needed for the induction).

Base case. Let $|\pi| = 0$, i.e., π is the empty path and $(s'_n, t'_n) = (s', t')$. This implies that $\sigma = \rho_{\mathcal{A}} = \rho_{\mathcal{B}} = \epsilon$, and hence $s' \xrightarrow{\rho_{\mathcal{A}}}_{\mathcal{A}} s'_n$ and $t' \xrightarrow{\rho_{\mathcal{B}}}_{\mathcal{B}} t'_n$. Also, $s \xrightarrow{\rho_{\mathcal{A}}}_{\mathcal{A}}$ and $t \xrightarrow{\rho_{\mathcal{B}}}_{\mathcal{B}}$ since $\epsilon \in \mathit{traces}_{\mathcal{A}}(s)$ and $\epsilon \in \mathit{traces}_{\mathcal{B}}(t)$. To see why $(s,t) \xrightarrow{\sigma}_{\mathcal{A}\|\mathcal{B}} (s_m, t_m)$ for every $(s_m, t_m) \in \mathit{reach}_{\mathcal{A}}(s, \rho_{\mathcal{A}}) \times \mathit{reach}_{\mathcal{B}}(t, \rho_{\mathcal{B}})$, note that since $\sigma = \rho_{\mathcal{A}} = \rho_{\mathcal{B}} = \epsilon$, $\mathit{reach}_{\mathcal{A}}(t, \rho_{\mathcal{A}})$ and $\mathit{reach}_{\mathcal{B}}(t, \rho_{\mathcal{B}})$ contain precisely all states that can be reached from s and t, respectively, by only taking τ-transitions. By Definition 3.11, these τ-transitions (if any) can also be executed in all possible interleavings starting from (s,t), since $\mathcal{A}$ and $\mathcal{B}$ do not synchronise on τ-transitions.

Inductive case. Let π' be the path from (s'_0, t'_0) to (s'_{n-1}, t'_{n-1}), and let $\sigma' = \mathit{trace}(\pi')$. Assume that (1) $s' \xrightarrow{\rho'_{\mathcal{A}}}_{\mathcal{A}} s'_{n-1}$ and $t' \xrightarrow{\rho'_{\mathcal{B}}}_{\mathcal{B}} t'_{n-1}$, where $\rho'_{\mathcal{A}} = \sigma' \upharpoonright (L_{\mathcal{A}} \cup \{\delta\})$ and $\rho'_{\mathcal{B}} = \sigma' \upharpoonright (L_{\mathcal{B}} \cup \{\delta\})$, that (2) $s \xrightarrow{\rho'_{\mathcal{A}}}_{\mathcal{A}}$ and $t \xrightarrow{\rho'_{\mathcal{B}}}_{\mathcal{B}}$, and that (3) $(s,t) \xrightarrow{\sigma'}_{\mathcal{A}\|\mathcal{B}} (s_m, t_m)$ for every pair $(s_m, t_m) \in \mathit{reach}_{\mathcal{A}}(s, \rho'_{\mathcal{A}}) \times \mathit{reach}_{\mathcal{B}}(t, \rho'_{\mathcal{B}})$. Let $\sigma = \sigma' a = \mathit{trace}(\pi)$. Since $\sigma \in \mathit{traces}_{\mathcal{A}\|\mathcal{B}}((s', t'))$, we have $a \in L_{\mathcal{A}\|\mathcal{B}} \cup \{\epsilon, \delta\}$. We look at the cases $a = \epsilon$, $a \in L_{\mathcal{A}} \setminus L_{\mathcal{B}}$, $a \in L_{\mathcal{B}} \setminus L_{\mathcal{A}}$, and $a \in (L_{\mathcal{A}} \cap L_{\mathcal{B}}) \cup \{\delta\}$ separately.

- If $a = \epsilon$, then apparently $a_n = \tau$ and $\sigma = \sigma'\epsilon = \sigma'$. By Definition 3.11, this implies that either $s'_{n-1} = s'_n$ and $t'_{n-1} \xrightarrow{\tau}_{\mathcal{B}} t'_n$, or $t'_{n-1} = t'_n$ and $s'_{n-1} \xrightarrow{\tau}_{\mathcal{A}} s'_n$. Both cases imply that $s' \xrightarrow{\rho_{\mathcal{A}}}_{\mathcal{A}} s'_n$ and $t' \xrightarrow{\rho_{\mathcal{B}}}_{\mathcal{B}} t'_n$, since $\rho_i = \rho'_i \cdot (a \upharpoonright (L_{\mathcal{A}} \cup \{\delta\})) = \rho'_i \cdot (\epsilon \upharpoonright (L_{\mathcal{A}} \cup \{\delta\})) = \rho'_i$ for $i \in \{\mathcal{A}, \mathcal{B}\}$ and we assumed $s' \xrightarrow{\rho'_{\mathcal{A}}}_{\mathcal{A}} s'_{n-1}$ and $t' \xrightarrow{\rho'_{\mathcal{B}}}_{\mathcal{B}} t'_{n-1}$. Also, since $\rho'_i = \rho_i$ and $\sigma' = \sigma$, by the induction hypothesis we have $s \xrightarrow{\rho_{\mathcal{A}}}_{\mathcal{A}}$, $t \xrightarrow{\rho_{\mathcal{B}}}_{\mathcal{B}}$, and $(s,t) \xrightarrow{\sigma}_{\mathcal{A}\|\mathcal{B}} (s_m, t_m)$ for every $(s_m, t_m) \in \mathit{reach}_{\mathcal{A}}(s, \rho_{\mathcal{A}}) \times \mathit{reach}_{\mathcal{B}}(t, \rho_{\mathcal{B}})$.
- If $a \in L_{\mathcal{A}} \setminus L_{\mathcal{B}}$, then $a_n = a$ and $(s'_{n-1}, t'_{n-1}) \xrightarrow{a_n}_{\mathcal{A}\|\mathcal{B}} (s'_n, t'_n)$ implies, by Definition 3.11, that $s'_{n-1} \xrightarrow{a}_{\mathcal{A}} s'_n$ and $t'_{n-1} = t'_n$. Since $s' \xrightarrow{\rho'_{\mathcal{A}}}_{\mathcal{A}} s'_{n-1}$ and $\rho_{\mathcal{A}} = \rho'_{\mathcal{A}} \cdot a$, this implies that $s' \xrightarrow{\rho_{\mathcal{A}}}_{\mathcal{A}} s'_n$, and since $t' \xrightarrow{\rho'_{\mathcal{B}}}_{\mathcal{B}} t'_{n-1}$ and $\rho'_{\mathcal{B}} = \rho_{\mathcal{B}}$, we have $t' \xrightarrow{\rho_{\mathcal{B}}}_{\mathcal{B}} t'_n$. Since $\mathit{traces}_{\mathcal{A}}(s') \subseteq \mathit{traces}_{\mathcal{A}}(s)$ and $\mathit{traces}_{\mathcal{B}}(t') \subseteq \mathit{traces}_{\mathcal{B}}(t)$, also $s \xrightarrow{\rho_{\mathcal{A}}}_{\mathcal{A}}$ and $t \xrightarrow{\rho_{\mathcal{B}}}_{\mathcal{B}}$. Clearly, $\mathit{reach}_{\mathcal{B}}(t, \rho_{\mathcal{B}}) = \mathit{reach}_{\mathcal{B}}(t, \rho'_{\mathcal{B}})$, since $\rho_{\mathcal{B}} = \rho'_{\mathcal{B}}$. Furthermore, for every state $v \in \mathit{reach}_{\mathcal{A}}(s, \rho_{\mathcal{A}})$ there exists a state $u \in \mathit{reach}_{\mathcal{A}}(s, \rho'_{\mathcal{A}})$ such that $u \xrightarrow{a}_{\mathcal{A}} v$. Hence, since $(s,t) \xrightarrow{\sigma'}_{\mathcal{A}\|\mathcal{B}} (s_m, t_m)$ for every pair $(s_m, t_m) \in \mathit{reach}_{\mathcal{A}}(s, \rho'_{\mathcal{A}}) \times \mathit{reach}_{\mathcal{B}}(t, \rho'_{\mathcal{B}})$, by Definition 3.11 also $(s,t) \xrightarrow{\sigma}_{\mathcal{A}\|\mathcal{B}} (s_n, t_n)$ for every pair $(s_n, t_n) \in \mathit{reach}_{\mathcal{A}}(s, \rho_{\mathcal{A}}) \times \mathit{reach}_{\mathcal{B}}(t, \rho_{\mathcal{B}})$.
- If $a \in L_{\mathcal{B}} \setminus L_{\mathcal{A}}$, the proof is symmetrical to the previous case.
- If $a \in L_{\mathcal{A}} \cap L_{\mathcal{B}}$ or $a = \delta$, then $a_n = a$ and $(s'_{n-1}, t'_{n-1}) \xrightarrow{a_n}_{\mathcal{A}\|\mathcal{B}} (s'_n, t'_n)$ implies, by Definition 3.11, that $s'_{n-1} \xrightarrow{a}_{\mathcal{A}} s'_n$ and $t'_{n-1} \xrightarrow{a}_{\mathcal{B}} t'_n$. Since $s' \xrightarrow{\rho'_{\mathcal{A}}}_{\mathcal{A}} s'_{n-1}$ and $\rho_{\mathcal{A}} = \rho'_{\mathcal{A}} \cdot a$, this implies that $s' \xrightarrow{\rho_{\mathcal{A}}}_{\mathcal{A}} s'_n$; $t' \xrightarrow{\rho_{\mathcal{A}}}_{\mathcal{B}} t'_n$ follows symmetrically. Since $\mathit{traces}_{\mathcal{A}}(s') \subseteq \mathit{traces}_{\mathcal{A}}(s)$ and $\mathit{traces}_{\mathcal{B}}(t') \subseteq \mathit{traces}_{\mathcal{B}}(t)$, also $s \xrightarrow{\rho_{\mathcal{A}}}_{\mathcal{A}}$ and $t \xrightarrow{\rho_{\mathcal{B}}}_{\mathcal{B}}$. Furthermore, for every state $v \in \mathit{reach}_{\mathcal{A}}(s, \rho_{\mathcal{A}})$ there exists a state $u \in \mathit{reach}_{\mathcal{A}}(s, \rho'_{\mathcal{A}})$ such that $u \xrightarrow{a}_{\mathcal{A}} v$; for $\mathit{reach}_{\mathcal{B}}(t, \rho_{\mathcal{B}})$ the same property (but with $\mathit{reach}_{\mathcal{B}}(t, \rho'_{\mathcal{B}})$ rather than $\mathit{reach}_{\mathcal{A}}(s, \rho'_{\mathcal{A}})$) holds. Hence, since $(s,t) \xrightarrow{\sigma'}_{\mathcal{A}\|\mathcal{B}} (s_m, t_m)$ for every pair $(s_m, t_m) \in \mathit{reach}_{\mathcal{A}}(s, \rho'_{\mathcal{A}}) \times \mathit{reach}_{\mathcal{B}}(t, \rho'_{\mathcal{B}})$, by Definition 3.11 also $(s,t) \xrightarrow{\sigma}_{\mathcal{A}\|\mathcal{B}} (s_n, t_n)$ for every pair $(s_n, t_n) \in \mathit{reach}_{\mathcal{A}}(s, \rho_{\mathcal{A}}) \times \mathit{reach}_{\mathcal{B}}(t, \rho_{\mathcal{B}})$.

4. To prove that $\mathcal{A} \parallel \mathcal{B}$ satisfies rule R4, we must show that for all pairs of states $(s, t), (s', t')$,
$(s'', t'') \in S_{\mathcal{A}\parallel\mathcal{B}}$:

$$\text{if } (s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s',t') \text{ and } (s',t') \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s'',t''),$$
$$\text{then } \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s',t')) = \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s'',t''))$$

Consider any pair of transitions $(s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s',t')$ and $(s',t') \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s'',t'')$ with (s,t), $(s',t'), (s'',t'') \in S_{\mathcal{A}\parallel\mathcal{B}}$. From Definition 3.11 it follows that $s \xrightarrow{\delta}_{\mathcal{A}} s'$, $s' \xrightarrow{\delta}_{\mathcal{A}} s''$, $t \xrightarrow{\delta}_{\mathcal{B}} t'$ and $t \xrightarrow{\delta}_{\mathcal{B}} t''$. By rule R4, we then have $\mathit{traces}_{\mathcal{A}}(s') = \mathit{traces}_{\mathcal{A}}(s'')$ and $\mathit{traces}_{\mathcal{B}}(t') = \mathit{traces}_{\mathcal{B}}(t'')$. To prove that $\mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s',t')) = \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s'',t''))$, we must prove that both $\mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s',t')) \subseteq \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s'',t''))$ and $\mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s'',t'')) \subseteq \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s',t'))$. The latter follows directly from rule R3, so all that's left to show is $\mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s',t')) \subseteq \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s'',t''))$. The proof for this is similar to the proof for rule R3, but using the fact that $\mathit{traces}_{\mathcal{A}}(s') = \mathit{traces}_{\mathcal{A}}(s'')$ and $\mathit{traces}_{\mathcal{B}}(t') = \mathit{traces}_{\mathcal{B}}(t'')$, instead of $\mathit{traces}_{\mathcal{A}}(s') \subseteq \mathit{traces}_{\mathcal{A}}(s)$ and $\mathit{traces}_{\mathcal{B}}(t') \subseteq \mathit{traces}_{\mathcal{B}}(t)$. □

As mentioned in Definition 3.13, we do not allow the hiding of actions to lead to the creation of divergent paths. Assuming this does not occur, QTSs are also closed under the operation of action hiding.

Theorem 3.17. *Well-formed QTSs are closed under action hiding, i.e., given a well-formed QTS $\mathcal{A}$ and a set of labels $H \subseteq L^{\mathrm{O}}_{\mathcal{A}}$ such that there is no cyclic path $\pi = s_0 a_1 s_1 a_2 s_2 \ldots$ in A with $a_i \in (H \cup \{\, \tau \,\})$ for all a_i, $\mathcal{A} \setminus H$ is also a well-formed QTS.*

Proof. Let $\mathcal{A} = \langle\, S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow_{\mathcal{A}} \,\rangle$ be a well-formed QTS and let $H \subseteq L^{\mathrm{O}}$ be a set of outputs. We then have $\mathcal{A} \setminus H = \langle\, S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}} \setminus H, \rightarrow_H \,\rangle$, as defined in Definition 3.13. To prove that well-formed QTSs are closed under action hiding we must show that $\mathcal{A} \setminus H$ is a well-formed QTS, i.e., that it satisfies each of the rules R1, R2, R3 and R4. In the following, we use $\mathit{traces}_H(s)$ to denote the set of all traces of $\mathcal{A} \setminus H$ starting in the state $s \in S$.

1. To prove that $\mathcal{A} \setminus H$ satisfies rule R1, we must show that for all states $s \in S$:

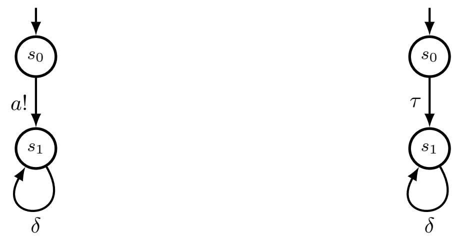
$$\text{if } q(s), \text{ then } s \xrightarrow{\delta}_H$$

Let $s \in S$ be any state such that $q(s)$ holds in $\mathcal{A} \setminus H$. Since hiding of actions effectively relabels output-transitions to internal transitions, it follows that $q(s)$ must also hold in $\mathcal{A}$. By rule R1, we then have $s \xrightarrow{\delta}_{\mathcal{A}}$. Since hiding does not change existing δ-transitions, we then also have $s \xrightarrow{\delta}_H$.

2. To prove that $\mathcal{A} \setminus H$ satisfies rule R2, we must show that for all states $s, s' \in S$:

$$\text{if } s \xrightarrow{\delta}_H s', \text{ then } q(s')$$

Consider any transition $s \xrightarrow{\delta}_H s'$ with $s, s' \in S$. It follows from Definition 3.13 that if $s \xrightarrow{\delta}_H s'$, then $s \xrightarrow{\delta}_{\mathcal{A}} s'$, and by rule R2 it then follows that s' must have been quiescent prior to hiding. Since hiding does not introduce new output-transitions or internal transitions for quiescent states, it follows that s' is still quiescent in $\mathcal{A} \setminus H$.



(a) $\mathcal{A}$ (b) $\mathcal{A} \backslash \{ a \}$

Figure 3.6: Well-formed QTSs are not closed under action hiding if states with internal transitions are considered quiescent.

3. To prove that $\mathcal{A} \setminus H$ satisfies rule R3, we must show that for all states $s, s' \in S$:

$$\text{if } s \xrightarrow{\delta}_H s', \text{ then } \mathit{traces}_H(s') \subseteq \mathit{traces}_H(s)$$

Consider any transition $s \xrightarrow{\delta}_H s'$ with $s, s' \in S$. It follows from Definition 3.13 that if $s \xrightarrow{\delta}_H s'$, then $s \xrightarrow{\delta}_{\mathcal{A}} s'$, and by rule R3 we have $\mathit{traces}_{\mathcal{A}}(s') \subseteq \mathit{traces}_{\mathcal{A}}(s)$. Clearly, $\mathit{traces}(\mathcal{A} \setminus H) = \mathit{traces}(A) \upharpoonright (L \setminus H)$, and therefore $\mathit{traces}_H(s) = \mathit{traces}_{\mathcal{A}}(s) \upharpoonright (L \setminus H)$ and $\mathit{traces}_H(s') = \mathit{traces}_{\mathcal{A}}(s') \upharpoonright (L \setminus H)$. Thus, since $\mathit{traces}_{\mathcal{A}}(s') \subseteq \mathit{traces}_{\mathcal{A}}(s)$, we have $\mathit{traces}_{\mathcal{A}}(s') \upharpoonright (L \backslash H) \subseteq \mathit{traces}_{\mathcal{A}}(s) \upharpoonright (L \backslash H)$, and consequently $\mathit{traces}_H(s') \subseteq \mathit{traces}_H(s)$.

4. To prove that $\mathcal{A} \setminus H$ satisfies rule R4, we must show that for all states $s, s', s'' \in S$:

$$\text{if } s \xrightarrow{\delta}_H s' \text{ and } s' \xrightarrow{\delta}_H s'', \text{ then } \mathit{traces}_H(s') = \mathit{traces}_H(s'')$$

Consider any pair of transitions $s \xrightarrow{\delta}_H s'$ and $s' \xrightarrow{\delta}_H s''$ with $s, s', s'' \in S$. It follows from Definition 3.13 that if $s \xrightarrow{\delta}_H s'$ and $s' \xrightarrow{\delta}_H s'$, then $s \xrightarrow{\delta}_{\mathcal{A}} s'$ and $s' \xrightarrow{\delta}_{\mathcal{A}} s''$; and therefore, by rule R4, $\mathit{traces}_{\mathcal{A}}(s') = \mathit{traces}_{\mathcal{A}}(s'')$. Clearly, $\mathit{traces}(\mathcal{A} \setminus H) = \mathit{traces}(A) \upharpoonright (L \backslash H)$, and therefore $\mathit{traces}_H(s') = \mathit{traces}_{\mathcal{A}}(s') \upharpoonright (L \backslash H)$ and $\mathit{traces}_H(s'') = \mathit{traces}_{\mathcal{A}}(s'') \upharpoonright (L \setminus H)$. Thus, since $\mathit{traces}_{\mathcal{A}}(s') = \mathit{traces}_{\mathcal{A}}(s'')$, we have $\mathit{traces}_{\mathcal{A}}(s') \upharpoonright (L \setminus H) = \mathit{traces}_{\mathcal{A}}(s'') \upharpoonright (L \setminus H)$, and consequently $\mathit{traces}_H(s') = \mathit{traces}_H(s'')$. □

As a side note, earlier we mentioned that quiescent states may not have internal transitions enabled, since otherwise well-formed QTSs would no longer be closed under certain operations. To see that this is indeed the case, consider the QTS $\mathcal{A}$ in Figure 3.6a. Clearly, this QTS is well-formed. In particular, the state s_0 does not have a δ-labelled self-loop, since the output a is enabled in this state. Now, consider what happens if we hide this output a, as shown in Figure 3.6b. Assuming that quiescent states may have internal transitions enabled, the QTS $\mathcal{A} \backslash \{ a \}$ no longer satisfies rule R1: the state s_0 is now quiescent, but does not have an outgoing δ-transition.

3.5.2 Commutativity Properties

Now, we investigate the commutativity of function composition of deltafication with determinisation, action hiding and parallel composition, when applied to QTSs. Here, we are only interested in trace equivalence, since two QTSs behave the same if they have the same sets of traces. Thus, we consider the function compositions of two operations to be commutative

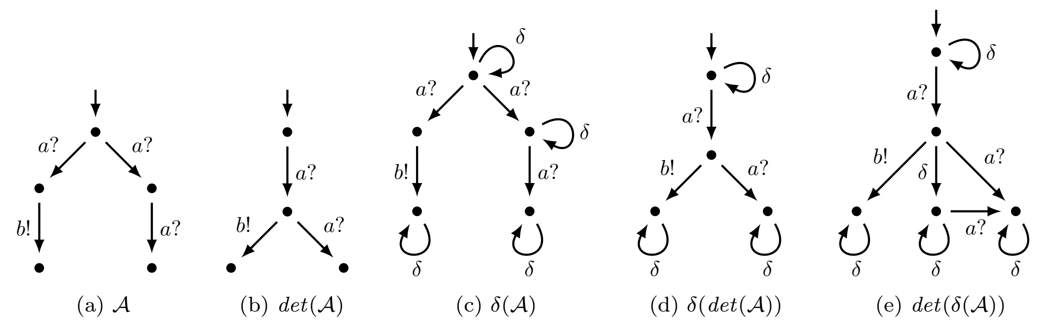

a?
a?
b!
a?
b!
a?
δ
δ
(a) $\mathcal{A}$
(b) $det(\mathcal{A})$
(c) $\delta(\mathcal{A})$
(d) $\delta(det(\mathcal{A}))$
(e) $det(\delta(\mathcal{A}))$

Figure 3.7: The determinisation and deltafication of the IOTS $\mathcal{A}$ do not commute. Note that some a-labelled self-loops have been left out to reduce clutter. Examples taken from [TBS11].

if the end results of applying both operations in either order are trace equivalent. We will show that parallel composition and action hiding can safely be swapped with deltafication, but that determinisation has to precede deltafication to get sensible results.

Proposition 3.18. *Deltafication and determinisation do not commute, i.e., given an IOTS $\mathcal{A}$ such that $\delta \notin L$, it is not necessarily the case that $det(\delta(\mathcal{A})) \approx_{\text{tr}} \delta(det(\mathcal{A}))$.*

Proof. Consider the IOTS $\mathcal{A}$, and its determinisation $det(\mathcal{A})$ and deltafication $\delta(\mathcal{A})$, shown in Figure 3.7. Clearly, the deltafication of the determinisation of $\mathcal{A}$ (i.e., $\delta(det(\mathcal{A}))$), shown in Figure 3.7d, results in an incorrect observation automaton, as it does not model the fact that in the nondeterministic QTS $\delta(\mathcal{A})$ quiescence may be observed after an initial a input, as required by rule R1.

Contrary to the deltafication of the determinisation of $\mathcal{A}$, the determinisation of the deltafication of $\mathcal{A}$ (i.e., $det(\delta(\mathcal{A}))$), which is shown in Figure 3.7e, does preserve the fact that quiescence may be observed after an initial a input. This shouldn't come as a surprise, since for any IOTS $\mathcal{A}$ the determinisation $det(\mathcal{A})$ is trace equivalent to the original automaton [BK08]. □

Thus, when transforming a nondeterministic IOTS $\mathcal{A}$ to a deterministic, well-formed QTS, one should first derive $\delta(\mathcal{A})$ and afterwards determinise.

The following results show that deltafication does commute with both action hiding and parallel composition.

Theorem 3.19. *Deltafication and action hiding commute, i.e., given an IOTS $\mathcal{A}$ such that $\delta \notin L$ and a set of labels $H \subseteq L^{\text{O}}_{\mathcal{A}}$, we have $\delta(\mathcal{A} \setminus H) \approx_{\text{tr}} \delta(\mathcal{A}) \setminus H$.*

Proof. The hiding of actions only results in the relabelling of output transitions to internal transtions, while deltafication only results in the addition of δ-labelled self-loops to states that have no outgoing output or internal transitions. Hence, the two operations work on disjoint sets of states; commutativity is therefore immediate. □

Note that we implicitly assume that the hiding of actions does not lead to the creation of divergent paths.

Theorem 3.20. *Deltafication and parallel composition commute, i.e., given two IOTSs $\mathcal{A}$ and $\mathcal{B}$ such that $\delta \notin L_{\mathcal{A}}$ and $\delta \notin L_{\mathcal{B}}$, and $L^{\text{O}}_{\mathcal{A}} \cap L^{\text{O}}_{\mathcal{B}} = \emptyset$, we have $\delta(\mathcal{A} \parallel \mathcal{B}) \approx_{\text{tr}} \delta(\mathcal{A}) \parallel \delta(\mathcal{B})$.*

Proof. Let $\mathcal{A} = \langle S_{\mathcal{A}}, S^0_{\mathcal{A}}, L^{\mathrm{I}}_{\mathcal{A}}, L^{\mathrm{O}}_{\mathcal{A}}, \rightarrow_{\mathcal{A}} \rangle$ and $\mathcal{B} = \langle S_{\mathcal{B}}, S^0_{\mathcal{B}}, L^{\mathrm{I}}_{\mathcal{B}}, L^{\mathrm{O}}_{\mathcal{B}}, \rightarrow_{\mathcal{B}} \rangle$ be two IOTSs such that $\delta \notin L_{\mathcal{A}}$, $\delta \notin L_{\mathcal{B}}$, and $L^{\mathrm{O}}_{\mathcal{A}} \cap L^{\mathrm{O}}_{\mathcal{B}} = \emptyset$, and let $\delta(\mathcal{A} \parallel \mathcal{B}) = \langle S_{\mathcal{C}}, S^0_{\mathcal{C}}, L^{\mathrm{I}}_{\mathcal{C}}, L^{\mathrm{O}}_{\mathcal{C}}, \rightarrow_{\mathcal{C}} \rangle$ and $\delta(\mathcal{A}) \parallel \delta(\mathcal{B}) = \langle S_{\mathcal{D}}, S^0_{\mathcal{D}}, L^{\mathrm{I}}_{\mathcal{D}}, L^{\mathrm{O}}_{\mathcal{D}}, \rightarrow_{\mathcal{D}} \rangle$, as defined by Definition 3.8 and Definition 3.11. We have $S_{\mathcal{C}} = S_{\mathcal{D}} = S_{\mathcal{A}} \times S_{\mathcal{B}}$ and $S^0_{\mathcal{C}} = S^0_{\mathcal{D}} = S^0_{\mathcal{A}} \times S^0_{\mathcal{B}}$, since deltafication does not change any states; we also have $L_{\mathcal{C}} = L_{\mathcal{D}} = L_{\mathcal{A}} \cup L_{\mathcal{B}}$. To prove that $\delta(\mathcal{A} \parallel \mathcal{B}) \approx_{\mathrm{tr}} \delta(\mathcal{A}) \parallel \delta(\mathcal{B})$, we will prove a stronger property: we will show a bijection exists between the two. Clearly, two automata that are isomorphic are also trace equivalent. Hence, we will show that there exists a bijection $h\colon S_{\mathcal{C}} \to S_{\mathcal{D}}$ such that the following holds:

1. for all $s_0 \in S^0_{\mathcal{C}}$ there exists a $t_0 \in S^0_{\mathcal{D}}$ such that $h(s_0) = t_0$, and vice versa;
2. $s \xrightarrow{a}_{\mathcal{C}} s'$ if and only if $h(s) \xrightarrow{a}_{\mathcal{D}} h(s')$, for all $s, s' \in S_{\mathcal{C}}$ and $a \in L_{\mathcal{C}} \cup \{\, \delta, \tau \,\}$.

Since $S_C = S_D$, we let h be the identity function, i.e., for all $s \in S_{\mathcal{C}}$ we have $h(s) = s$. Clearly, h is a bijection, and since $S^0_{\mathcal{C}} = S^0_{\mathcal{D}}$, for all $s_0 \in S^0_{\mathcal{C}}$ there exists a $t_0 \in S^0_{\mathcal{D}}$ such that $h(s_0) = t_0$, namely $t_0 = s_0$; and symmetrically for all $t_0 \in S^0_{\mathcal{D}}$. Since h is the identity function, to prove that $s \xrightarrow{a}_{\mathcal{C}} s'$ if and only if $h(s) \xrightarrow{a}_{\mathcal{D}} h(s')$, we must show that $s \xrightarrow{a}_{\mathcal{C}} s'$ if and only if $s \xrightarrow{a}_{\mathcal{D}} s'$, i.e., if $s \xrightarrow{a}_{\mathcal{C}} s'$, then $s \xrightarrow{a}_{\mathcal{D}} s'$, and if $s \xrightarrow{a}_{\mathcal{D}} s'$, then $s \xrightarrow{a}_{\mathcal{C}} s'$. We will only prove the former case, the proof for the latter case is largely symmetrical. We look at the cases (1) $a = \tau$; (2) $a = \delta$; (3) $a \in L^{\mathrm{I}}_{\mathcal{C}}$; and (4) $a \in L^{\mathrm{O}}_{\mathcal{C}}$, separately. For the various proofs, let $s = (u, v)$ and $s' = (u', v')$ with $u, u' \in S_{\mathcal{A}}$, and $v, v' \in S_{\mathcal{B}}$.

1. Assume $a = \tau$, i.e., $(u, v) \xrightarrow{\tau}_{\mathcal{C}} (u', v')$. In this case, since deltafication does not affect or introduce τ-transitions, we have, by Definition 3.11, either $u \xrightarrow{\tau}_{\mathcal{A}} u'$ and $v = v'$, or $v \xrightarrow{\tau}_{\mathcal{B}} v'$ and $u = u'$. In both cases, these transitions will still exist after the deltafication of $\mathcal{A}$ and $\mathcal{B}$, respectively. Thus, it follows directly from Definition 3.11 that also $(u, v) \xrightarrow{\tau}_{\mathcal{D}} (u', v')$.

2. If $a = \delta$, i.e., $(u, v) \xrightarrow{\delta}_{\mathcal{C}} (u', v')$, then the δ-transition was added by the deltafication of $\mathcal{A} \parallel \mathcal{B}$, and $(u, v) = (u', v')$. By Definition 3.8, the state (u, v) must be quiescent in $\mathcal{A} \parallel \mathcal{B}$. Since $\mathcal{A}$ and $\mathcal{B}$ are input-enabled, we can conclude from Definition 3.11 that both u and v must also be quiescent. Hence, after deltafication of $\mathcal{A}$ and $\mathcal{B}$, both u and v, respectively, will have δ-labelled self-loops. Consequently, by Definition 3.11, $(u, v) \xrightarrow{\delta}_{\mathcal{D}} (u', v')$.

3. Assume $a \in L^{\mathrm{I}}_{\mathcal{C}}$, i.e., $(u, v) \xrightarrow{a}_{\mathcal{C}} (u', v')$. Deltafication does not affect or introduce input-labelled transitions, nor does it affect $L_{\mathcal{A}}$ or $L_{\mathcal{B}}$, so it follows from Definition 3.11 that there are three possibilities:

   (a) $u \xrightarrow{a}_{\mathcal{A}} u'$ and $v \xrightarrow{a}_{\mathcal{B}} v'$.
   (b) $u \xrightarrow{a}_{\mathcal{A}} u'$, $v = v'$ and $a \notin L_{\mathcal{B}}$.
   (c) $v \xrightarrow{a}_{\mathcal{B}} v'$, $u = u'$ and $a \notin L_{\mathcal{A}}$.

   In all cases, these transitions will still exist after the deltafication of $\mathcal{A}$ and $\mathcal{B}$. Neither will $L_{\mathcal{A}}$ or $L_{\mathcal{B}}$ have been changed. Thus, it follows directly from Definition 3.11 that also $(u, v) \xrightarrow{a}_{\mathcal{D}} (u', v')$.

4. Finally, assume $a \in L^{\mathrm{O}}_{\mathcal{C}}$, i.e., $(u, v) \xrightarrow{a}_{\mathcal{C}} (u', v')$. Deltafication does not affect or introduce output-labelled transitions, nor does it affect $L_{\mathcal{A}}$ or $L_{\mathcal{B}}$, so it follows from Definition 3.11 that there are four possibilities:

   (a) $u \xrightarrow{a}_{\mathcal{A}} u'$, $v \xrightarrow{a}_{\mathcal{B}} v'$ and $a \in L^{\mathrm{O}}_{\mathcal{A}}$, $a \in L^{\mathrm{I}}_{\mathcal{B}}$.

(b) $u \xrightarrow{a}_{\mathcal{A}} u'$, $v \xrightarrow{a}_{\mathcal{B}} v'$ and $a \in L^{\mathrm{I}}_{\mathcal{A}}$, $a \in L^{\mathrm{O}}_{\mathcal{B}}$.

(c) $u \xrightarrow{a}_{\mathcal{A}} u'$, $v = v'$ and $a \notin L_{\mathcal{B}}$.

(d) $v \xrightarrow{a}_{\mathcal{B}} v'$, $u = u'$ and $a \notin L_{\mathcal{A}}$.

In all cases, these transitions will still exist after the deltafication of $\mathcal{A}$ and $\mathcal{B}$. Neither will $L_{\mathcal{A}}$ or $L_{\mathcal{B}}$ have been changed. Thus, it follows directly from Definition 3.11 that also $(u, v) \xrightarrow{a}_{\mathcal{D}} (u', v')$. □

The above results are very important, as they allow great modelling flexibility. In practice, after all, hiding and parallel composition are often already applied to the IOTSs that describe a specification and its implementation. We now showed that this yields the same well-formed QTSs as in the case these operations are applied after deltafication.

3.6 Discussion of the Well-formedness Rules

In this section, we explain why we have defined the well-formedness rules, and what alternative rules we have considered (and ultimately rejected).

3.6.1 Rationale behind the Well-formedness Rules

As mentioned before, QTSs are similar to the Suspension Automata (SAs) introduced in Section 2.5. Since SAs are derived from existing IOTSs, and we assume that these IOTSs correctly capture the behaviour of the systems they model, we find that SAs are 'well-formed' in the sense that their observable behaviour corresponds to that of realistic specifications or implementations. Since we also desire this property to hold for well-formed QTSs, which may be built from scratch rather than derived from existing IOTSs, the rules R1, R2, R3 and R4 have been carefully crafted to be the minimal set of rules such that the following conditions are satisfied:

C1: For every SA $\mathcal{S}$ there exists a well-formed QTS $\mathcal{Q}$ such that $\mathcal{S} \approx_{\mathrm{tr}} \mathcal{Q}$.

C2: For every well-formed QTS $\mathcal{Q}$ there exists a SA $\mathcal{S}$ such that $\mathcal{Q} \approx_{\mathrm{tr}} \mathcal{S}$.

By ensuring that well-formed QTSs satisfy conditions C1 and C2, we can be sure that QTSs always conform to realistic specifications or implementations in terms of expressible behaviour. Furthermore, as discussed in Section 2.5, SAs are used as the basis for several model-based testing frameworks such as `ioco`. Hence, an additional benefit of satisfying the conditions C1 and C2 is that well-formed QTSs can be used as a drop-in replacement for SAs in these frameworks, since the two models are equally expressive.

The following two theorems prove that well-formed QTSs satisfy conditions C1 and C2, hence well-formed QTSs and SAs are equivalent in terms of expressible behaviour. Recall from Section 2.5 that a SA is constructed by taking an existing IOTS and adding δ-labelled self-loops to all quiescent states (similar to our deltafication procedure for constructing QTSs from IOTSs), and afterwards applying the determinisation operation.

Theorem 3.21. *For every SA $\mathcal{S}$ there exists a well-formed QTS $\mathcal{Q}$ such that $\mathcal{S} \approx_{\mathrm{tr}} \mathcal{Q}$* **(C1).**

Proof. Let $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow_{\mathcal{A}} \rangle$ be an IOTS, and $\mathcal{S}$ the corresponding SA. Hence, $\mathcal{S}$ is the determinisation of the IOTS $\mathcal{A}' = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, \rightarrow'_{\mathcal{A}} \rangle$, where $\rightarrow'_{\mathcal{A}}$ is defined as follows:

$$\rightarrow'_{\mathcal{A}} = \rightarrow_{\mathcal{A}} \cup \{ (s, \delta, s) \in S \times \{ \delta \} \times S \mid q(s) \text{ holds in } \mathcal{A} \}$$

Now, observe that $\mathcal{A}'$ was obtained from $\mathcal{A}$ by adding δ-labelled self-loops to all quiescent states, which is exactly the same as the deltafication procedure for creating QTSs from IOTSs. Hence, $\mathcal{A}'$ and $\delta(\mathcal{A})$ are also isomorphic, and consequently they are trace-equivalent. Furthermore, by Theorem 3.10, $\delta(\mathcal{A})$ is a well-formed QTS. Since $\mathcal{S}$ is obtained by determinising $\mathcal{A}'$, we find that $\mathcal{S}$ is also trace-equivalent to $\delta(\mathcal{A})$, since determinisation preserves traces [BK08]. □

Theorem 3.22. *For every well-formed QTS $\mathcal{Q}$ there exists a SA $\mathcal{S}$ such that $\mathcal{Q} \approx_{\text{tr}} \mathcal{S}$* **(C2).**

Proof. Let $\mathcal{Q} = \langle S, S^0, L^{\text{I}}, L^{\text{O}}, \rightarrow_{\mathcal{Q}} \rangle$ be a well-formed QTS. Without loss of generality, we assume the following two properties of $\mathcal{Q}$:

1. $\mathcal{Q}$ does not contain any path of the form $s \xrightarrow{\delta}_{\mathcal{Q}} t \xrightarrow{\delta}_{\mathcal{Q}} u$ with $t, u \in S$ and $t \neq u$. This can be assumed, since rule R4 prescribes that in such a case the traces of t and u should coincide. Therefore, they can be merged to remove the unwanted path fragment, without changing the traces of $\mathcal{Q}$.

2. $\mathcal{Q}$ is deterministic. This can be assumed, since determinisation preserves traces [BK08].

Note that the first assumption implies that there are no cycles in $\mathcal{Q}$ consisting solely of δ-transitions, except for self-loops.

Since SAs cannot be built from scratch, but only arise implicitly by adding δ-transitions to IOTSs, we construct an IOTS $\mathcal{A}$ such that the SA $\mathcal{S}$ obtained from $\mathcal{A}$ is trace-equivalent to the QTS $\mathcal{Q}$. Now, let $\mathcal{A} = \langle S, S^0, L^{\text{I}}, L^{\text{O}}, \rightarrow_{\mathcal{A}} \rangle$ be an IOTS, where $\rightarrow_{\mathcal{A}}$ is defined as follows:

$$\begin{aligned} \rightarrow_{\mathcal{A}} \; = \; & \{ (s, a, t) \in \rightarrow_{\mathcal{Q}} \mid a \neq \delta \} \\ \cup \; & \{ (s, \tau, t) \in S \times \{ \tau \} \times S \mid (s, \delta, t) \in \rightarrow_{\mathcal{Q}} \wedge s \neq t \} \end{aligned}$$

Note that, by assumption (1), indeed $\rightarrow_{\mathcal{Q}} \subseteq S \times (L^{\text{I}} \cup L^{\text{O}} \cup \{ \tau \}) \times S$, and hence we have defined a proper IOTS. The corresponding SA $\mathcal{S}$ is the determinisation of the IOTS $\mathcal{A}' = \langle S, S^0, L^{\text{I}}, L^{\text{O}}, \rightarrow'_{\mathcal{A}} \rangle$, where $\rightarrow'_{\mathcal{A}}$ is defined by

$$\rightarrow'_{\mathcal{A}} = \rightarrow_{\mathcal{A}} \cup \{ (s, \delta, s) \in S \times \{ \delta \} \times S \mid q(s) \text{ holds in } \mathcal{A} \}$$

Since, as mentioned before, determinisation preserves traces, we will only show that $\mathcal{A}'$ is trace-equivalent to $\mathcal{Q}$. It then follows immediately that the SA S is also trace-equivalent to $\mathcal{Q}$. Hence, we need to show that $\mathit{traces}(\mathcal{Q}) = \mathit{traces}(\mathcal{A}')$, i.e., that both $\mathit{traces}(\mathcal{Q}) \subseteq \mathit{traces}(\mathcal{A}')$ and $\mathit{traces}(\mathcal{A}') \subseteq \mathit{traces}(\mathcal{Q})$. We will first prove the former, then the latter.

1. First, we prove that $\mathit{traces}(\mathcal{Q}) \subseteq \mathit{traces}(\mathcal{A}')$. Let $\sigma \in \mathit{traces}(\mathcal{Q})$. We must prove that also $\sigma \in \mathit{traces}(\mathcal{A}')$. If $\sigma \in \mathit{traces}(\mathcal{Q})$, there exists a path $\pi = s_0\, a_1\, s_1\, a_2\, s_2\, \ldots\, a_n\, s_n$ in $\mathcal{Q}$ such that $\mathit{trace}(\pi) = \sigma$, $s_i \in S$, $a_i \in L \cup \{ \tau, \delta \}$, and $s_0 \in S^0$. By backwards induction on the length of π, we show for every suffix $\pi' = s_k\, a_{k+1}\, s_{k+1}\, \ldots\, a_n\, s_n$ of π that $\mathit{trace}(\pi') \in \mathit{traces}_{\mathcal{A}'}(s_k)$. This then implies that for $\sigma = \mathit{trace}(\pi)$ we have $\sigma \in \mathit{traces}_{\mathcal{A}'}(s_0)$, and since $\mathit{traces}_{\mathcal{A}'}(s_0) = \mathit{traces}(\mathcal{A}')$, we have then proven that $\sigma \in \mathit{traces}(\mathcal{A}')$.

Base case. For $k = n$, we have $\pi' = s_n$ and hence $\mathit{trace}(\pi') = \epsilon$. In this case, we obviously have $\mathit{trace}(\pi') \in \mathit{traces}_{\mathcal{A}'}(s_n)$.

Inductive case. Assume $\mathit{trace}(\pi'') \in \mathit{traces}_{\mathcal{A}'}(s_{k+1})$ for the path $\pi'' = s_{k+1}\, a_{k+2}\, s_{k+2} \, \ldots \, a_n\, s_n$. We now must show that $\mathit{trace}(\pi') \in \mathit{traces}_{\mathcal{A}'}(s_k)$ for $\pi' = s_k\, a_{k+1}\, s_{k+1}\, a_{k+2}\, s_{k+2} \, \ldots \, a_n\, s_n$. Note that $\mathit{trace}(\pi') = a_{k+1} \cdot \mathit{trace}(\pi'')$, since there are no τ-transitions in $\mathcal{Q}$, which follows from the second assumption made above on the structure of $\mathcal{Q}$. We make a case distinction based on whether (1) $a_{k+1} \neq \delta$, (2) $a_{k+1} = \delta$ and $s_k = s_{k+1}$, and (3) $a_{k+1} = \delta$ and $s_k \neq s_{k+1}$.

(a) If $a_{k+1} \neq \delta$, then by definition of $\mathcal{A}$ and $\mathcal{A}'$ we have $s_k \xrightarrow{a_{k+1}}_{\mathcal{A}'} s_{k+1}$ in $\mathcal{A}'$. Hence, since $\pi'' \in \mathit{traces}_{\mathcal{A}'}(s_{k+1})$, it immediately follows that $\pi' \in \mathit{traces}_{\mathcal{A}'}(s_k)$.

(b) If $a_{k+1} = \delta$ and $s_k = s_{k+1}$, then it follows from rule R2 that s_k is quiescent in $\mathcal{Q}$. Furthermore, by the assumption that $\mathcal{Q}$ is deterministic, there cannot exist any other outgoing δ-transitions from s_k in $\mathcal{Q}$, and therefore no τ-transitions are added to s_k in the construction of $\mathcal{A}$. Consequently, s_k is also quiescent in $\mathcal{A}$, and hence we find that indeed $s_k \xrightarrow{\delta}_{\mathcal{A}'} s_{k+1}$ in $\mathcal{A}'$, by definition of $\mathcal{A}'$. Hence, since $\pi'' \in \mathit{traces}_{\mathcal{A}'}(s_{k+1})$, it immediately follows that $\pi' \in \mathit{traces}_{\mathcal{A}'}(s_k)$.

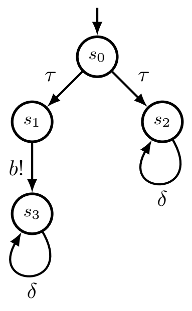
(c) If $a_{k+1} = \delta$ and $s_k \neq s_{k+1}$, then due to rule R2 we find that s_{k+1} is quiescent, and it follows from rule R1 that s_{k+1} must have an outgoing δ-transition. By the assumption that no path fragment of the form $s \xrightarrow{\delta}_{\mathcal{Q}} t \xrightarrow{\delta}_{\mathcal{Q}} u$ with $t, u \in S$ and $t \neq u$ is present in $\mathcal{Q}$, this implies that $s_{k+1} \xrightarrow{\delta}_{\mathcal{Q}} s_{k+1}$. It then follows by definition of $\mathcal{A}'$ that there a is no τ-transition added to s_{k+1} in the construction of $\mathcal{A}$, and therefore s_{k+1} is also quiescent in $\mathcal{A}$. Hence, we have $s_{k+1} \xrightarrow{\delta}_{\mathcal{A}'} s_{k+1}$. Also, since $s_k \xrightarrow{\delta}_{\mathcal{Q}} s_{k+1}$, we can conclude by the definitions of $\mathcal{A}$ and $\mathcal{A}'$ that $s_k \xrightarrow{\tau}_{\mathcal{A}'} s_{k+1}$. Consequently, in $\mathcal{A}'$ there exists a path $s_k \xrightarrow{\tau}_{\mathcal{A}'} s_{k+1} \xrightarrow{\delta}_{\mathcal{A}'} s_{k+1}$ and therefore a trace δ from s_k to s_{k+1} . Thus, since $\pi'' \in \mathit{traces}_{\mathcal{A}'}(s_{k+1})$, it immediately follows that $\pi' \in \mathit{traces}_{\mathcal{A}'}(s_k)$.s

2. Next, we prove that $\mathit{traces}(\mathcal{A}') \subseteq \mathit{traces}(\mathcal{Q})$. Let $\sigma \in \mathit{traces}(\mathcal{A}')$. We must prove that also $\sigma \in \mathit{traces}(\mathcal{Q})$. If $\sigma \in \mathit{traces}(\mathcal{A}')$, there exists a path $\pi = s_0\, a_1\, s_1\, a_2\, s_2 \, \ldots \, a_n\, s_n$ in $\mathcal{A}'$ such that $\mathit{trace}(\pi) = \sigma$, $s_i \in S$, $a_i \in L \cup \{\tau, \delta\}$, and $s_0 \in S^0$. By backwards induction on the length of π, we show for every suffix $\pi' = s_k\, a_{k+1}\, s_{k+1} \, \ldots \, a_n\, s_n$ of π that $\mathit{trace}(\pi') \in \mathit{traces}_{\mathcal{Q}}(s_k)$. This then implies that for $\sigma = \mathit{trace}(\pi)$ we have $\sigma \in \mathit{traces}_{\mathcal{Q}}(s_0)$, and since $\mathit{traces}_{\mathcal{Q}}(s_0) = \mathit{traces}(\mathcal{Q})$, we have then proven that $\sigma \in \mathit{traces}(\mathcal{Q})$.

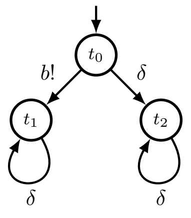
Base case. For $k = n$, we have $\pi' = s_n$ and hence $\mathit{trace}(\pi') = \epsilon$. In this case, we obviously have $\mathit{trace}(\pi') \in \mathit{traces}_{\mathcal{Q}}(s_n)$.

Inductive case. Assume $\mathit{trace}(\pi'') \in \mathit{traces}_{\mathcal{Q}}(s_{k+1})$ for the path $\pi'' = s_{k+1}\, a_{k+2}\, s_{k+2} \, \ldots \, a_n\, s_n$. We now must show that $\mathit{trace}(\pi') \in \mathit{traces}_{\mathcal{Q}}(s_k)$ for $\pi' = s_k\, a_{k+1}\, s_{k+1}\, a_{k+2}\, s_{k+2} \, \ldots \, a_n\, s_n$. Note that $\pi' = a_{k+1} \cdot \pi''$ if $a_{k+1} \neq \tau$ and $\pi' = \pi''$ if $a_{k+1} = \tau$. We make a case distinction based on whether (1) $a_{k+1} \neq \delta$ and $a_{k+1} \neq \tau$, (2) $a_{k+1} = \delta$, (3) $a_{k+1} = \tau$ and $s_k \xrightarrow{\tau}_{\mathcal{Q}} s_{k+1}$, and (4) $a_{k+1} = \tau$ and $s_k \xrightarrow{\delta}_{\mathcal{Q}} s_{k+1}$.

(1) If $a_{k+1} \neq \delta$ and $a_{k+1} \neq \tau$, then we can conclude from the definitions of $\mathcal{A}$ and $\mathcal{A}'$ that $s_k \xrightarrow{a_{k+1}}_{\mathcal{Q}} s_{k+1}$. Hence, since $\pi'' \in \mathit{traces}_{\mathcal{Q}}(s_{k+1})$, it immediately follows that $\pi' \in \mathit{traces}_{\mathcal{Q}}(s_k)$.



(a) $\mathcal{A}$



(b) $det(\mathcal{A})$

Figure 3.8: A well-formed QTS $\mathcal{A}$ and its determinisation $det(\mathcal{A})$.

(2) If $a_{k+1} = \delta$, then it follows from the definitions of $\mathcal{A}$ and $\mathcal{A}'$ that it must have been added during the construction of $\mathcal{A}'$ (and hence it follows that $s_{k+1} = s_k$), since s_k was quiescent in $\mathcal{A}$. Therefore, s_k is also quiescent in $\mathcal{Q}$ (since $\mathcal{Q}$ cannot have more output transitions or τ-transitions than $\mathcal{A}$), and consequently $s_k \xrightarrow{\delta}_{\mathcal{Q}} s_k$ by rule R1. Thus, since $\pi'' \in traces_{\mathcal{Q}}(s_{k+1})$ and $s_{k+1} = s_k$, it immediately follows that $\pi' \in traces_{\mathcal{Q}}(s_k)$.

(3 and 4) If $a_{k+1} = \tau$, then $\pi' = \pi''$. If this transition was added due to the presence of the transition $s_k \xrightarrow{\tau}_{\mathcal{Q}} s_{k+1}$, then, since $\pi'' \in traces_{\mathcal{Q}}(s_{k+1})$, it immediately follows that $\pi' \in traces_{\mathcal{Q}}(s_k)$. Otherwise, if this transition was added due to the transition $s_k \xrightarrow{\delta}_{\mathcal{Q}} s_{k+1}$, then from rule R3 it follows that $traces_{\mathcal{Q}}(s_{k+1}) \subseteq traces_{\mathcal{Q}}(s_k)$. Thus, since $\pi'' \in traces_{\mathcal{Q}}(s_{k+1})$, this implies that $\pi' \in traces_{\mathcal{Q}}(s_k)$. □

3.6.2 Alternatives to the Well-formedness Rules

In this section, we discuss several alternatives we have considered for some of the well-formedness rules, and explain why these alternatives were not used. As remarked in Section 3.2, four conditions for valid SAs are given in [Wil07] that coincide exactly with the rules R1, R2, R3 and R4. These conditions are phrased in a unnecessarily complex manner, however, hence we prefer to use our own rules.

Alternative for rule R2

$$\text{if } s \xrightarrow{\delta} s', \text{ then } q(s)$$

This alternative rule may no longer hold after the determinisation of a well-formed QTS. Hence, when using this rule instead of the original rule R2, well-formed QTSs are no longer closed under determinisation. To see this, consider the non-deterministic well-formed QTS $\mathcal{A}$ in Figure 3.8a and its determinisation $det(\mathcal{A})$ in Figure 3.8b. Clearly, the alternative rule holds for $\mathcal{A}$, since s_2 and s_3 are the only states with outgoing δ-transitions and are both quiescent. However, the rule does not apply to $det(\mathcal{A})$, since $t_0 \xrightarrow{\delta} t_2$, but t_0 has an outgoing output-transition enabled and is therefore not quiescent.

Alternative for rule R3

$$\text{if } s \xrightarrow{\delta} s', \text{ then } traces(s') = traces(s)$$

When using this rule instead of the original rule R3, well-formed QTSs are no longer closed under determinisation. To see this, consider the non-deterministic well-formed QTS $\mathcal{A}$ in Figure 3.8a and its determinisation $det(\mathcal{A})$ in Figure 3.8b. Clearly, the alternative rule holds for $\mathcal{A}$, since there are only δ-labelled self-loops present. However, the rule does not apply to $det(\mathcal{A})$, since $t_0 \xrightarrow{\delta} t_2$, but $b \in traces(t_0)$ and $b \notin traces(t_2)$.

3.7 Summary

In this chapter, we have introduced Quiescent Transition Systems (QTSs), which are a specialisation of IOTSs that capture the notion of quiescence. QTSs accomplish this by explicitly marking quiescent states with a special δ-transition, similar to SAs. Furthermore, we have defined what it means for a QTS to be well-formed, by imposing four constraints on the occurrence of δ-transitions. Following that, we have introduced the deltafication operation, using which an IOTS can be transformed into an equivalent QTS that captures all possible behaviour of it, including quiescence.

The familiar IOTS operations of determinisation, parallel composition and action hiding have also been defined for QTS, and several interesting closure and commutativity properties regarding these operations have been investigated. In particular, we have shown that QTSs are closed under determinisation, parallel composition and action hiding. Finally, we have further discussed the rationale behind the well-formedness rules, by comparing QTSs with SAs, and discussed some alternatives for some of these rules. Hence, QTSs address most of the shortcomings of SAs laid out in Section 2.5, except for the convergence requirement. This requirement will be lifted by Divergent Quiescent Transition Systems (DQTSs), which will be introduced in the next chapter.

Chapter 4

Divergent Quiescent Transition Systems

One major limitation of QTSs is that they do not allow divergent paths, i.e, infinite paths consisting of only internal transitions, to occur. However, divergent paths do occur in practice, typically in the form of τ-loops that appear after hiding output actions. To address this issue, we introduce *Divergent Quiescent Transition Systems* (DQTSs) in this chapter. DQTSs are a variant on regular QTSs that support state-recurrent (see Definition 4.4) divergence, thus allowing more specification freedom.

Rather than extending IOTSs like QTSs, DQTSs extend IOAs (see Section 2.4) with support for quiescence. Hence, like QTSs, DQTSs distinguish between input actions, output actions, and internal actions, but there may be more than one internal action. Again, we use the special output label δ to indicate the occurrence of quiescence. The reason that DQTSs are based on IOAs rather than IOTSs is that the fairness (and task partition) notions of IOAs can be used to resolve the ambiguity arising when combining divergence and quiescence, as will be discussed in the next section.

We also introduce the familiar operations of determinisation and parallel composition for DQTSs. The operation of action hiding is also supported by the DQTS formalism, but is necessarily more complicated than the action hiding operation for IOAs, as we will discuss in Section 4.4. For all these operations we investigate several important closure and commutativity aspects, and show that DQTSs, like QTSs, exhibit desirable compositionality properties.

4.1 The DQTS Model

DQTSs form a extension of regular IOAs in which the observation of quiescence is explicitly represented using special δ-transitions.

Definition 4.1 (Divergent Quiescent Transition Systems)**.** A *Divergent Quiescent Transition System* (DQTS) is a tuple $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow \rangle$, where:

- S is a non-empty set of states;
- $S^0 \subseteq S$ is a non-empty set of initial states;
- L^{I}, L^{O} and L^{H} are pairwise disjoint sets of input, output and internal labels, respectively. Furthermore, $\delta \notin (L^{\mathrm{I}} \cup L^{\mathrm{O}} \cup L^{\mathrm{H}})$ is a special output label that is used to denote the observation of quiescence;

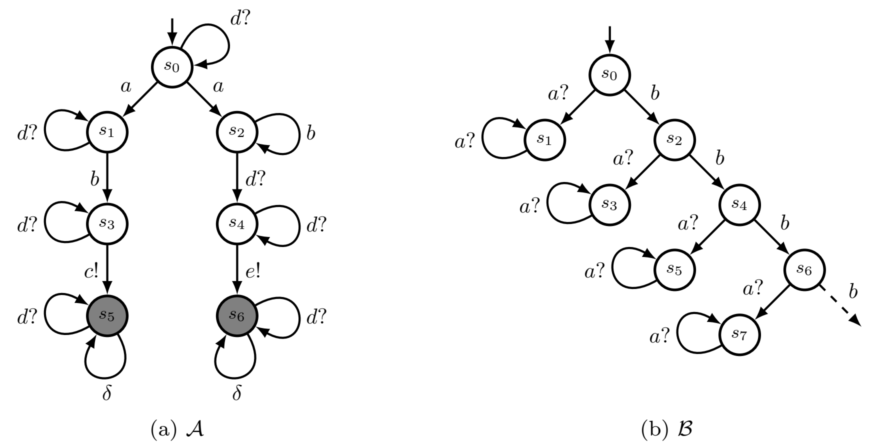


Figure 4.1: Visual representation of DQTSs $\mathcal{A}$ and $\mathcal{B}$. Quiescent states are marked gray.

- P is a partition of $L^{\mathrm{O}} \cup L^{\mathrm{H}}$, i.e., the locally controlled labels;
- $\rightarrow\ \subseteq S \times (L^{\mathrm{I}} \cup L^{\mathrm{O}} \cup L^{\mathrm{H}} \cup \{\delta\}) \times S$ is the transition relation.

As for IOAs, we define $L = L^{\mathrm{I}} \cup L^{\mathrm{O}} \cup L^{\mathrm{H}}$.

Similar to regular IOAs, DQTSs must be input-enabled. All other definitions introduced in Section 2.4 for IOAs also apply to DQTSs. The concept of quiescent states, introduced in Section 3.1 for QTSs, also applies to DQTSs. However, since IOAs (and therefore DQTSs) can have multiple internal actions rather than just τ, the definition for a quiescent state in a DQTS is slightly different.

Definition 4.2 (Quiescent states in IOAs or DQTSs)**.** Let $\mathcal{A}$ be an IOA or DQTS. A state $s \in S$ is *quiescent*, denoted $q(s)$, if it has no locally controlled actions enabled, i.e., $q(s)$ if for all $a \in L^{\mathrm{O}} \cup L^{\mathrm{H}}$ we have $s \not\xrightarrow{a}$. The set of all quiescent states of $\mathcal{A}$ is denoted $q(\mathcal{A})$.

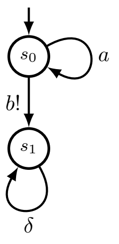
Example 4.3. Figure 4.1a visualises a DQTS $\mathcal{A}$. Recall that internal actions are labelled without suffixes, hence a and b are internal actions. Consequently, only the states s_5 and s_6 are quiescent. □

In contrast to QTSs, we do allow divergent paths to occur in DQTSs. However, there is one restriction: all infinite paths in a DQTS, or in an IOA that is to be converted to a DQTS, must be *state-finite*. This restriction serves to ensure that the deltafication of an IOA, which will be discussed in Section 4.3, is always defined, and can be algorithmically computed.

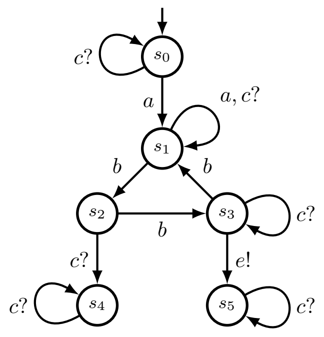
Definition 4.4 (State-finite path)**.** Let $\mathcal{A}$ be an IOA or DQTS and let $\pi \in \mathit{paths}(\mathcal{A})$ be an infinite path in $\mathcal{A}$. If $|\mathit{states}(\pi)| < \infty$, then π is *state-finite*.

Hence, divergent paths are allowed in DQTSs if they are also state-finite.

Example 4.5. Consider again the DQTS $\mathcal{A}$ in Figure 4.1a. Clearly, the infinite divergent path $\pi_1 = s_0\, a\, s_2\, b\, s_2\, b\, s_2\, \ldots$ is state-finite, since $|\mathit{states}(\pi_1)| = |\{s_0, s_2\}| = 2$. On the other hand, the infinite divergent path $\pi_2 = s_0\, b\, s_2\, b\, s_4\, b\, s_6\, b\, \ldots$ of the DQTS $\mathcal{B}$, shown in Figure 4.1b, is not state-finite, since $|\mathit{states}(\pi_2)| = \infty$. □



(a) $\mathcal{A}$



(b) $\mathcal{B}$

Figure 4.2: Visual representation of two divergent DQTSs $\mathcal{A}$ and $\mathcal{B}$.

Since finite-state IOAs or DQTSs cannot contain paths with an infinite number of states, the state-finite path restriction is not a severe one. Furthermore, note that every infinite state-finite path is cyclic.

The occurrence of (state-finite) divergent paths in IOAs or DQTSs may result in the observation of quiescence in states that are not necessarily quiescent. Consider the DQTS $\mathcal{A}$ in Figure 4.2a. Clearly, the state s_0 is not quiescent, since the output transition b are enabled in this state. Nevertheless, if the divergent path $\pi = s_0\, a\, s_0\, a \ldots$ is executed, the output b will never be observed. Hence, the observation of quiescence seems like a possibility in state s_0, even though it is not quiescent. The question then remains whether the system indeed has a possible execution that corresponds to the path π, or not.

Recall that in Section 2.4.1 and Definition 2.32 we introduced the notion of fairness for IOAs, and explicitly assumed that unfair paths do not occur. Hence, the path π defined above only will be executed, and hence the observation of quiescence can only occur, if π is fair. For example, if $P_{\mathcal{A}} = \{\{a\},\{b\}\}$, the path π is not fair and there will be no observation of quiescence. On the other hand, if $P_{\mathcal{A}} = \{\{a, b\}\}$, then an observation of quiescence can be made in state s_0.

Thus, apart from the presence of quiescent states, quiescence can also be observed when paths exist that are both divergent and fair. We call such paths *fairly divergent.*

Definition 4.6 (Fairly divergent path)**.** Given an IOA or DQTS $\mathcal{A}$, an infinite path $\pi \in$ *paths*$(\mathcal{A})$ is *fairly divergent* if π is both fair and divergent, i.e., if $\pi \in$ *fpaths*$(\mathcal{A}) \cap$ *dpaths*$(\mathcal{A})$. The set of all fairly divergent paths of $\mathcal{A}$ is denoted *fdpaths*$(\mathcal{A})$.

Example 4.7. As explained above, the path $\pi = s_0\, a\, s_0\, a \ldots$ in DQTS $\mathcal{A}$, shown in Figure 4.2a, is fairly divergent for $P_{\mathcal{A}} = \{\{a, b\}\}$, but not for $P_{\mathcal{A}} = \{\{a\},\{b\}\}$. □

Since quiescence may be observed for fairly divergence paths, we want to mark these possible observations with δ-transitions. When the system is executing a fairly divergent path, it is continually looping through a finite number (since the path must be state-finite) of states on this path. Since the path is infinite, these states will therefore appear infinitely often on the path. Hence, when the observation of quiescence is made for a fairly divergent path, the system will reside in one of the states that occur infinitely often on that fairly divergent path. We call these states fairly divergent.

Definition 4.8 (Fairly divergent state)**.** Let $\mathcal{A}$ be an IOA or DQTS. A state $s \in S$ is *fairly divergent*, denoted $\mathit{fd}(s)$, if there is a (state-finite) fairly divergent path on which s occurs infinitely often, i.e., if there is a path $\pi \in \mathit{fdpaths}(\mathcal{A})$ such that $s \in \omega\text{-}\mathit{states}(\pi)$. The set of all fairly divergent states of $\mathcal{A}$ is denoted $\mathit{fd}(\mathcal{A})$.

Example 4.9. Consider the DQTS $\mathcal{B}$ in Figure 4.2b. Clearly, the path $\pi_1 = s_0\, a\, s_1\, a\, s_1\, a\, \ldots$ is fairly divergent, as it consists of only internal transitions, and no outputs are enabled in any of its states. Both states s_0 and s_1 occur on path π_1, but since $\omega\text{-}\mathit{states}(\pi_1) = \{\, s_1 \,\}$, only s_1 is fairly divergent. Whether the states s_2 and s_3 are fairly divergent depends on the partition $P_{\mathcal{A}}$. Only if there is an element $A \in P$ such that $\{\, b, e \,\} \in A$, i.e., b and e are controlled by the same subcomponent, then the divergent path $s_1\, b\, s_2\, b\, s_3\, s_1\, b\, s_2\, \ldots$ is fair, and s_2 and s_3 are fairly divergent. □

Remark 4.10. Note that a state cannot be both quiescent and fairly divergent.

4.2 Well-formedness

Earlier on, we introduced four rules R1, R2, R3 and R4 for QTSs that define exactly when a syntactically correct QTS is well-formed. A QTS that is not well-formed is considered semantically unsound, i.e., its observable behaviour does not correspond to any realistic specification or implementation. The exact same four rules also apply to well-formed DQTSs, except for rule R1: since quiescence can also be observed in fairly divergent states, as discussed above, we also require fairly divergent states to have an outgoing δ-transition.

Definition 4.11 (Well-formedness for DQTSs)**.** Let $\mathcal{A} = \langle\, S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow \,\rangle$ be a DQTS. $\mathcal{A}$ is *well-formed* if it satisfies the following rules for all states $s, s', s'' \in S$ and inputs $a \in L^{\mathrm{I}}$:

Rule R1$_{\mathbf{D}}$ (Quiescence should be observable)**:** if $q(s)$ or $\mathit{fd}(s)$, then $s \xrightarrow{\delta}$.

This DQTS-specific rule requires that not only every quiescent state, but also each fairly divergent state, has an outgoing δ-transition, since in both kinds of states quiescence may be observed, as discussed earlier.

Rule R2 (Quiescent state after observation of quiescence)**:** if $s \xrightarrow{\delta} s'$, then $q(s')$.

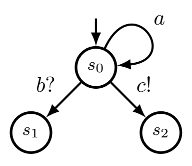
This rule demands that a quiescent state is entered after a δ-transition, i.e., no output or internal transition may take place before a new input is provided. Note that s' cannot be fairly divergent.

Rule R3 (Quiescence introduces no new behaviour)**:** if $s \xrightarrow{\delta} s'$, then $\mathit{traces}(s') \subseteq \mathit{traces}(s)$.

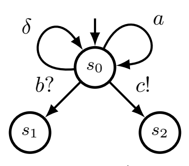
This rule prevents an observation of quiescence from enabling new behaviour.

Rule R4 (Continued quiescence preserves behaviour)**:** if $s \xrightarrow{\delta} s'$ and $s' \xrightarrow{\delta} s''$, then $\mathit{traces}(s'') = \mathit{traces}(s')$.

This rule ensures that the behaviour that can be observed after a single observation of quiescence, is the same as the behaviour that can be observed after multiple observations of quiescence in a row.



(a) $\mathcal{A}$



(b) $\mathcal{A}'$

Figure 4.3: An IOA $\mathcal{A}$ and an attempt at deltafication $\mathcal{A}'$.

Hence, rules R2, R3 and R4 are exactly the same for QTSs as for DQTSs; rule R1$_\mathrm{D}$ is similar to rule R1, but also takes fairly divergent states into account.

Clearly, for every well-formed QTS $\mathcal{Q} = \langle\, S, S^0, L^\mathrm{I}, L^\mathrm{O}, \rightarrow \,\rangle$ there exists a well-formed DQTS $\mathcal{D} = \langle\, S, S^0, L^\mathrm{I}, L^\mathrm{O}, L^\mathrm{H}, P, \rightarrow \,\rangle$ with $L^\mathrm{H} = \{\, \tau \,\}$ and $P = \{\, L^\mathrm{O} \cup \{\, \tau \,\} \,\}$, such that Q and D are isomorphic, and therefore also trace-equivalent. Therefore, the following theorem follows directly from Theorem 3.21.

Theorem 4.12. *For every SA $\mathcal{S}$ there exists a well-formed DQTS $\mathcal{D}$ such that $\mathcal{S} \approx_\mathrm{tr} \mathcal{D}$.*

Now, consider any well-formed DQTS $\mathcal{D} = \langle\, S, S^0, L^\mathrm{I}, L^\mathrm{O}, L^\mathrm{H}, P, \rightarrow \,\rangle$. Let $\mathit{det}(\mathcal{D}) = \langle\, S_\mathrm{D}, S^0_\mathrm{D}, L^\mathrm{I}, L^\mathrm{O}, L^\mathrm{H}, P, \rightarrow_\mathrm{D} \,\rangle$ be its determinisation, as defined in Definition 2.11. As will be shown in Theorem 4.21, $\mathit{det}(\mathcal{D})$ is also a well-formed DQTS, and by [BK08], is trace-equivalent to $\mathcal{D}$. Since $\mathit{det}(\mathcal{D})$ does not contain any internal transitions, it is now trivial to obtain a well-formed QTS $Q = \langle\, S_\mathrm{D}, S^0_\mathrm{D}, L^\mathrm{I}, L^\mathrm{O}, \rightarrow_\mathrm{D} \,\rangle$ that is isomorphic, and therefore trace-equivalent to $\mathit{det}(\mathcal{D})$, and hence also to $\mathcal{D}$. The following theorem then follows directly from Theorem 3.22.

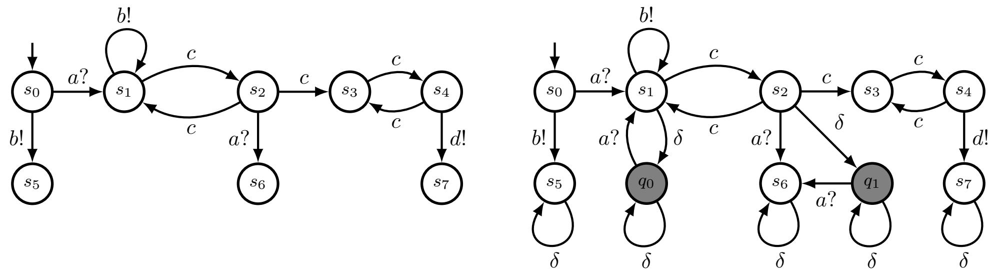
Theorem 4.13. *For every well-formed DQTS $\mathcal{D}$ there exists a SA $\mathcal{S}$ such that $\mathcal{D} \approx_\mathrm{tr} \mathcal{S}$.*

As discussed in Section 3.6, SAs are well-formed in the sense that their observable behaviour corresponds to that of realistic specifications or implementations. By the above two theorems, well-formed DQTSs are equivalent in terms of expressible behaviour. Consequently, just like QTSs, DQTSs always conform to realistic specifications or implementations in terms of expressible behaviour. Furthermore, again like QTSs, DQTSs can be used as a drop-in replacement for SAs in frameworks like `ioco`.

4.3 Deltafication: from IOA to DQTS

In Section 3.3, we introduced deltafication as a method to convert an IOTS to a well-formed QTS that captures all possible observations of it, including quiescence. As part of this approach, δ-labelled self-loops were added to all quiescent states. For QTSs, this is sufficient, as divergence was not assumed to occur and therefore no fairly divergent states could exist.

On the other hand, we do allow (state-finite) divergent paths to occur in DQTSs, and consequently quiescence may be observed in fairly divergent states of IOAs. Hence, when converting an IOA to a DQTS that captures all possible observations, including quiescence, we need a way to mark fairly divergent states with δ-transitions, in order to satisfy rule R1$_\mathrm{D}$. Simply adding δ-labelled self-loops to all fairly divergent states, as is done for all quiescent states, may yield DQTSs that are not well-formed. To see this, consider the IOA $\mathcal{A}$ in Figure 4.3a and assume $P_\mathcal{A} = \{\, \{\, a, c \,\} \,\}$. Clearly, the state s_0 is fairly divergent, as it occurs infinitely often on the fairly divergent path $s_0\, a\, s_0\, a\, \ldots$ in $\mathcal{A}$. Adding a δ-labelled self-loop



(a) $\mathcal{A}$ (b) $\delta(\mathcal{A})$

Figure 4.4: An IOA $\mathcal{A}$ and its deltafication $\delta(\mathcal{A})$. Newly introduced states are marked gray.

to s_0, as shown in Figure 4.3b, violates rule R2: state s_0 can be reached with a δ-transition from itself, but is not quiescent since the output c is also enabled.

Since fairly divergent states must have an outgoing δ-transition to satisfy rule $\mathrm{R1_D}$, and adding δ-labelled self-loops is not the correct solution, we must introduce a new state for every fairly divergent state, which then acts as a *quiescence observation state* for it. Thus, when quiescence is observed in a fairly divergent state, the outgoing δ-transition will lead to the associated quiescence observation state. However, in order to preserve the original behaviour while ensuring input-enabledness, all inputs that are enabled in the fairly divergent state must still be enabled in the corresponding quiescence observation state, and must lead to the same states that the original input transitions led to. All these considerations together lead to the following definition for the deltafication procedure for IOAs.

Definition 4.14 (Deltafication of IOAs)**.** Let $\mathcal{A} = \langle\, S_{\mathcal{A}}, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow_{\mathcal{A}} \rangle$ be an IOA with $\delta \notin L$, such that any divergent paths are state-finite. The *deltafication* of $\mathcal{A}$ is the DQTS $\delta(\mathcal{A}) = \langle\, S_\delta, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow_\delta \rangle$. We define $S_\delta = S_{\mathcal{A}} \cup \{\, qos_s \mid s \in S_{\mathcal{A}} \text{ and } fd(s) \text{ in } \mathcal{A} \}$, i.e., S_δ contains all the states in $S_{\mathcal{A}}$ plus a new state $qos_s \notin S_{\mathcal{A}}$ for every fairly divergent state $s \in S_{\mathcal{A}}$ of $\mathcal{A}$; the state qos_s is the *quiescence observation state* of s. The transition relation $\rightarrow_\delta$ is defined as follows:

$$\begin{array}{rcll} \rightarrow_\delta \;=\; \rightarrow_{\mathcal{A}} & \cup & \{\, (s, \delta, s) & \mid\; s \in S_{\mathcal{A}} \,\wedge\, s \in q(\mathcal{A}) \,\} \\ & \cup & \{\, (s, \delta, qos_s) & \mid\; s \in S_{\mathcal{A}} \,\wedge\, s \in fd(\mathcal{A}) \,\} \\ & \cup & \{\, (qos_s, \delta, qos_s) & \mid\; s \in S_{\mathcal{A}} \,\wedge\, s \in fd(\mathcal{A}) \,\} \\ & \cup & \{\, (qos_s, a?, s') & \mid\; s, s' \in S_{\mathcal{A}} \,\wedge\, s \in fd(\mathcal{A}) \,\wedge\, a? \in L^{\mathrm{I}} \,\wedge\, s \xrightarrow{a?}_{\mathcal{A}} s' \,\} \end{array}$$

Thus, the deltafication of an IOA adds δ-labelled self-loops to all quiescent states, similar to deltafication for IOTSs. Furthermore, a new quiescence observation state qos_s is introduced for every fairly divergent state $s \in S$, which can be reached with a newly added δ-transition from s. Additional outgoing input-transitions are added to the quiescence observation states to ensure previously enabled inputs are still enabled after observing quiescence, as discussed above. In this way, the deltafication operation preserves input-enabledness.

Example 4.15. An IOA $\mathcal{A}$ and its deltafication $\delta(\mathcal{A})$ are shown in Figure 4.4. We assume $P_{\mathcal{A}} = \{\, \{\, b, c \,\}, \{\, d \,\} \,\}$. Hence, s_1 and s_2 are fairly divergent in $\mathcal{A}$. The states s_3 and s_4 are not fairly divergent, since the d-output is enabled in s_4. Consequently, deltafication has introduced the quiescence observation states q_0 and q_1, alongside the necessary input-transitions, for s_1 and s_2, respectively. Furthermore, the deltafication has resulted in new δ-labelled self-loops for the quiescent states s_5, s_6 and s_7. □

The following theorem states that this deltafication approach indeed yields a well-formed DQTS.

Theorem 4.16. *Given an IOA $\mathcal{A}$ with $\delta \notin L$ such that all divergent paths in $\mathcal{A}$ are state-finite, $\delta(\mathcal{A})$ is a well-formed DQTS.*

Proof. Let $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow_{\mathcal{A}} \rangle$ be an IOA such that $\delta \notin L$, and let $\delta(\mathcal{A}) = \langle S_\delta, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow_\delta \rangle$ be its deltafication, as defined in Definition 4.14. To show that $\delta(\mathcal{A})$ is a well-formed DQTS, we need to prove that $\delta(\mathcal{A})$ satisfies each of the rules R1$_{\mathrm{D}}$, R2, R3 and R4. In the following, we use $\mathit{traces}_\delta(s)$ to denote the set of all traces of $\delta(\mathcal{A})$ starting in the state $s \in S_\delta$.

1. To prove that $\delta(\mathcal{A})$ satisfies rule R1$_{\mathrm{D}}$, we must show that for all states $s \in S_\delta$:

   $$\text{if } q(s) \text{ or } \mathit{fd}(s), \text{ then } s \xrightarrow{\delta}_\delta$$

   Since $s \in S_\delta$ and $q(s)$ or $\mathit{fd}(s)$ holds in $\delta(\mathcal{A})$, it follows from Definition 4.14 that the following cases are possible: (a) $s \in S$ and $q(s)$ holds in $\delta(\mathcal{A})$; (b) $s \in S$ and $\mathit{fd}(s)$ in $\delta(autA)$; and (c) $s \in S_\delta \setminus S$ (and $q(s)$ holds in $\delta(\mathcal{A})$). Clearly, it is not possible that $s \in S_\delta \setminus S$ and $\mathit{fd}(s)$ holds in $\delta(\mathcal{A})$.

   (a) Assume $s \in S$ and $q(s)$ holds in $\delta(\mathcal{A})$. Since deltafication does not hide or remove any existing output or internal transitions, $q(s)$ then also holds in $\mathcal{A}$. By Definition 4.14, we have $(s, \delta, s) \in \rightarrow_\delta$ after deltafication and therefore $s \xrightarrow{\delta}_\delta$.

   (b) Assume $s \in S$ and $\mathit{fd}(s)$ holds in $\delta(\mathcal{A})$. In other words, s occurs infinitely often on a fairly divergent path π in $\delta(\mathcal{A})$. Since deltafication does not hide any existing output transitions, nor creates any new internal transitions, the fairly divergent path π must also be present in $\mathcal{A}$. Consequently, $\mathit{fd}(s)$ also holds in $\mathcal{A}$. By Definition 4.14, we have $(s, \delta, \mathit{qos}_s) \in \rightarrow_\delta$ after deltafication, where qos_s is a new quiescence observation state for s. Thus, $s \xrightarrow{\delta}_\delta$.

   (c) Assume $s \in S_\delta \setminus S$. Hence, s is a newly added quiescence observation state for some fairly divergent state, and by Definition 4.14 we have both $q(s)$ and $s \xrightarrow{\delta}_\delta s$.

2. To prove that $\delta(\mathcal{A})$ satisfies rule R2, we must show that for all states $s, s' \in S_\delta$:

   $$\text{if } s \xrightarrow{\delta}_\delta s', \text{ then } q(s')$$

   Since $s, s' \in S_\delta$ and $s \xrightarrow{\delta}_\delta s'$, it follows from Definition 4.14 that the following cases are possible: (a) $s, s' \in S$; (b) $s \in S$ and $s' \in S_\delta \setminus S$; and (c) $s, s' \in S_\delta \setminus S$. Clearly, it is not possible that $s \in S_\delta \setminus S$, $s' \in S$, and $s \xrightarrow{\delta}_\delta s'$.

   (a) Assume $s, s' \in S$ and $s \xrightarrow{\delta}_\delta s'$. By Definition 4.14, we have $s = s'$, and s (and therefore also s') is quiescent.

   (b) Assume $s \in S$, $s' \in S_\delta \setminus S$, and $s \xrightarrow{\delta}_\delta s'$. From Definition 4.14, it follows that s' is the quiescence observation state for the fairly divergent state s, and s' is quiescent.

   (c) Assume $s, s' \in S_\delta \setminus S$ and $s \xrightarrow{\delta}_\delta s'$. From Definition 4.14, it follows that s' is a quiescence observation state, $s = s'$, and s' is quiescent.

3. To prove that $\delta(\mathcal{A})$ satisfies rule R3, we must show that for all states $s, s' \in S_\delta$:

$$\text{if } s \xrightarrow{\delta}_\delta s', \text{ then } \mathit{traces}_\delta(s') \subseteq \mathit{traces}_\delta(s)$$

   Since $s, s' \in S_\delta$ and $s \xrightarrow{\delta}_\delta s'$, it follows from Definition 4.14 that the following cases are possible: (a) $s, s' \in S$; (b) $s \in S$ and $s' \in S_\delta \setminus S$; and (c) $s, s' \in S_\delta \setminus S$. Clearly, it is not possible that $s \in S_\delta \setminus S$, $s' \in S$, and $s \xrightarrow{\delta}_\delta s'$.

   (a) Assume $s, s' \in S$ and $s \xrightarrow{\delta}_\delta s'$. By Definition 4.14, we have $s = s'$, and therefore $\mathit{traces}_\delta(s') \subseteq \mathit{traces}_\delta(s)$.

   (b) Assume $s \in S$, $s' \in S_\delta \setminus S$ and $s \xrightarrow{\delta}_\delta s'$. From Definition 4.14, it follows that s' is a quiescence observation state for the fairly divergent state s. Let $\sigma \in \mathit{traces}_\delta(s')$. We have to show that also $\sigma \in \mathit{traces}_\delta(s)$. There are two cases to consider: either $|\sigma| = 0$ or $|\sigma| \geq 1$. If $|\sigma| = 0$, then $\sigma = \epsilon$, and by definition $\sigma \in \mathit{traces}_\delta(s)$. If $|\sigma| \geq 1$, then, by Definition 4.14, $\sigma = a \cdot \sigma'$, where either $a = \delta$, or $a \in L^{\mathrm{I}}(s)$. In the first case we have $s' \xrightarrow{\delta}_\delta s'$ and $s' \xRightarrow{\sigma'}_\delta$. Since also $s \xrightarrow{\delta}_\delta s'$, it directly follows that $\sigma \in \mathit{traces}_\delta(s)$. In the second case we have $s' \xrightarrow{a}_\delta s''$ and $s'' \xRightarrow{\sigma'}_\delta$ for some $s'' \in S$. By Definition 4.14, we then must have $s \xrightarrow{a}_\mathcal{A} s''$, and therefore also $s \xrightarrow{a}_\delta s''$. Hence, since we have $s'' \xRightarrow{\sigma'}_\delta$, we find $\sigma \in \mathit{traces}_\delta(s)$.

   (c) Assume $s, s' \in S_\delta \setminus S$ and $s \xrightarrow{\delta}_\delta s'$. From Definition 4.14, it follows that s is a quiescence observation state and $s = s'$. Thus, $\mathit{traces}_\delta(s') \subseteq \mathit{traces}_\delta(s)$.

4. To prove that $\delta(\mathcal{A})$ satisfies rule R4, we must show that for all states $s, s', s'' \in S_\delta$:

$$\text{if } s \xrightarrow{\delta}_\delta s' \text{ and } s' \xrightarrow{\delta}_\delta s'', \text{ then } \mathit{traces}_\delta(s'') = \mathit{traces}_\delta(s')$$

   Since $s, s', s'' \in S_\delta$, $s \xrightarrow{\delta}_\delta s'$ and $s' \xrightarrow{\delta}_\delta s''$, it follows from Definition 4.14 that the following cases are possible: (a) $s, s', s'' \in S$; (b) $s \in S$ and $s', s'' \in S_\delta \setminus S$; and (c) $s, s', s'' \in S_\delta \setminus S$. All other permutations are not possible.

   (a) Assume $s, s', s'' \in S$, $s \xrightarrow{\delta}_\delta s'$ and $s' \xrightarrow{\delta}_\delta s''$. By Definition 3.8, we have $s = s' = s''$, and therefore $\mathit{traces}_\delta(s') = \mathit{traces}_\delta(s'')$.

   (b) Assume $s \in S$, $s', s'' \in S_\delta \setminus S$, $s \xrightarrow{\delta}_\delta s'$ and $s' \xrightarrow{\delta}_\delta s''$. From Definition 4.14, it follows that s' is the quiescence observation state for the fairly divergent state s, and $s' = s''$. Clearly then, $\mathit{traces}_\delta(s'') = \mathit{traces}_\delta(s')$.

   (c) Assume $s, s', s'' \in S_\delta \setminus S$, $s \xrightarrow{\delta}_\delta s'$ and $s' \xrightarrow{\delta}_\delta s''$. From Definition 4.14, it follows that s is a quiescence observation state and $s = s' = s''$. Thus, $\mathit{traces}_\delta(s'') = \mathit{traces}_\delta(s')$. □

4.4 Operations

Now, we are ready to take a look at some of the operations that can be applied to DQTSs: determinisation, parallel composition and action hiding. Determinisation for DQTSs is exactly the same as for IOAs. Parallel composition is also very similar, but action hiding is more complicated in DQTSs due to the possibility of newly divergent states arising after hiding.

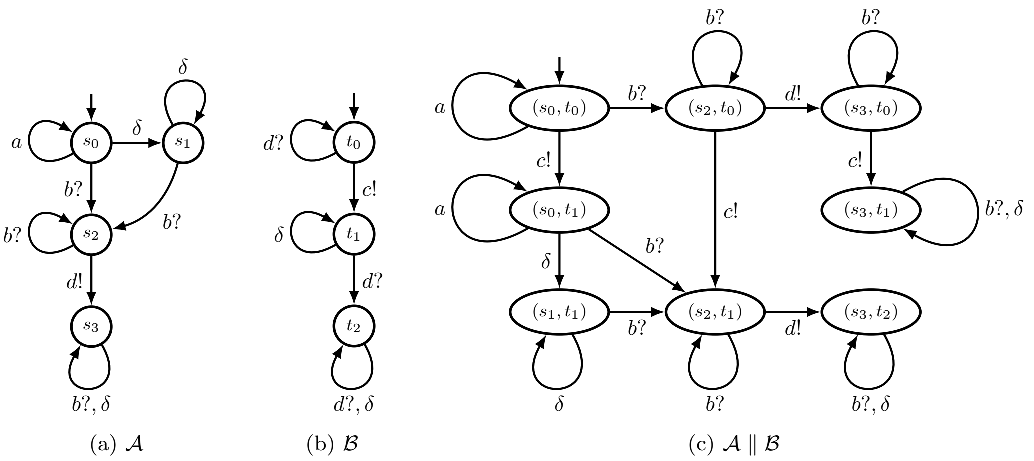


Figure 4.5: The DQTSs $\mathcal{A}$ and $\mathcal{B}$, and their parallel composition $\mathcal{A} \parallel \mathcal{B}$.

4.4.1 Parallel Composition

The parallel composition of two DQTSs is exactly the same as for IOAs, except that there is an added synchronisation on the δ-label, as was the case for the parallel composition of two QTSs. The compatibility requirement (see Definition 2.34) is also exactly the same as for IOAs, except that both DQTSs may share the δ-output.

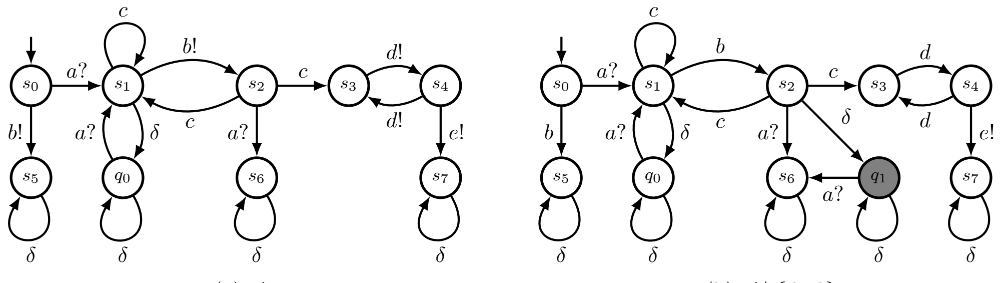
Definition 4.17 (Parallel composition of DQTSs)**.** Given two compatible, well-formed DQTSs $\mathcal{A} = \langle\, S_{\mathcal{A}}, S^0_{\mathcal{A}}, L^{\mathrm{I}}_{\mathcal{A}}, L^{\mathrm{O}}_{\mathcal{A}}, L^{\mathrm{H}}_{\mathcal{A}}, P_{\mathcal{A}}, \rightarrow_{\mathcal{A}} \rangle$ and $\mathcal{B} = \langle\, S_{\mathcal{B}}, S^0_{\mathcal{B}}, L^{\mathrm{I}}_{\mathcal{B}}, L^{\mathrm{O}}_{\mathcal{B}}, L^{\mathrm{H}}_{\mathcal{B}}, P_{\mathcal{B}}, \rightarrow_{\mathcal{B}} \rangle$, their *parallel composition* is the DQTS $\mathcal{A} \parallel \mathcal{B} = \langle\, S_{\mathcal{A}\parallel\mathcal{B}}, S^0_{\mathcal{A}\parallel\mathcal{B}}, L^{\mathrm{I}}_{\mathcal{A}\parallel\mathcal{B}}, L^{\mathrm{O}}_{\mathcal{A}\parallel\mathcal{B}}, L^{\mathrm{H}}_{\mathcal{A}\parallel\mathcal{B}}, P_{\mathcal{A}\parallel\mathcal{B}}, \rightarrow_{\mathcal{A}\parallel\mathcal{B}} \rangle$, where $S_{\mathcal{A}\parallel\mathcal{B}}$, $S^0_{\mathcal{A}\parallel\mathcal{B}}$, $L^{\mathrm{I}}_{\mathcal{A}\parallel\mathcal{B}}$, $L^{\mathrm{O}}_{\mathcal{A}\parallel\mathcal{B}}$, $L^{\mathrm{H}}_{\mathcal{A}\parallel\mathcal{B}}$, $P_{\mathcal{A}\parallel\mathcal{B}}$ and $\rightarrow_{\mathcal{A}\parallel\mathcal{B}}$ are defined as follows:

$$\begin{array}{rcl}
S_{\mathcal{A}\parallel\mathcal{B}} & = & S_{\mathcal{A}} \times S_{\mathcal{B}} \\
S^0_{\mathcal{A}\parallel\mathcal{B}} & = & S^0_{\mathcal{A}} \times S^0_{\mathcal{B}} \\
L^{\mathrm{I}}_{\mathcal{A}\parallel\mathcal{B}} & = & (L^{\mathrm{I}}_{\mathcal{A}} \cup L^{\mathrm{I}}_{\mathcal{B}}) \setminus (L^{\mathrm{O}}_{\mathcal{A}} \cup L^{\mathrm{O}}_{\mathcal{B}}) \\
L^{\mathrm{O}}_{\mathcal{A}\parallel\mathcal{B}} & = & L^{\mathrm{O}}_{\mathcal{A}} \cup L^{\mathrm{O}}_{\mathcal{B}} \\
L^{\mathrm{H}}_{\mathcal{A}\parallel\mathcal{B}} & = & L^{\mathrm{H}}_{\mathcal{A}} \cup L^{\mathrm{H}}_{\mathcal{B}} \\
P_{\mathcal{A}\parallel\mathcal{B}} & = & P_{\mathcal{A}} \cup P_{\mathcal{B}} \\
\rightarrow_{\mathcal{A}\parallel\mathcal{B}} & = & \{\, ((s,t),a,(s',t')) \in S_{\mathcal{A}\parallel\mathcal{B}} \times ((L_{\mathcal{A}} \cap L_{\mathcal{B}}) \cup \{\,\delta\,\}) \times S_{\mathcal{A}\parallel\mathcal{B}} \mid s \xrightarrow{a}_{\mathcal{A}} s' \;\wedge\; t \xrightarrow{a}_{\mathcal{B}} t' \,\} \\
& \cup & \{\, ((s,t),a,(s',t)) \in S_{\mathcal{A}\parallel\mathcal{B}} \times (L_{\mathcal{A}} \setminus L_{\mathcal{B}}) \times S_{\mathcal{A}\parallel\mathcal{B}} \mid s \xrightarrow{a}_{\mathcal{A}} s' \,\} \\
& \cup & \{\, ((s,t),a,(s,t')) \in S_{\mathcal{A}\parallel\mathcal{B}} \times (L_{\mathcal{B}} \setminus L_{\mathcal{A}}) \times S_{\mathcal{A}\parallel\mathcal{B}} \mid t \xrightarrow{a}_{\mathcal{B}} t' \,\}
\end{array}$$

As with parallel composed IOAs, we have $L_{\mathcal{A}\parallel\mathcal{B}} = L^{\mathrm{I}}_{\mathcal{A}\parallel\mathcal{B}} \cup L^{\mathrm{O}}_{\mathcal{A}\parallel\mathcal{B}} \cup L^{\mathrm{H}}_{\mathcal{A}\parallel\mathcal{B}} = L_{\mathcal{A}} \cup L_{\mathcal{B}}$.

The first clause of $\rightarrow_{\mathcal{A}\parallel\mathcal{B}}$ ensures that parallel composed DQTSs synchronise on shared actions and the δ-label. The next two clauses enable them to perform non-shared actions independently from each other.

Example 4.18. See Figure 4.5 for two well-formed, compatible DQTSs $\mathcal{A}$ and $\mathcal{B}$, and their parallel composition $\mathcal{A} \parallel \mathcal{B}$. We have $P_{\mathcal{A}} = \{\,\{\,a,d\,\}\,\}$ and $P_{\mathcal{B}} = \{\,\{\,c\,\}\,\}$, therefore $P_{\mathcal{A}\parallel\mathcal{B}} = \{\,\{\,a,d\,\},\{\,c\,\}\,\}$. Consequently, even though state s_0 is fairly divergent in $\mathcal{A}$, the composite state (s_0,t_0) in $\mathcal{A} \parallel \mathcal{B}$ is not, since the output c, which is controlled independently by



(a) $\mathcal{A}$ (b) $\mathcal{A}\setminus\{\, b, d \,\}$

Figure 4.6: The DQTSs $\mathcal{A}$ and $\mathcal{A}\setminus\{\, b, d \,\}$. Newly introduced states are marked gray.

component $\mathcal{B}$, is enabled in this state. Thus, the task partition P (and the corresponding notion of fairness) allows us to correctly determine in which states divergence can be observed and in which not. □

4.4.2 Action Hiding

Action hiding is more complicated for DQTSs than for QTSs or regular IOAs, as transforming output actions to internal actions can lead to new fairly divergent states. To see this, consider the DQTS $\mathcal{A}$ in Figure 4.6a. Hiding the output b would result in the state s_2 becoming fairly divergent, as it then would occur infinitely often on the fairly divergent path $s_2\, c\, s_1\, b\, s_2\, c\, \ldots$, for example. As a consequence, after hiding new quiescence observation states may have to be added for newly fairly divergent states.

Definition 4.19 (Action hiding in DQTSs)**.** Let $\mathcal{A} = \langle\, S_{\mathcal{A}}, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow_{\mathcal{A}} \rangle$ be a well-formed DQTS and $H \subseteq L^{\mathrm{O}}$ a set of outputs, then *hiding H in $\mathcal{A}$* yields the DQTS $\mathcal{A} \setminus H = \langle\, S_H, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}_H, L^{\mathrm{H}}_H, P, \rightarrow_H \rangle$, with $L^{\mathrm{O}}_H = L^{\mathrm{O}} \setminus H$ and $L^{\mathrm{H}}_H = L^{\mathrm{H}} \cup H$. Furthermore, we define $S_H = S_{\mathcal{A}} \cup \{\, qos_s \mid s \in (fd(\mathcal{A} \setminus H) \setminus fd(\mathcal{A}))\,\}$, i.e., S_H contains all the states of $S_{\mathcal{A}}$ plus a new quiescence observation state $qos_s \notin S_{\mathcal{A}}$ for every state $s \in S_{\mathcal{A}}$ that has become newly fairly divergent in $\mathcal{A} \setminus H$. Finally, $\rightarrow_H$ is defined as follows:

$$\begin{aligned}
\rightarrow_H \;=\; \rightarrow_{\mathcal{A}} \;&\cup\; \{\, (s, \delta, qos_s) &&\mid s \in (fd(\mathcal{A} \setminus H) \setminus fd(\mathcal{A}))\,\} \\
&\cup\; \{\, (qos_s, \delta, qos_s) &&\mid s \in (fd(\mathcal{A} \setminus H) \setminus fd(\mathcal{A}))\,\} \\
&\cup\; \{\, (qos_s, a?, s') &&\mid s \in (fd(\mathcal{A} \setminus H) \setminus fd(\mathcal{A})) \;\wedge\; a? \in L^{\mathrm{I}} \;\wedge\; s \xrightarrow{a?}_{\mathcal{A}} s' \,\}
\end{aligned}$$

Thus, the hiding operation simply removes the output labels that are to be hidden from the set of outputs, and adds them to the set of internal labels, similar to the hiding operation for IOAs. Furthermore, new quiescence observation states are introduced for all newly fairly divergent states in $\mathcal{A} \setminus H$, along with the required input-transitions, similar to the deltafication procedure for fairly divergent states in IOAs. Clearly, the hiding operation preserves input-enabledness.

Example 4.20. Consider the DQTSs $\mathcal{A}$ and $\mathcal{A}\setminus\{\, b, d \,\}$ in Figure 4.6, and assume $P_{\mathcal{A}} = \{\, \{\, b, c, d \,\}, \{\, e \,\} \,\}$. Note that we indicate that the outputs b and d have been hidden in $\mathcal{A}\setminus\{\, b, d \,\}$, and thus are treated as internal actions, by leaving out the exclamation marks in their labels. In this case, we have $fd(\mathcal{A}) = \{\, s_1 \,\}$ and $fd(\mathcal{A} \setminus H) = \{\, s_1, s_2 \,\}$. Note that s_3 and s_4 are not fairly divergent after hiding because the output e is still enabled in s_4. Consequently, the new quiescence observation state q_1 is introduced for the state s_2, along with the required δ-transitions and input-transitions. □

4.5 Properties

In this section, we investigate the closure and commutativity properties of DQTSs for the operations of determinisation, parallel composition, and action hiding.

4.5.1 Closure Properties

Similar to well-formed QTSs, well-formed DQTSs are closed under the operations of determinisation, parallel composition, and action hiding. These operations are therefore well-defined for well-formed DQTSs.

Theorem 4.21. *Well-formed DQTSs are closed under determinisation, i.e., given a well-formed DQTS $\mathcal{A}$, $det(\mathcal{A})$ is also a well-formed DQTS.*

Proof. Let $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow_{\mathcal{A}} \rangle$ be a well-formed DQTS and let $det(\mathcal{A}) = \langle S_{\mathrm{D}}, S^0_{\mathrm{D}}, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow_{\mathrm{D}} \rangle$ be its determinisation, as defined in Definition 2.11. To prove that well-formed DQTSs are closed under determinisation we must show that $det(\mathcal{A})$ is a well-formed DQTS, i.e., that it satisfies each of the rules R1$_{\mathrm{D}}$, R2, R3 and R4. In the following, we use $traces_{\mathrm{D}}(U)$ to denote the set of all traces of $det(\mathcal{A})$ starting in the state $U \in S_{\mathrm{D}}$.

1. To prove that $det(\mathcal{A})$ satisfies rule R1$_{\mathrm{D}}$, we must show that for all states $U \in S_{\mathrm{D}}$:

   $$\text{if } q(U) \text{ or } fd(U)\text{, then } U \xrightarrow{\delta}_{\mathrm{D}}$$

   By Definition 2.11, there are no more internal transitions present after determinisation. Hence, there can be no $U \in S_{\mathrm{D}}$ such that $fd(U)$ holds in $det(\mathcal{A})$. Instead, assume $q(U)$ holds in $det(\mathcal{A})$ for an $U \in S_{\mathrm{D}}$. This implies that all states $s \in U$ are quiescent in $\mathcal{A}$. From rule R1$_{\mathrm{D}}$ it follows that for every state $s \in U$ there exists another state $s' \in S$ such that $s \xrightarrow{\delta}_{\mathcal{A}} s'$. Therefore $reach_{\mathcal{A}}(U, \delta) \neq \emptyset$. By Definition 2.11, we then have $(U, \delta, reach_{\mathcal{A}}(U, \delta)) \in \rightarrow_{\mathrm{D}}$. Consequently, $U \xrightarrow{\delta}_{\mathrm{D}}$.

2. To prove that $det(\mathcal{A})$ satisfies rule R2, we must show that for all states $U, V \in S_{\mathrm{D}}$:

   $$\text{if } U \xrightarrow{\delta}_{\mathrm{D}} V\text{, then } q(V)$$

   Consider any transition $U \xrightarrow{\delta}_{\mathrm{D}} V$ with $U, V \in S_{\mathrm{D}}$. If $U \xrightarrow{\delta}_{\mathrm{D}} V$, then, by Definition 2.11, $V = reach_{\mathcal{A}}(U, \delta)$ and $V \neq \emptyset$. Hence, for every state $s' \in V$ there exists a state $s \in U$ such that $s \xrightarrow{\delta}_{\mathcal{A}} s'$. Using rule R2 we can then conclude that every $s' \in V$ is quiescent in $\mathcal{A}$, thus $q(V)$ holds in $det(\mathcal{A})$.

3. To prove that $det(\mathcal{A})$ satisfies rule R3, we must show that for all states $U, V \in S_{\mathrm{D}}$:

   $$\text{if } U \xrightarrow{\delta}_{\mathrm{D}} V\text{, then } traces_{\mathrm{D}}(V) \subseteq traces_{\mathrm{D}}(U)$$

   Consider any transition $U \xrightarrow{\delta}_{\mathrm{D}} V$ with $U, V \in S_{\mathrm{D}}$. Assume $\sigma \in traces_{\mathrm{D}}(V)$. We must show that also $\sigma \in traces_{\mathrm{D}}(U)$. If $\sigma \in traces_{\mathrm{D}}(V)$, then there clearly must exist a state $s' \in V$ such that $s' \xRightarrow{\sigma}_{\mathcal{A}}$. Since $U \xrightarrow{\delta}_{\mathrm{D}} V$, it follows from Definition 2.11 that $V = reach_{\mathcal{A}}(U, \delta)$ and $V \neq \emptyset$. Hence, there must exist a state $s \in U$ such that $s \xrightarrow{\delta}_{\mathcal{A}} s'$. Using rule R3 we can then conclude that $traces_{\mathcal{A}}(s') \subseteq traces_{\mathcal{A}}(s)$, and therefore $s \xRightarrow{\sigma}_{\mathcal{A}}$. Since $s \in U$, it follows that $\sigma \in traces_{\mathrm{D}}(U)$.

4. To prove that $det(\mathcal{A})$ satisfies rule R4, we must show that for all states $U, V, W \in S_{\mathrm{D}}$:

$$\text{if } U \xrightarrow{\delta}_{\mathrm{D}} V \text{ and } V \xrightarrow{\delta}_{\mathrm{D}} W, \text{ then } \mathit{traces}_{\mathrm{D}}(W) = \mathit{traces}_{\mathrm{D}}(V)$$

Consider any pair of transitions $U \xrightarrow{\delta}_{\mathrm{D}} V$ and $V \xrightarrow{\delta}_{\mathrm{D}} W$, with $U, V, W \in S_{\mathrm{D}}$. To prove that $\mathit{traces}_{\mathrm{D}}(W) = \mathit{traces}_{\mathrm{D}}(V)$, we must show that both $\mathit{traces}_{\mathrm{D}}(W) \subseteq \mathit{traces}_{\mathrm{D}}(V)$ and $\mathit{traces}(V) \subseteq \mathit{traces}_{\mathrm{D}}(W)$. The former follows directly from rule R3, so all that's left to prove is that $\mathit{traces}_{\mathrm{D}}(V) \subseteq \mathit{traces}_{\mathrm{D}}(W)$.

Assume $\sigma \in \mathit{traces}_{\mathrm{D}}(V)$. We must show that also $\sigma \in \mathit{traces}_{\mathrm{D}}(W)$. If $\sigma \in \mathit{traces}_{\mathrm{D}}(V)$, then there clearly must exist a state $s' \in V$ such that $s' \xRightarrow{\sigma}_{\mathcal{A}}$. Since $U \xrightarrow{\delta}_{\mathrm{D}} V$, it follows that there exists a state $s \in U$ such that $s \xrightarrow{\delta}_{\mathcal{A}} s'$. Furthermore, it follows from rule R2 that V is quiescent, and therefore all states in V are quiescent, including s'. Since $V \xrightarrow{\delta}_{\mathrm{D}} W$, we have $W = \mathit{reach}(V, \delta)$ and $W \neq \emptyset$. We can then conclude, using rule R1$_{\mathrm{D}}$, that there must exist a state $s'' \in W$ such that $s' \xrightarrow{\delta}_{\mathcal{A}} s''$. Thus, we have $s \xrightarrow{\delta}_{\mathcal{A}} s' \xrightarrow{\delta}_{\mathcal{A}} s''$. From rule R4 it then follows that $\mathit{traces}(s'') = \mathit{traces}(s')$ and consequently $s'' \xRightarrow{\sigma}_{\mathcal{A}}$. Since $s'' \in W$, it follows that $\sigma \in \mathit{traces}_{\mathrm{D}}(W)$. □

Theorem 4.22. *Well-formed DQTSs are closed under parallel composition, i.e., given two compatible well-formed DQTSs $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \parallel \mathcal{B}$ is also a well-formed DQTS.*

Proof. Given two well-formed DQTSs $\mathcal{A} = \langle S_{\mathcal{A}}, S^0_{\mathcal{A}}, L^{\mathrm{I}}_{\mathcal{A}}, L^{\mathrm{O}}_{\mathcal{A}}, L^{\mathrm{H}}_{\mathcal{A}}, P_{\mathcal{A}}, \rightarrow_{\mathcal{A}} \rangle$ and $\mathcal{B} = \langle S_{\mathcal{B}}, S^0_{\mathcal{B}}, L^{\mathrm{I}}_{\mathcal{B}}, L^{\mathrm{O}}_{\mathcal{B}}, L^{\mathrm{H}}_{\mathcal{B}}, P_{\mathcal{B}}, \rightarrow_{\mathcal{B}} \rangle$ that are compatible (see Definition 2.34), let the DQTS $\mathcal{A} \parallel \mathcal{B} = \langle S_{\mathcal{A}\|\mathcal{B}}, S^0_{\mathcal{A}\|\mathcal{B}}, L^{\mathrm{I}}_{\mathcal{A}\|\mathcal{B}}, L^{\mathrm{O}}_{\mathcal{A}\|\mathcal{B}}, L^{\mathrm{H}}_{\mathcal{A}\|\mathcal{B}}, P_{\mathcal{A}\|\mathcal{B}}, \rightarrow_{\mathcal{A}\|\mathcal{B}} \rangle$ be their parallel composition, as defined in Definition 4.17. To prove that well-formed DQTSs are closed under parallel composition we need to show that $\mathcal{A} \parallel \mathcal{B}$ is a well-formed DQTS, i.e., we need to prove that $\mathcal{A} \parallel \mathcal{B}$ satisfies each of the rules R1$_{\mathrm{D}}$, R2, R3 and R4.

1. To prove that $\mathcal{A} \parallel \mathcal{B}$ satisfies rule R1$_{\mathrm{D}}$, we must show that for every state $(s,t) \in S_{\mathcal{A}\|\mathcal{B}}$:

$$\text{if } q((s,t)) \text{ or } \mathit{fd}((s,t)), \text{ then } (s,t) \xrightarrow{\delta}_{\mathcal{A}\|\mathcal{B}}$$

Let $(s,t) \in S_{\mathcal{A}\|\mathcal{B}}$. We will look at the cases for $q((s,t))$ and $\mathit{fd}((s,t))$ separately.

First, assume $q((s,t))$ holds in $\mathcal{A} \parallel \mathcal{B}$. In this case, there is no $a \in L^{\mathrm{O}}_{\mathcal{A}\|\mathcal{B}} \cup L^{\mathrm{H}}_{\mathcal{A}\|\mathcal{B}}$ such that $(s,t) \xrightarrow{a}_{\mathcal{A}\|\mathcal{B}}$. Since both $\mathcal{A}$ and $\mathcal{B}$ are input-enabled, it follows from Definition 4.17 that there is no $a \in L^{\mathrm{O}}_{\mathcal{A}} \cup L^{\mathrm{H}}_{\mathcal{A}}$ such that $s \xrightarrow{a}_{\mathcal{A}}$ and no $a \in L^{\mathrm{O}}_{\mathcal{B}} \cup L^{\mathrm{H}}_{\mathcal{B}}$ such that $t \xrightarrow{a}_{\mathcal{B}}$. Hence, both s and t are quiescent, and by rule R1$_{\mathrm{D}}$ we have $s \xrightarrow{\delta}_{\mathcal{A}}$ and $t \xrightarrow{\delta}_{\mathcal{B}}$. From Definition 4.17 it then follows that $(s,t) \xrightarrow{\delta}_{\mathcal{A}\|\mathcal{B}}$.

Now, assume $\mathit{fd}((s,t))$ holds in $\mathcal{A} \parallel \mathcal{B}$, i.e., there exists a fairly divergent path $\pi \in \mathit{fdpaths}(\mathcal{A} \parallel \mathcal{B})$ such that $(s,t) \in \omega\text{-}\mathit{states}(\pi)$, i.e., the state (s,t) appears infinitely often on an infinite path π that is both divergent and fair. By Definition 4.17, each step of path π is a transition by either $\mathcal{A}$ or $\mathcal{B}$, since the sets of internal transitions of $\mathcal{A}$ and $\mathcal{B}$ are disjoint, and they cannot synchronise on them. We can therefore distinguish three cases: (a) $\mathcal{A}$ and $\mathcal{B}$ both carry out an infinite number of internal transitions in the path π; (b) $\mathcal{A}$ carries out a finite number of internal transitions, and $\mathcal{B}$ an infinite number; and (c), $\mathcal{B}$ carries out a finite number of internal transitions, and $\mathcal{A}$ an infinite number. For each case, we will show that both $s \xrightarrow{\delta}_{\mathcal{A}}$ and $t \xrightarrow{\delta}_{\mathcal{B}}$, and therefore, by Definition 4.17, also $(s,t) \xrightarrow{\delta}_{\mathcal{A}\|\mathcal{B}}$.

(a) Assume both $\mathcal{A}$ and $\mathcal{B}$ carry out an infinite number of internal transitions in the path π. Now assume that $\mathcal{A}$ carries out all the even transitions (i.e., the second, fourth, etc.) and $\mathcal{B}$ all the odd transitions (i.e., the first, third, etc.) in path π. However, the following proof can also be adapted for any other path π. Hence, path π is defined as follows:

$$\pi = (s_0, t_0) \xrightarrow{b_1}_{\mathcal{A}\|\mathcal{B}} (s_0, t_1) \xrightarrow{a_1}_{\mathcal{A}\|\mathcal{B}} (s_1, t_1) \xrightarrow{b_2}_{\mathcal{A}\|\mathcal{B}} (s_1, t_2) \xrightarrow{a_2}_{\mathcal{A}\|\mathcal{B}} (s_2, t_2) \ldots$$

where $s_i \in S_{\mathcal{A}}$, $t_i \in S_{\mathcal{B}}$, $a_i \in L^{\mathrm{H}}_{\mathcal{A}}$ and $b_i \in L^{\mathrm{H}}_{\mathcal{B}}$. Since $(s, t) \in \omega\text{-}states(\pi)$, it follows that $\exists^{\infty} i, j$ such that $(s_i, t_j) = (s, t)$. Furthermore, by Definition 4.17, the construction of path π implies the existence of two infinite paths $\pi_{\mathcal{A}}$ and $\pi_{\mathcal{B}}$ in respectively $\mathcal{A}$ and $\mathcal{B}$, such that:

$$\begin{aligned} \pi_{\mathcal{A}} &= s_0 \xrightarrow{a_1}_{\mathcal{A}} s_1 \xrightarrow{a_2}_{\mathcal{A}} s_2 \xrightarrow{a_3}_{\mathcal{A}} \cdots \\ \pi_{\mathcal{B}} &= t_0 \xrightarrow{b_1}_{\mathcal{B}} t_1 \xrightarrow{b_2}_{\mathcal{B}} t_2 \xrightarrow{b_3}_{\mathcal{B}} \cdots \end{aligned}$$

Clearly, both paths $\pi_{\mathcal{A}}$ and $\pi_{\mathcal{B}}$ are divergent, since $a_i \in L^{\mathrm{H}}_{\mathcal{A}}$ and $b_i \in L^{\mathrm{H}}_{\mathcal{B}}$. Since the path π is fair with respect to the task partition $P_{\mathcal{A}\|\mathcal{B}}$, it follows immediately that both paths $\pi_{\mathcal{A}}$ and $\pi_{\mathcal{B}}$ are fair with respect to the task partitionings $P_{\mathcal{A}}$ and $P_{\mathcal{B}}$, respectively. To see this, recall that we have $L^{\mathrm{H}}_{\mathcal{A}} \cap L^{\mathrm{H}}_{\mathcal{B}} = \emptyset$, $L^{\mathrm{O}}_{\mathcal{A}} \cap L^{\mathrm{O}}_{\mathcal{B}} = \emptyset$ and both $\mathcal{A}$ and $\mathcal{B}$ are input-enabled. Furthermore, by Definition 4.17, any locally controlled actions that are enabled in all states $s_i \in S_{\mathcal{A}}$ and $t_j \in S_{\mathcal{B}}$ will also be enabled in $(s_i, j_i) \in S_{\mathcal{A}\|\mathcal{B}}$. Hence, since $P_{\mathcal{A}\|\mathcal{B}} = P_{\mathcal{A}} \cup P_{\mathcal{B}}$, it follows that if either $\pi_{\mathcal{A}}$ or $\pi_{\mathcal{B}}$ was not fair, then π could not be fair either. Consequently, $\pi_{\mathcal{A}}$ and $\pi_{\mathcal{B}}$ are both fairly divergent paths.

As mentioned before, we have that $\exists^{\infty} i, j$ such that (s_i, t_j) is a state on the path π and $(s_i, t_j) = (s, t)$. From this, it immediately follows that $\exists^{\infty} i$ such that s_i is a state on the path $\pi_{\mathcal{A}}$ and $s_i = s$, and $\exists^{\infty} j$ such that t_j is a state on the path $\pi_{\mathcal{B}}$ and $t_j = t$. Thus, $s \in \omega\text{-}states(\pi_{\mathcal{A}})$ and $t \in \omega\text{-}states(\pi_{\mathcal{B}})$. Since $\pi_{\mathcal{A}}$ and $\pi_{\mathcal{B}}$ are fairly divergent, it then follows that $fd(s)$ holds in $\mathcal{A}$ and $fd(t)$ in $\mathcal{B}$. By rule R1$_{\mathrm{D}}$ we then must have $s \xrightarrow{\delta}_{\mathcal{A}}$ and $t \xrightarrow{\delta}_{\mathcal{B}}$.

(b) Assume $\mathcal{A}$ carries out a finite number of internal transitions in path π, and $\mathcal{B}$ an infinite number. Since π is infinite and the number of internal transitions of $\mathcal{A}$ is finite, this means that π can always be split into a finite path π' and an infinite path π'' such that all internal transitions carried out by $\mathcal{A}$ in π are on path π', and none are on path π''. Thus, the infinite path π'' only contains internal transitions of $\mathcal{B}$. Note that π' may consist of just a single state, in case $\mathcal{A}$ does not contribute to the path π at all. For example, assume path π is defined as follows:

$$\pi = u_0 \xrightarrow{a_1}_{\mathcal{A}\|\mathcal{B}} u_1 \xrightarrow{b_1}_{\mathcal{A}\|\mathcal{B}} u_2 \xrightarrow{a_2}_{\mathcal{A}\|\mathcal{B}} u_3 \xrightarrow{b_2}_{\mathcal{A}\|\mathcal{B}} u_4 \xrightarrow{b_3}_{\mathcal{A}\|\mathcal{B}} u_5 \xrightarrow{b_4}_{\mathcal{A}\|\mathcal{B}} \cdots$$

where $u_i \in S_{\mathcal{A}\|\mathcal{B}}$, $a_i \in L^{\mathrm{H}}_{\mathcal{A}}$ and $b_i \in L^{\mathrm{H}}_{\mathcal{B}}$. Hence, only internal transitions of $\mathcal{B}$ are executed after state u_3. Clearly then, a possible assignment for π' and π'' is the following:

$$\begin{aligned} \pi' &= u_0 \xrightarrow{a_1}_{\mathcal{A}\|\mathcal{B}} u_1 \xrightarrow{b_1}_{\mathcal{A}\|\mathcal{B}} u_2 \xrightarrow{a_2}_{\mathcal{A}\|\mathcal{B}} u_3 \\ \pi'' &= u_3 \xrightarrow{b_2}_{\mathcal{A}\|\mathcal{B}} u_4 \xrightarrow{b_3}_{\mathcal{A}\|\mathcal{B}} u_5 \xrightarrow{b_4}_{\mathcal{A}\|\mathcal{B}} \cdots \end{aligned}$$

Since $\mathcal{A}$ and $\mathcal{B}$ cannot synchronise on internal transitions, it follows that path π'' is defined as follows:

$$\pi'' = (s_0, t_0) \xrightarrow{b_1}_{\mathcal{A}\|\mathcal{B}} (s_0, t_1) \xrightarrow{b_2}_{\mathcal{A}\|\mathcal{B}} (s_0, t_2) \xrightarrow{b_3}_{\mathcal{A}\|\mathcal{B}} (s_0, t_3) \xrightarrow{b_4}_{\mathcal{A}\|\mathcal{B}} \cdots$$

where $s_0 \in S_{\mathcal{A}}$, $t_i \in S_{\mathcal{B}}$, and $b_i \in L^{\mathrm{H}}_{\mathcal{B}}$. Since path π is fairly divergent, path π'' is also fairly divergent. Furthermore, if $(s,t) \in \omega\text{-}states(\pi)$, then also $(s,t) \in \omega\text{-}states(\pi'')$. We must show that $s \xrightarrow{\delta}_{\mathcal{A}}$ and $t \xrightarrow{\delta}_{\mathcal{B}}$. We will do this by proving that $q(s)$ holds in $\mathcal{A}$ and $fd(t)$ holds in $\mathcal{B}$. The desired result then follows directly from rule R1$_{\mathrm{D}}$.

First, we will prove that $q(s)$ holds in $\mathcal{A}$. Since $(s,t) \in \omega\text{-}states(\pi'')$, it follows from the above definition of π'' that $s = s_0$. Let $L^{\mathrm{L}}_{\mathcal{A}}(s_0)$ denote the set of all locally controlled actions of $\mathcal{A}$ that are enabled in the state s_0. To prove that $q(s)$ holds in $\mathcal{A}$, we must show that $L^{\mathrm{L}}_{\mathcal{A}}(s_0) = \emptyset$. We do this by assuming the opposite, i.e., $L^{\mathrm{L}}_{\mathcal{A}}(s_0) \neq \emptyset$, and show that this leads to a contradiction.

From the definition of π'' and Definition 4.17 it follows that $L^{\mathrm{L}}_{\mathcal{A}}(s_0) \subseteq L(u)$ for all states $u \in S_{\mathcal{A}\|\mathcal{B}}$ on the path π''. If $L^{\mathrm{L}}_{\mathcal{A}}(s_0) \neq \emptyset$, then, by Definition 2.32, there is an $A \in P_{\mathcal{A}}$ such that $A \cap L^{\mathrm{L}}_{\mathcal{A}}(s_0) \neq \emptyset$. Consequently, since path π'' is fair, it must be the case that $\exists^{\infty} j$ such that a_j is an action executed on the path π'' and $a_j \in A$. However, only internal transitions from $\mathcal{B}$ are executed on path π'' and by Definition 2.34 we have $L^{\mathrm{H}}_{\mathcal{B}} \cap L_{\mathcal{A}} = \emptyset$.

Now, all that's left to prove is that $fd(t)$ holds in $\mathcal{B}$. Since $s = s_0$, π'' is defined as follows:

$$\pi'' = (s,t_0) \xrightarrow{b_1}_{\mathcal{A}\|\mathcal{B}} (s,t_1) \xrightarrow{b_2}_{\mathcal{A}\|\mathcal{B}} (s,t_2) \xrightarrow{b_3}_{\mathcal{A}\|\mathcal{B}} (s,t_3) \xrightarrow{b_4}_{\mathcal{A}\|\mathcal{B}} \cdots$$

where $t_i \in S_{\mathcal{B}}$, and $b_i \in L^{\mathrm{H}}_{\mathcal{B}}$. Hence, by Definition 4.17, we have the following infinite path $\pi_{\mathcal{B}}$ in $\mathcal{B}$:

$$\pi_{\mathcal{B}} = t_0 \xrightarrow{b_1}_{\mathcal{B}} t_1 \xrightarrow{b_2}_{\mathcal{B}} t_2 \xrightarrow{b_3}_{\mathcal{B}} t_3 \xrightarrow{b_4}_{\mathcal{B}} \cdots$$

Clearly, path $\pi_{\mathcal{B}}$ is divergent, since $b_i \in L^{\mathrm{H}}_{\mathcal{B}}$. Since the path π'' is fair with respect to the task partition $P_{\mathcal{A}\|\mathcal{B}}$, it follows immediately that $\pi_{\mathcal{B}}$ is also fair with respect to the task partitioning $P_{\mathcal{B}}$. To see this, recall that we have $L^{\mathrm{H}}_{\mathcal{A}} \cap L^{\mathrm{H}}_{\mathcal{B}} = \emptyset$, $L^{\mathrm{O}}_{\mathcal{A}} \cap L^{\mathrm{O}}_{\mathcal{B}} = \emptyset$ and both $\mathcal{A}$ and $\mathcal{B}$ are input-enabled. Furthermore, by Definition 4.17, any locally controlled actions that are enabled in all states $t_j \in S_{\mathcal{B}}$ will also be enabled in $(s,t_j) \in S_{\mathcal{A}\|\mathcal{B}}$. Hence, since $P_{\mathcal{B}} \subseteq P_{\mathcal{A}\|\mathcal{B}}$, it follows that if $\pi_{\mathcal{B}}$ was not fair, then π could not be fair either. Consequently, $\pi_{\mathcal{B}}$ is a fairly divergent path.

Furthermore, as we observed earlier, we have $(s,t) \in \omega\text{-}states(\pi'')$. From this, and the definition of $\pi_{\mathcal{B}}$, it follows that $\exists^{\infty} j$ such that t_j is a state on the path $\pi_{\mathcal{B}}$ and $t_j = t$. Hence, $t \in \omega\text{-}states(\pi_{\mathcal{B}})$. Since $\pi_{\mathcal{B}}$ is also fairly divergent, it then follows that $fd(t)$ holds in $\mathcal{B}$.

(c) Assume $\mathcal{B}$ carries out a finite number of internal transitions in path π, and $\mathcal{A}$ an infinite number. The proof for this case is then symmetric to the proof for the previous case.

2. To prove that $\mathcal{A} \parallel \mathcal{B}$ satisfies rule R2, we must show that for all pairs of states $(s,t),(s',t') \in S_{\mathcal{A}\|\mathcal{B}}$:

$$\text{if } (s,t) \xrightarrow{\delta}_{\mathcal{A}\|\mathcal{B}} (s',t'), \text{ then } q((s',t'))$$

Consider any transition $(s,t) \xrightarrow{\delta}_{\mathcal{A}\|\mathcal{B}} (s',t')$ with $(s,t),(s',t') \in S_{\mathcal{A}\|\mathcal{B}}$. From Definition 4.17 it then follows that $s \xrightarrow{\delta}_{\mathcal{A}} s'$ and $t \xrightarrow{\delta}_{\mathcal{B}} t'$. By rule R2, both s' and t' are quiescent. Thus, by Definition 4.17, $q((s',t'))$ holds in $\mathcal{A} \parallel \mathcal{B}$.

3. To prove that $\mathcal{A} \parallel \mathcal{B}$ satisfies rule R3, we must show that for all pairs of states $(s,t),(s',t') \in S_{\mathcal{A}\parallel\mathcal{B}}$:

$$\text{if } (s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s',t'), \text{ then } \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s',t')) \subseteq \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s,t))$$

The proof for this is the same as the third case of the proof for Theorem 3.16.

4. To prove that $\mathcal{A} \parallel \mathcal{B}$ satisfies rule R4, we must show that for all pairs of states $(s,t),(s',t')$,
$(s'',t'') \in S_{\mathcal{A}\parallel\mathcal{B}}$:

$$\begin{aligned}\text{if } (s,t) \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s',t') \text{ and } (s',t') \xrightarrow{\delta}_{\mathcal{A}\parallel\mathcal{B}} (s'',t''),\\ \text{then } \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s',t')) = \mathit{traces}_{\mathcal{A}\parallel\mathcal{B}}((s'',t''))\end{aligned}$$

The proof for this is the same as the fourth case of the proof for Theorem 3.16. □

Theorem 4.23. *Well-formed DQTSs are closed under action hiding, i.e., given a well-formed DQTS $\mathcal{A}$ and a set of labels $H \subseteq L^{\mathrm{O}}_{\mathcal{A}}$, $\mathcal{A} \setminus H$ is also a well-formed DQTS.*

Proof. Let $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow_{\mathcal{A}} \rangle$ be a well-formed DQTS and let $H \subseteq L^{\mathrm{O}}$ be a set of outputs. We then have $\mathcal{A} \setminus H = \langle S_H, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}_H, L^{\mathrm{H}}_H, P, \rightarrow_H \rangle$, as defined in Definition 4.19. To prove that well-formed DQTSs are closed under action hiding we must show that $\mathcal{A} \setminus H$ is a well-formed DQTS, i.e., that it satisfies each of the rules R1$_{\mathrm{D}}$, R2, R3 and R4. In the following, we use $\mathit{traces}_H(s)$ to denote the set of all traces of $\mathcal{A} \setminus H$ starting in the state $s \in S$.

1. To prove that $\mathcal{A} \setminus H$ satisfies rule R1$_{\mathrm{D}}$, we must show that for all states $s \in S_H$:

$$\text{if } q(s) \text{ or } \mathit{fd}(s), \text{ then } s \xrightarrow{\delta}_H$$

Since $s \in S_H$ and $q(s)$ or $\mathit{fd}(s)$ holds, it follows from Definition 4.19 that only the following cases are possible: (a) $s \in S$ and $q(s)$ holds in $\mathcal{A} \setminus H$; (b) $s \in S$ and $\mathit{fd}(s)$ holds in $\mathcal{A} \setminus H$; and (c) $s \in S_H \setminus S$ (and $q(s)$ holds in $\mathcal{A} \setminus H$).

   (a) Assume $s \in S$ and $q(s)$ holds in $\mathcal{A} \setminus H$. Since hiding of actions effectively relabels output-transitions to internal transitions, it follows that $q(s)$ must also hold in $\mathcal{A}$. By rule R1$_{\mathrm{D}}$, we then have $s \xrightarrow{\delta}_{\mathcal{A}}$. Since hiding does not affect existing δ-transitions, we then also have $s \xrightarrow{\delta}_H$.

   (b) Assume $s \in S$ and $\mathit{fd}(s)$ holds in $\mathcal{A} \setminus H$. We can distinguish two cases: either $\mathit{fd}(s)$ also holds in $\mathcal{A}$, or it does not. In the first case, we have, by rule R1$_{\mathrm{D}}$, $s \xrightarrow{\delta}_{\mathcal{A}}$. Since hiding does not affect existing δ-transitions, we then also have $s \xrightarrow{\delta}_H$. If $\mathit{fd}(s)$ does not hold in $\mathcal{A}$, then s has become newly fairly divergent in $\mathcal{A} \setminus H$. By Definition 4.19, we then have $s \xrightarrow{\delta}_H$.

   (c) Assume $s \in S_H \setminus S$. Hence, s is a newly added quiescence observation state for some newly fairly divergent state, and by Definition 4.19 we have $s \xrightarrow{\delta}_H s$.

2. To prove that $\mathcal{A} \setminus H$ satisfies rule R2, we must show that for all states $s, s' \in S_H$:

$$\text{if } s \xrightarrow{\delta}_H s', \text{ then } q(s')$$

Since $s, s' \in S_H$ and $s \xrightarrow{\delta}_H s'$, it follows from Definition 4.19 that only the following cases are possible: (a) $s, s' \in S$; (b) $s \in S$ and $s' \in S_H \setminus S$; and (c) $s, s' \in S_H \setminus S$.

(a) Assume $s, s' \in S$ and $s \xrightarrow{\delta}_H s'$. Since hiding of actions does not result in the addition of new δ-transitions between states that already existed before the hiding operation took place, it follows that we also have $s \xrightarrow{\delta}_{\mathcal{A}} s'$. Rule R2 then implies that $q(s')$ holds in $\mathcal{A}$, and therefore, by Definition 4.19, hiding will not introduce any new outgoing transitions for this state. Consequently, $q(s')$ also holds in $\mathcal{A} \setminus H$.

(b) Assume $s \in S$, $s' \in S_H \setminus S$ and $s \xrightarrow{\delta}_H s'$. From Definition 4.19, it follows that s' is a newly created quiescence observation state for the newly fairly divergent state s, and s' is quiescent.

(c) Assume $s, s' \in S_H \setminus S$ and $s \xrightarrow{\delta}_H s'$. From Definition 4.19, it follows that s' is a newly created quiescence observation state, $s = s'$, and s' is quiescent.

3. To prove that $\mathcal{A} \setminus H$ satisfies rule R3, we must show that for all states $s, s' \in S_H$:

$$\text{if } s \xrightarrow{\delta}_H s', \text{ then } \mathit{traces}_H(s') \subseteq \mathit{traces}_H(s)$$

Since $s, s' \in S_H$ and $s \xrightarrow{\delta}_H s'$, it follows from Definition 4.19 that only the following cases are possible: (a) $s, s' \in S$; (b) $s \in S$ and $s' \in S_H \setminus S$; and (c) $s, s' \in S_H \setminus S$.

(a) Assume $s, s' \in S$ and $s \xrightarrow{\delta}_H s'$. Since hiding of actions does not result in the addition of new δ-transitions between states that already existed before the hiding operation took place, it follows that we also have $s \xrightarrow{\delta}_{\mathcal{A}} s'$. Rule R3 then implies that $\mathit{traces}_{\mathcal{A}}(s') \subseteq \mathit{traces}_{\mathcal{A}}(s)$. From Rule R2 we can also conclude that $q(s')$ holds in $\mathcal{A}$, and therefore, by Definition 4.19, hiding will not introduce any new outgoing transitions for state s'. Consequently, it follows that $\mathit{traces}_H(s') = \mathit{traces}_{\mathcal{A}}(s')$. Furthermore, by Definition 4.19, we have $\mathit{traces}_{\mathcal{A}}(s) \subseteq \mathit{traces}_H(s)$, since new traces may be added by the hiding operation (when s is newly fairly divergent), but existing traces are preserved. From this, it directly follows that $\mathit{traces}_H(s') \subseteq \mathit{traces}_H(s)$.

(b) Assume $s \in S$, $s' \in S_H \setminus S$ and $s \xrightarrow{\delta}_H s'$. From Definition 4.19, it follows that s' is a newly added quiescence observation state for the newly fairly divergent state s. Let $\sigma \in \mathit{traces}_H(s')$. We have to show that also $\sigma \in \mathit{traces}_H(s)$. There are two cases to consider: either $|\sigma| = 0$ or $|\sigma| \geq 1$. If $|\sigma| = 0$, then $\sigma = \epsilon$, and by definition $\sigma \in \mathit{traces}_H(s)$. If $|\sigma| \geq 1$, then, by Definition 4.19, $\sigma = a \cdot \sigma'$, where either $a = \delta$, or $a \in L^{\mathrm{I}}(s)$. In the first case we have $s' \xrightarrow{\delta}_H s'$ and $s' \xRightarrow{\sigma'}_H$. Since also $s \xrightarrow{\delta}_H s'$, it directly follows that $\sigma \in \mathit{traces}_H(s)$. In the second case we have $s' \xrightarrow{a}_H s''$ and $s'' \xRightarrow{\sigma'}_H$ for some $s'' \in S$. By Definition 4.19, we then must have $s \xrightarrow{a}_{\mathcal{A}} s''$, and therefore also $s \xrightarrow{a}_H s''$. Hence, since we have $s'' \xRightarrow{\sigma'}_H$, we find $\sigma \in \mathit{traces}_H(s)$.

(c) Assume $s, s' \in S_H \setminus S$ and $s \xrightarrow{\delta}_H s'$. From Definition 4.19, it follows that s is a quiescence observation state and $s = s'$. Thus, $\mathit{traces}_H(s') \subseteq \mathit{traces}_H(s)$.

4. To prove that $\mathcal{A} \setminus H$ satisfies rule R4, we must show that for all states $s, s', s'' \in S_H$:

$$\text{if } s \xrightarrow{\delta}_H s' \text{ and } s' \xrightarrow{\delta}_H s'', \text{ then } \mathit{traces}_H(s') = \mathit{traces}_H(s'')$$

Since $s, s', s'' \in S_H$, $s \xrightarrow{\delta}_H s'$ and $s' \xrightarrow{\delta}_H s''$, it follows from Definition 4.19 that only the following cases are possible: (a) $s, s', s'' \in S$; (b) $s \in S$ and $s', s'' \in S_H \setminus S$; and (c) $s, s', s'' \in S_H \setminus S$.

(a) Assume $s, s', s'' \in S$, $s \xrightarrow{\delta}_H s'$ and $s' \xrightarrow{\delta}_H s''$. It then follows from Definition 4.19 that also $s \xrightarrow{\delta}_{\mathcal{A}} s'$ and $s' \xrightarrow{\delta}_{\mathcal{A}} s''$; and therefore, by rule R4, $\mathit{traces}_{\mathcal{A}}(s') = \mathit{traces}_{\mathcal{A}}(s'')$. From Rule R2 we can also conclude that $q(s')$ and $q(s'')$ hold in $\mathcal{A}$, and therefore, by Definition 4.19, hiding will not introduce any new outgoing transitions for both states s' and s''. Consequently, it follows that $\mathit{traces}_H(s') = \mathit{traces}_{\mathcal{A}}(s')$ and $\mathit{traces}_H(s'') = \mathit{traces}_{\mathcal{A}}(s'')$. From this, it directly follows that $\mathit{traces}_H(s') = \mathit{traces}_H(s'')$.

(b) Assume $s \in S$, $s', s'' \in S_H \setminus S$, $s \xrightarrow{\delta}_H s'$ and $s' \xrightarrow{\delta}_H s''$. From Definition 4.19, it follows that s' is the newly added quiescence observation state for the newly fairly divergent state s, and $s' = s''$. Clearly then, $\mathit{traces}_\delta(s'') = \mathit{traces}_\delta(s')$.

(c) Assume $s, s', s'' \in S_H \setminus S$, $s \xrightarrow{\delta}_H s'$ and $s' \xrightarrow{\delta}_H s''$. From Definition 4.19, it follows that s is a newly added quiescence observation state and $s = s' = s''$. Thus, $\mathit{traces}_\delta(s'') = \mathit{traces}_\delta(s')$. □

4.5.2 Commutativity Properties

Now, we investigate the commutativity of function composition of IOA deltafication with determinisation, action hiding and parallel composition. Again, we are only interested in trace equivalence and therefore consider the function compositions of two operations to be commutative if the end results of applying both operations in either order are trace equivalent. We will show that, similar to QTSs, parallel composition can safely be swapped with deltafication, but that deltafication has to precede determinisation to get correct results. The case for action hiding is slightly more complicated, as will be shown below.

Proposition 4.24. *Deltafication and determinisation do not commute, i.e., given an IOA $\mathcal{A}$ such that $\delta \notin L$, it is not necessarily the case that $\mathit{det}(\delta(\mathcal{A})) \approx_{\mathrm{tr}} \delta(\mathit{det}(\mathcal{A}))$.*

Proof. The proof for this proposition is very similar to the proof given for Proposition 3.18, the only difference is that obviously IOAs and DQTSs should be used rather than IOTSs and QTSs. □

Consequently, as is the case for QTSs, when transforming a nondeterministic IOA $\mathcal{A}$ to a deterministic, well-formed DQTS, one should first derive $\delta(\mathcal{A})$ and afterwards determinise.

The following results show that deltafication does commute with both action hiding and parallel composition. Note, however, that the action hiding operation for DQTSs (defined in Definition 4.19) differs significantly from action hiding for IOAs (defined in Definition 2.36): new quiescence observation states may need to be introduced for newly fairly divergent states. Hence, the function composition of action hiding for DQTSs is not strictly commutative with deltafication, in the sense that two very different hiding operations are used, depending on the order in which the operations are composed. To emphasise this, in the following theorem we use the notations $\setminus_{\mathrm{I}}$ and $\setminus_{\mathrm{D}}$ for the IOA and DQTS hiding operations, respectively.

Theorem 4.25. *Deltafication and action hiding commute, i.e., given an IOA $\mathcal{A}$ such that $\delta \notin L$ and a set of labels $H \subseteq L^{\mathrm{O}}_{\mathcal{A}}$, we have $\delta(\mathcal{A} \setminus_{\mathrm{I}} H) \approx_{\mathrm{tr}} \delta(\mathcal{A}) \setminus_{\mathrm{D}} H$.*

Proof. Let $\mathcal{A} = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow \rangle$ be an IOA such that $\delta \notin L$, and let $H \subseteq L^{\mathrm{O}}$. Furthermore, let $\mathcal{B} = \mathcal{A} \setminus_{\mathrm{I}} H = \langle S, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}_{\mathcal{B}}, L^{\mathrm{H}}_{\mathcal{B}}, P, \rightarrow_{\mathcal{B}} \rangle$, as defined in Definition 2.36, and let $\mathcal{C} = \delta(\mathcal{A}) = \langle S_{\mathcal{C}}, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}, L^{\mathrm{H}}, P, \rightarrow_{\mathcal{C}} \rangle$, as defined in Definition 4.14. Finally, let $\mathcal{D} = \delta(\mathcal{A} \setminus_{\mathrm{I}} H) = \langle S_{\mathcal{D}}, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}_{\mathcal{D}}, L^{\mathrm{H}}_{\mathcal{D}}, P, \rightarrow_{\mathcal{D}} \rangle$, as defined in Definition 4.14, and let $\mathcal{E} = \delta(\mathcal{A}) \setminus_{\mathrm{D}} H = \langle S_{\mathcal{E}}, S^0, L^{\mathrm{I}}, L^{\mathrm{O}}_{\mathcal{E}}, L^{\mathrm{H}}_{\mathcal{E}}, P, \rightarrow_{\mathcal{E}} \rangle$, as defined in Definition 4.19. Note that $L^{\mathrm{O}}_{\mathcal{B}} = L^{\mathrm{O}}_{\mathcal{D}} = L^{\mathrm{O}}_{\mathcal{E}}$ and $L^{\mathrm{H}}_{\mathcal{B}} = L^{\mathrm{H}}_{\mathcal{D}} = L^{\mathrm{H}}_{\mathcal{E}}$, since the same set of outputs H is being hidden.

To prove that $\delta(\mathcal{A} \setminus_{\mathrm{I}} H) \approx_{\mathrm{tr}} \delta(\mathcal{A}) \setminus_{\mathrm{D}} H$, we must show that $\mathit{traces}(\delta(\mathcal{A} \setminus_{\mathrm{I}} H)) = \mathit{traces}(\delta(\mathcal{A}) \setminus_{\mathrm{D}} H)$. To prove this, in turn, we need to show that $\mathit{traces}(\delta(\mathcal{A} \setminus_{\mathrm{I}} H)) \subseteq \mathit{traces}(\delta(\mathcal{A}) \setminus_{\mathrm{D}} H)$ and $\mathit{traces}(\delta(\mathcal{A} \setminus_{\mathrm{I}} H)) \subseteq \mathit{traces}(\delta(\mathcal{A}) \setminus_{\mathrm{D}} H)$, i.e., that $\mathit{traces}(\mathcal{D}) \subseteq \mathit{traces}(\mathcal{E})$ and $\mathit{traces}(\mathcal{E}) \subseteq \mathit{traces}(\mathcal{D})$. We will only prove the former; the proof for the latter is largely symmetrical and therefore omitted.

Let $\sigma \in \mathit{traces}(\mathcal{D})$; we must show that also $\sigma \in \mathit{traces}(\mathcal{E})$. Assume $\sigma = a_1\, a_2 \ldots a_n$ with $a_i \in L_{\mathcal{D}}$. Since $\mathcal{D} = \delta(\mathcal{A} \setminus_{\mathrm{I}} H)$, $\mathcal{D}$ was obtained from the IOA $\mathcal{B}$ by applying deltafication. Consequently, the trace σ can either contain δ-transitions that were newly added by the deltafication procedure, or it contains no δ-transitions at all. We will look at both cases separately.

1. Assume the trace σ does not contain any δ-transitions. In this case, we obviously have $\sigma \in \mathit{traces}(\mathcal{B})$. Since $\mathcal{B} = \mathcal{A} \setminus_{\mathrm{I}} H$, it follows from Definition 2.36 that there exists a trace $\rho \in \mathit{traces}(\mathcal{A})$ such that $\rho \upharpoonright (L_{\mathcal{A}} \setminus H) = \sigma$. Hence, $\rho = B_1\, a_1\, C_1\, B_2\, a_2\, C_2 \ldots B_n\, a_n\, C_n$, with $B_i, C_i \in H^*$. Because $\mathcal{C} = \delta(\mathcal{A})$, and deltafication does not remove existing transitions, it then immediately follows that also $\rho \in \mathit{traces}(\mathcal{C})$. Consequently, there exists a path $\pi = s_0 \xRightarrow{B_1}_{\mathcal{C}} t_0 \xrightarrow{a_1}_{\mathcal{C}} u_0 \xRightarrow{C_1}_{\mathcal{C}} s_1 \xRightarrow{B_2}_{\mathcal{C}} t_1 \xrightarrow{a_2}_{\mathcal{C}} u_1 \xRightarrow{C_2}_{\mathcal{C}} \ldots \xRightarrow{B_n}_{\mathcal{C}} t_{n-1} \xrightarrow{a_n}_{\mathcal{C}} u_{n-1} \xRightarrow{C_n}_{\mathcal{C}} s_n$ in $\mathcal{C}$ with $s_0 \in S^0$, $s_i, t_i, u_i \in S_{\mathcal{C}}$, and $B_i, C_i \in H^*$. From Definition 4.19, it then follows that there must be a path $\pi' = s_0 \xRightarrow{a_1}_{\mathcal{E}} s_1 \xRightarrow{a_2}_{\mathcal{E}} \ldots \xRightarrow{a_n}_{\mathcal{E}} s_n$ in $\mathcal{E}$, since $\mathcal{E} = \mathcal{C} \setminus_{\mathrm{D}} H$. Thus, since $\mathit{trace}(\pi') = a_1\, a_2 \ldots a_n = \sigma$, we find $\sigma \in \mathit{traces}(\mathcal{E})$.

2. Now, we look at the case that the deltafication of $\mathcal{B}$ did introduce new δ-transitions to the trace σ. Assume, without loss of generality, that a_j with $1 \leq j \leq n$ is the only such δ-transition in the trace σ, i.e., $\sigma = a_1 \ldots a_{j-1}\, \delta\, a_{j+1} \ldots a_n$. Note that by rule R2, a_{j+1} cannot be an output. Let $\sigma' = a_1 \ldots a_{j-1}$ and $\sigma'' = a_{j+1} \ldots a_n$; thus, $\sigma = \sigma'\, \delta\, \sigma''$. Since $\sigma \in \mathit{traces}(\mathcal{D})$, it follows there exist states $s \in S^0$ and $s', s'', s''' \in S_{\mathcal{D}}$ such that $s \xRightarrow{\sigma'}_{\mathcal{D}} s'$, $s' \xrightarrow{\delta}_{\mathcal{D}} s''$, and $s'' \xRightarrow{\sigma''}_{\mathcal{D}} s'''$. Hence, the new δ-transition has been created between states s' and s''. Since $\mathcal{D}$ is the deltafication of $\mathcal{B}$, from Definition 4.14 we can conclude that in this case either $q(s)$ holds in $\mathcal{B}$ and $s' = s''$, or $\mathit{fd}(s)$ holds in $\mathcal{B}$ and s'' is the quiescence observation state of s'. In both cases, we find that since $s'' \xRightarrow{\sigma''}_{\mathcal{D}} s'''$, then also $s' \xRightarrow{\sigma''}_{\mathcal{D}} s'''$. Hence, since $s \xRightarrow{\sigma'}_{\mathcal{D}} s'$ and $s' \xRightarrow{\sigma''}_{\mathcal{D}} s'''$, and neither σ' nor σ'' contains δ-transitions, we also have $s \xRightarrow{\sigma'}_{\mathcal{B}} s'$ and $s' \xRightarrow{\sigma''}_{\mathcal{B}} s'''$.

   Since $s \xRightarrow{\sigma'}_{\mathcal{B}} s'$, $s' \xRightarrow{\sigma''}_{\mathcal{B}} s'''$ and $\mathcal{B} = \mathcal{A} \setminus_{\mathrm{I}} H$, it follows from Definition 2.36 that there exist traces $\rho', \rho'' \in \mathit{traces}(\mathcal{A})$ such that $\rho' \upharpoonright (L_{\mathcal{A}} \setminus H) = \sigma'$, $\rho'' \upharpoonright (L_{\mathcal{A}} \setminus H) = \sigma''$, $s \xRightarrow{\rho'}_{\mathcal{A}} s'$ and $s' \xRightarrow{\rho''}_{\mathcal{A}} s'''$. Hence, $\rho' = B_1\, a_1\, C_1 \ldots B_{j-1}\, a_{j-1}\, C_{j-1}$ and $\rho'' = B_{j+1}\, a_{j+1}\, C_{j+1} \ldots B_n\, a_n\, C_n$, with $B_i, C_i \in H^*$. Note that, as mentioned above, a_{j+1} cannot be an output. Since deltafication does not remove any existing transitions, and $\mathcal{C} = \delta(\mathcal{A})$, we also have $s \xRightarrow{\rho'}_{\mathcal{C}} s'$ and $s' \xRightarrow{\rho''}_{\mathcal{C}} s'''$.

   We now have to consider two different cases, as mentioned above: either (a) $q(s')$ holds in $\mathcal{B}$ and $s' = s''$, or (b) $\mathit{fd}(s')$ holds in $\mathcal{B}$ and s'' is the quiescence observation state of s' in $\mathcal{D}$.

   (a) Assume $q(s')$ holds in $\mathcal{B}$ and $s' = s''$. In this case, it follows from Definition 2.36 that $q(s')$ must also hold in $\mathcal{A}$. During deltafication, a δ-labelled self-loop is then added to the state s' in $\mathcal{C}$, and we have $s' \xrightarrow{\delta}_{\mathcal{C}} s'$. Putting this all together yields the path $\pi = s \xRightarrow{\rho'}_{\mathcal{C}} s' \xrightarrow{\delta}_{\mathcal{C}} s' \xRightarrow{\rho''}_{\mathcal{C}} s'''$ in $\mathcal{C}$. Hence, $\pi = s \xRightarrow{B_1}_{\mathcal{C}} t_0 \xrightarrow{a_1}_{\mathcal{C}} u_0 \xRightarrow{C_1}_{\mathcal{C}}$

$\ldots \xrightarrow{B_{j-1}}_{\mathcal{C}} t_{j-2} \xrightarrow{a_{j-1}}_{\mathcal{C}} u_{j-2} \xrightarrow{C_{j-1}}_{\mathcal{C}} s' \xrightarrow{\delta}_{\mathcal{C}} s' \xrightarrow{B_{j+1}}_{\mathcal{C}} t_j \xrightarrow{a_{j+1}}_{\mathcal{C}} u_j \xrightarrow{C_{j+1}}_{\mathcal{C}}$ $\ldots \xrightarrow{B_n}_{\mathcal{C}} t_{n-1} \xrightarrow{a_n}_{\mathcal{C}} u_{n-1} \xrightarrow{C_n}_{\mathcal{C}} s'''$ with $t_i, u_i \in S_{\mathcal{C}}$, and $B_i, C_i \in H^*$. From Definition 4.19, it then follows that there must be a path $\pi' = s \overset{\sigma'}{\Longrightarrow}_{\mathcal{E}} s' \overset{\delta}{\Longrightarrow}_{\mathcal{C}} s' \overset{\sigma''}{\Longrightarrow}_{\mathcal{C}} s'''$ in $\mathcal{E}$, since $\mathcal{E} = \mathcal{C} \setminus_{\mathrm{D}} H$, $\sigma' = a_1 \ldots a_{j-1}$, and $\sigma'' = a_{j+1} \ldots a_n$. Thus, since $\mathit{trace}(\pi') = \sigma' \, \delta \, \sigma'' = \sigma$, we have $\sigma \in \mathit{traces}(\mathcal{E})$.

(b) Assume $\mathit{fd}(s')$ holds in $\mathcal{B}$ and s'' is the quiescence observation state of s' in $\mathcal{D}$. Since $\mathcal{B} = \mathcal{A} \setminus_{\mathrm{I}} H$, there are two possibilities: either $\mathit{fd}(s')$ also holds in $\mathcal{A}$, or not. We will look at these cases separately.

First, assume $\mathit{fd}(s')$ also holds in $\mathcal{A}$. Since $\mathcal{C}$ is the deltafication of $\mathcal{A}$, it follows from Definition 4.14 that a quiescence observation state $\mathit{qos}_{s'}$ is added for the state s' in $\mathcal{C}$, and we have $s' \xrightarrow{\delta}_{\mathcal{C}} \mathit{qos}_{s'}$. Furthermore, for every $a \in L^{\mathrm{I}}$ and $t \in S_{\mathcal{A}}$ such that $s' \xrightarrow{a}_{\mathcal{A}} t$, we have $\mathit{qos}_{s'} \xrightarrow{a}_{\mathcal{C}} t$. Since the first label in the trace ρ" cannot be an output, as mentioned above, it follows from the fact that $s' \overset{\rho''}{\Longrightarrow}_{\mathcal{C}} s'''$, that also $\mathit{qos}_{s'} \overset{\rho''}{\Longrightarrow}_{\mathcal{C}} s'''$. Consequently, we find that the path $\pi = s \overset{\rho'}{\Longrightarrow}_{\mathcal{C}} s' \xrightarrow{\delta}_{\mathcal{C}} \mathit{qos}_{s'} \overset{\rho''}{\Longrightarrow}_{\mathcal{C}} s'''$ exists in $\mathcal{C}$. Hence, $\pi = s \xrightarrow{B_1}_{\mathcal{C}} t_0 \xrightarrow{a_1}_{\mathcal{C}} u_0 \xrightarrow{C_1}_{\mathcal{C}} \ldots \xrightarrow{B_{j-1}}_{\mathcal{C}} t_{j-2} \xrightarrow{a_{j-1}}_{\mathcal{C}} u_{j-2} \xrightarrow{C_{j-1}}_{\mathcal{C}} s' \xrightarrow{\delta}_{\mathcal{C}} \mathit{qos}_{s'} \xrightarrow{B_{j+1}}_{\mathcal{C}} t_j \xrightarrow{a_{j+1}}_{\mathcal{C}} u_j \xrightarrow{C_{j+1}}_{\mathcal{C}} \ldots \xrightarrow{B_n}_{\mathcal{C}} t_{n-1} \xrightarrow{a_n}_{\mathcal{C}} u_{n-1} \xrightarrow{C_n}_{\mathcal{C}} s'''$ with $t_i, u_i \in S_{\mathcal{C}}$, and $B_i, C_i \in H^*$. From Definition 4.19, it then follows that there must be a path $\pi' = s \overset{\sigma'}{\Longrightarrow}_{\mathcal{E}} s' \overset{\delta}{\Longrightarrow}_{\mathcal{E}} \mathit{qos}_{s'} \overset{\sigma''}{\Longrightarrow}_{\mathcal{E}} s'''$ in $\mathcal{E}$, since $\mathcal{E} = \mathcal{C} \setminus_{\mathrm{D}} H$. Thus, since $\mathit{trace}(\pi') = \sigma' \, \delta \, \sigma'' = \sigma$, we have $\sigma \in \mathit{traces}(\mathcal{E})$.

Now, assume that $\mathit{fd}(s')$ does not hold in $\mathcal{A}$. In this case, the hiding of the output set H has made the state s' fairly divergent in $\mathcal{B}$. Hence, by Definition 4.8 and Definition 4.6, there must exist a fair infinite path $\pi = t_0 \, b_1 \, t_1 \, b_2 \, \ldots$ in $\mathcal{A}$ with $t_i \in S_{\mathcal{A}}$, $b_i \in L_{\mathcal{A}}$, such that $b_i \in (L^{\mathrm{H}}_{\mathcal{A}} \cup H)$ for all $1 \leq i \leq n$, and $s' \in \omega\text{-}\mathit{states}(\pi)$. Note that for at least one b_i we must have $b_i \in H$, otherwise s' would also be fairly divergent in $\mathcal{A}$. Clearly, π is also a fair infinite path of $\mathcal{C}$, since during deltafication the task partition P remains unchanged and no new output transitions or internal transitions are created. Subsequently hiding the output set H makes π a fairly divergent path, since all actions on path π are either internal actions, or actions from the set H. Hence, since $s' \in \omega\text{-}\mathit{states}(\pi)$, $\mathit{fd}(s')$ holds in $\mathcal{E} = \mathcal{C} \setminus_{\mathrm{D}} H$, and is therefore newly fairly divergent. Consequently, by Definition 4.14, a new quiescence observation state $\mathit{qos}_{s'}$ is created by the hiding operation for the state s', and we have $s' \xrightarrow{\delta}_{\mathcal{E}} \mathit{qos}_{s'}$.

Because $s \overset{\rho'}{\Longrightarrow}_{\mathcal{C}} s'$ and $s' \overset{\rho''}{\Longrightarrow}_{\mathcal{C}} s'''$, we also have $s \overset{\sigma'}{\Longrightarrow}_{\mathcal{E}} s'$ and $s' \overset{\sigma''}{\Longrightarrow}_{\mathcal{E}} s'''$, since $\rho' \upharpoonright (L_{\mathcal{A}} \setminus H) = \sigma'$, $\rho'' \upharpoonright (L_{\mathcal{A}} \setminus H) = \sigma''$. Like in the previous case, it follows from the facts that $s' \overset{\sigma''}{\Longrightarrow}_{\mathcal{E}} s'''$, $\mathit{qos}_{s'}$ is the quiescence observation state of s', and σ'' does not start with an output, that also $\mathit{qos}_{s'} \overset{\sigma''}{\Longrightarrow}_{\mathcal{E}} s'''$. Hence, there exists a path $\pi' = s \overset{\sigma'}{\Longrightarrow}_{\mathcal{E}} s' \xrightarrow{\delta}_{\mathcal{E}} \mathit{qos}_{s'} \overset{\sigma''}{\Longrightarrow}_{\mathcal{E}} s'''$ in $\mathcal{E}$. As $\mathit{trace}(\pi') = \sigma' \, \delta \, \sigma'' = \sigma$, we have $\sigma \in \mathit{traces}(\mathcal{E})$. $\square$

Theorem 4.26. *Deltafication and parallel composition commute, i.e., given two compatible IOAs $\mathcal{A}$ and $\mathcal{B}$ such that $\delta \notin L_{\mathcal{A}}$ and $\delta \notin L_{\mathcal{B}}$, we have $\delta(\mathcal{A} \parallel \mathcal{B}) \approx_{\mathrm{tr}} \delta(\mathcal{A}) \parallel \delta(\mathcal{B})$.*

Proof. Let $\mathcal{A} = \langle S_{\mathcal{A}}, S^0_{\mathcal{A}}, L^{\mathrm{I}}_{\mathcal{A}}, L^{\mathrm{O}}_{\mathcal{A}}, L^{\mathrm{H}}_{\mathcal{A}}, P_{\mathcal{A}}, \rightarrow_{\mathcal{A}} \rangle$ and $\mathcal{B} = \langle S_{\mathcal{B}}, S^0_{\mathcal{B}}, L^{\mathrm{I}}_{\mathcal{B}}, L^{\mathrm{O}}_{\mathcal{B}}, L^{\mathrm{H}}_{\mathcal{B}}, P_{\mathcal{B}}, \rightarrow_{\mathcal{B}} \rangle$ be compatible IOAs with $\delta \notin L_{\mathcal{A}} \cup L_{\mathcal{B}}$. Let $\delta(\mathcal{A} \parallel \mathcal{B}) = \langle S_{\mathcal{C}}, S^0_{\mathcal{C}}, L^{\mathrm{I}}_{\mathcal{C}}, L^{\mathrm{O}}_{\mathcal{C}}, L^{\mathrm{H}}_{\mathcal{C}}, P_{\mathcal{C}}, \rightarrow_{\mathcal{C}} \rangle$ and $\delta(\mathcal{A}) \parallel \delta(\mathcal{B}) = \langle S_{\mathcal{D}}, S^0_{\mathcal{D}}, L^{\mathrm{I}}_{\mathcal{D}}, L^{\mathrm{O}}_{\mathcal{D}}, L^{\mathrm{H}}_{\mathcal{D}}, P_{\mathcal{D}}, \rightarrow_{\mathcal{D}} \rangle$, as defined by Definition 4.14 and Definition 4.17. We have $S^0_{\mathcal{C}} = S^0_{\mathcal{D}} = S^0_{\mathcal{A}} \times S^0_{\mathcal{B}}$, and $L_{\mathcal{C}} = L_{\mathcal{D}} = L_{\mathcal{A}} \cup L_{\mathcal{B}}$. To prove that

$\delta(\mathcal{A} \parallel \mathcal{B}) \approx_{\text{tr}} \delta(\mathcal{A}) \parallel \delta(\mathcal{B})$, we will prove a stronger property: we will show that they are isomorphic. Clearly, two automata that are isomorphic are also trace equivalent. Hence, we will show that there exists a bijection $h\colon S_{\mathcal{C}} \to S_{\mathcal{D}}$ such that the following holds:

1. for every state $(s_0, t_0) \in S^0_{\mathcal{C}}$ there exists a state $(u_0, v_0) \in S^0_{\mathcal{D}}$ such that $h((s_0, t_0)) = (u_0, v_0)$, and vice versa;
2. $(s, t) \xrightarrow{a}_{\mathcal{C}} (s', t')$ if and only if $h((s, t)) \xrightarrow{a}_{\mathcal{D}} h((s', t'))$, for all $(s, t), (s', t') \in S_{\mathcal{C}}$ and $a \in L_{\mathcal{C}} \cup \{ \delta \}$.

First, we define the function h. By Definition 4.14, the deltafication procedure creates new quiescence observation states for fairly divergent states. As a consequence, we have $S_{\mathcal{C}} \supseteq S_{\mathcal{A}} \times S_{\mathcal{B}}$ and $S_{\mathcal{D}} \supseteq S_{\mathcal{A}} \times S_{\mathcal{B}}$, but it is not necessarily the case that $S_{\mathcal{C}} = S_{\mathcal{D}}$ due to the presence of the quiescence observation states. Therefore, we define the function h as follows:

$$\begin{array}{rll} h & = \{ ((s,t),(s,t)) & \mid (s,t) \in S_{\mathcal{A}} \times S_{\mathcal{B}} \} \\ & \cup \ \{ (qos_{(s,t)}, (qos_s, qos_t)) & \mid qos_{(s,t)} \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}}) \ \wedge \ s \in fd(\mathcal{A}) \ \wedge \ t \in fd(\mathcal{B}) \} \\ & \cup \ \{ (qos_{(s,t)}, (qos_s, t)) & \mid qos_{(s,t)} \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}}) \ \wedge \ s \in fd(\mathcal{A}) \ \wedge \ t \in q(\mathcal{B}) \} \\ & \cup \ \{ (qos_{(s,t)}, (s, qos_t)) & \mid qos_{(s,t)} \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}}) \ \wedge \ s \in q(\mathcal{A}) \ \wedge \ t \in fd(\mathcal{B}) \} \end{array}$$

Hence, the function h maps all states in $S_{\mathcal{A}} \times S_{\mathcal{B}}$ to themselves, as these states exist in both $S_{\mathcal{C}}$ and $S_{\mathcal{D}}$. All states that are in $S_{\mathcal{C}}$ but not in $S_{\mathcal{A}} \times S_{\mathcal{B}}$ are newly created quiescence observation states for fairly divergent states in $S_{\mathcal{A}} \times S_{\mathcal{B}}$. As we have seen in the proof for Theorem 4.22, when $fd((s,t))$ holds for some state $(s,t) \in \mathcal{A} \parallel \mathcal{B}$, there are three possibilities for the component states $s \in S_{\mathcal{A}}$ and $t \in S_{\mathcal{B}}$: $fd(s)$ and $fd(t)$ hold in $\mathcal{A}$ and $\mathcal{B}$, respectively; $fd(s)$ and $q(t)$ hold in $\mathcal{A}$ and $\mathcal{B}$, respectively; or $q(s)$ and $fd(t)$ hold in $\mathcal{A}$ and $\mathcal{B}$, respectively. In the first case, we can simply map $qos_{(s,t)}$ to the composite state (qos_s, qos_t) in $S_{\mathcal{D}}$, as the deltafications of $\mathcal{A}$ and $\mathcal{B}$ will have created the quiescence observation states qos_s and qos_t for the fairly divergent states s and t. In the second case, however, t is quiescent rather than fairly divergent in $\mathcal{B}$. Hence, a quiescence observation state will be created for the fairly divergent state s, but not for t, since t acts as its own quiescence observation state. Consequently, we map $qos_{(s,t)}$ to the composite state (qos_s, t) in this case. The same principle applies for the third case.

We have to prove that h is indeed a bijection, i.e., that is it both injective and surjective. First, we show that h is injective. Consider two states $(s,t), (u,v) \in S_{\mathcal{C}}$ such that $(s,t) \neq (u,v)$. Clearly, if $(s,t), (u,v) \in S_{\mathcal{A}} \times S_{\mathcal{B}}$, then $h((s,t)) = (s,t) \neq (u,v) = h((u,v))$. If $(s,t) \in S_{\mathcal{A}} \times S_{\mathcal{B}}$ and $(u,v) \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}})$, then (u,v) is a quiescence observation state, and is therefore mapped by h to a state $(x,y) \in S_{\mathcal{D}}$, where either x or y, or both, are quiescence observation states. Since $(s,t) \in S_{\mathcal{A}} \times S_{\mathcal{B}}$, it directly follows that $h((s,t)) = (s,t) \neq (x,y) = h((u,v))$. A similar argument shows that if $(u,v) \in S_{\mathcal{A}} \times S_{\mathcal{B}}$ and $(s,t) \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}})$, then also $h((s,t)) \neq h((u,v))$. Now, assume $(s,t), (u,v) \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}})$. In this case, both (s,t) and (u,v) are quiescence observation states, for some states (s',t') and (u',v') in $S_{\mathcal{A}} \times S_{\mathcal{B}}$. Consequently, (s,t) is mapped to either $(qos_{s'}, qos_{t'})$, $(qos_{s'}, t')$, or $(s', qos_{t'})$. Similarly, (u,v) is mapped to either $(qos_{u'}, qos_{v'})$, $(qos_{u'}, v')$, or $(u', qos_{v'})$. Since $qos_{s'} \neq qos_{u'}$ if $s' \neq u'$, and $qos_{t'} \neq qos_{v'}$ if $t' \neq v'$, it immediately follows that $h((s,t)) \neq h((u,v))$.

Next, we show that h is also surjective. Let (u,v) be some state in $S_{\mathcal{D}}$. We have to show that there exists a state $(s,t) \in S_{\mathcal{C}}$ such that $h((s,t)) = (u,v)$. If $(u,v) \in S_{\mathcal{A}} \times S_{\mathcal{B}}$, then we can take $(s,t) = (u,v)$, since $h((u,v)) = (u,v)$ and $(u,v) \in S_{\mathcal{C}}$. Assume $(u,v) \in S_{\mathcal{D}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}})$. Hence, (u,v) is either equal to $(qos_{u'}, qos_{v'})$, $(qos_{u'}, v')$, or $(u', qos_{v'})$, for states $u' \in S_{\mathcal{A}}$, $v' \in S_{\mathcal{B}}$. For all these cases, we have $h((qos_{u',v'})) = (u,v)$.

Now that we have a bijection h that maps all elements from $S_{\mathcal{C}}$ to elements of $S_{\mathcal{D}}$, we need to prove that this bijection satisfies the two conditions outlined above. Since $S^0_{\mathcal{C}} = S^0_{\mathcal{D}}$ and $S^0_{\mathcal{C}} \subseteq S_{\mathcal{A}} \times S_{\mathcal{B}}$, clearly for all $s_0 \in S^0_{\mathcal{C}}$ there exists a $t_0 \in S^0_{\mathcal{D}}$ such that $h(s_0) = t_0$, namely $t_0 = s_0$; and symmetrically for all $t_0 \in S^0_{\mathcal{D}}$. To prove that $(s,t) \xrightarrow{a}_{\mathcal{C}} (s',t')$ if and only if $h((s,t)) \xrightarrow{a}_{\mathcal{D}} h((s',t'))$, we must show that if $(s,t) \xrightarrow{a}_{\mathcal{C}} (s',t')$, then $h((s,t)) \xrightarrow{a}_{\mathcal{D}} h((s',t'))$, and if $h((s,t)) \xrightarrow{a}_{\mathcal{D}} h((s',t'))$, then $(s,t) \xrightarrow{a}_{\mathcal{C}} (s',t')$. We will only prove the former case, the proof for the latter case is largely symmetrical. We look at the cases (1) $a \in L^{\mathrm{H}}_{\mathcal{C}}$; (2) $a = \delta$; (3) $a \in L^{\mathrm{I}}_{\mathcal{C}}$; and (4) $a \in L^{\mathrm{O}}_{\mathcal{C}}$, separately.

1. Assume $a \in L^{\mathrm{H}}_{\mathcal{C}}$, i.e., $(s,t) \xrightarrow{a}_{\mathcal{C}} (s',t')$ for some $a \in L^{\mathrm{H}}_{\mathcal{C}}$. In this case, we have $(s,t),(s',t') \in S_{\mathcal{A}} \times S_{\mathcal{B}}$, since, by Definition 4.14, quiescence observation states cannot have incoming or outgoing internal transitions. Consequently, we must show that also $(s,t) \xrightarrow{a}_{\mathcal{D}} (s',t')$, since $h((s,t)) = (s,t)$ and $h((s',t')) = (s',t')$. As deltafication does not affect nor introduce internal transitions, $(s,t) \xrightarrow{a}_{\mathcal{C}} (s',t')$ implies, by Definition 4.17, either $s \xrightarrow{a}_{\mathcal{A}} s'$ and $t = t'$, or $t \xrightarrow{a}_{\mathcal{B}} t'$ and $s = s'$. In both cases, these transitions will still exist after the deltafication of $\mathcal{A}$ and $\mathcal{B}$, respectively. Then, it follows directly from Definition 4.17 that also $(s,t) \xrightarrow{a}_{\mathcal{D}} (s',t')$.

2. Assume $a = \delta$, i.e., $(s,t) \xrightarrow{\delta}_{\mathcal{C}} (s',t')$. From Definition 4.14 we can conclude that there are three possible cases: (a) $(s,t),(s',t') \in S_{\mathcal{A}} \times S_{\mathcal{B}}$; (b) $(s,t) \in S_{\mathcal{A}} \times S_{\mathcal{B}}$ and $(s',t') \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}})$; or (c), $(s,t),(s',t') \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}})$. We will look at these cases separately.

   (a) Assume $(s,t),(s',t') \in S_{\mathcal{A}} \times S_{\mathcal{B}}$. By Definition 4.14, the state (s,t) is quiescent in $\mathcal{A} \parallel \mathcal{B}$ and we have $(s,t) = (s',t')$. Furthermore, we have $h((s,t)) = (s,t)$, and therefore also $h((s',t')) = (s',t')$. Since $\mathcal{A}$ and $\mathcal{B}$ are input-enabled, we can conclude from Definition 4.17 that both s and t must also be quiescent in $\mathcal{A}$ and $\mathcal{B}$, respectively. Hence, after deltafication of $\mathcal{A}$ and $\mathcal{B}$, both s and t will have δ-labelled self-loops. Consequently, by Definition 4.17, $(s,t) \xrightarrow{\delta}_{\mathcal{D}} (s',t')$.

   (b) Assume $(s,t) \in S_{\mathcal{A}} \times S_{\mathcal{B}}$ and $(s',t') \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}})$. In this case, by Definition 4.14, the state (s',t') is the quiescence observation state for the state (s,t), and the state (s,t) is fairly divergent in $\mathcal{A} \parallel \mathcal{B}$. Furthermore, we have $h((s,t)) = (s,t)$. The state that (s',t') is mapped to by h depends on whether the states s and t are quiescent or fairly divergent. As discussed above, there are three cases to consider: (i) $\mathit{fd}(s)$ holds in $\mathcal{A}$ and $\mathit{fd}(t)$ holds in $\mathcal{B}$; (ii) $\mathit{fd}(s)$ holds in $\mathcal{A}$ and $q(t)$ holds in $\mathcal{B}$; and (iii), $q(s)$ holds in $\mathcal{A}$ and $\mathit{fd}(t)$ holds in $\mathcal{B}$. We will look at each of those cases in turn.

      i. Assume $\mathit{fd}(s)$ holds in $\mathcal{A}$ and $\mathit{fd}(t)$ holds in $\mathcal{B}$. In this case, we have $h(s',t') = (qos_s, qos_t)$. We must show that $(s,t) \xrightarrow{\delta}_{\mathcal{D}} (qos_s, qos_t)$. By Definition 4.14, we have $s \xrightarrow{\delta}_{\delta(\mathcal{A})} qos_s$ and $t \xrightarrow{\delta}_{\delta(\mathcal{B})} qos_t$. It then follows directly from Definition 4.17 that $(s,t) \xrightarrow{\delta}_{\mathcal{D}} (qos_s, qos_t)$.

      ii. Assume $\mathit{fd}(s)$ holds in $\mathcal{A}$ and $q(t)$ holds in $\mathcal{B}$. In this case, we have $h(s',t') = (qos_s, t)$. We must show that $(s,t) \xrightarrow{\delta}_{\mathcal{D}} (qos_s, t)$. By Definition 4.14, we have $s \xrightarrow{\delta}_{\delta(\mathcal{A})} qos_s$ and $t \xrightarrow{\delta}_{\delta(\mathcal{B})} t$. It then follows directly from Definition 4.17 that $(s,t) \xrightarrow{\delta}_{\mathcal{D}} (qos_s, t)$.

      iii. Assume $q(s)$ holds in $\mathcal{A}$ and $\mathit{fd}(t)$ holds in $\mathcal{B}$. The proof for this case is symmetrical to the proof for the previous case.

(c) Finally, assume $(s,t),(s',t') \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}})$. In this case, by Definition 4.14, we have $(s,t) = (s',t')$, and the state (s,t) is the quiescence observation state for some fairly divergent state (s'',t'') in $\mathcal{A} \parallel \mathcal{B}$. The state that (s,t) is mapped to by h depends on whether the states s'' and t'' are quiescent or fairly divergent. Thus, as above, there are three cases to consider: (i) $\mathit{fd}(s'')$ holds in $\mathcal{A}$ and $\mathit{fd}(t'')$ holds in $\mathcal{B}$; (ii) $\mathit{fd}(s'')$ holds in $\mathcal{A}$ and $q(t'')$ holds in $\mathcal{B}$; and (iii) $q(s'')$ holds in $\mathcal{A}$ and $\mathit{fd}(t'')$ holds in $\mathcal{B}$. We will look at each of those cases in turn.

   i. Assume $\mathit{fd}(s'')$ holds in $\mathcal{A}$ and $\mathit{fd}(t'')$ holds in $\mathcal{B}$. In this case, we have $h(s,t) = (qos_{s''}, qos_{t''})$. We must show that $(qos_{s''}, qos_{t''}) \xrightarrow{\delta}_{\mathcal{D}} (qos_{s''}, qos_{t''})$. By Definition 4.14, we have $qos_{s''} \xrightarrow{\delta}_{\delta(\mathcal{A})} qos_{s''}$ and $qos_{t''} \xrightarrow{\delta}_{\delta(\mathcal{B})} qos_{t''}$. It then follows directly from Definition 4.17 that $(qos_{s''}, qos_{t''}) \xrightarrow{\delta}_{\mathcal{D}} (qos_{s''}, qos_{t''})$.
   ii. Assume $\mathit{fd}(s'')$ holds in $\mathcal{A}$ and $q(t'')$ holds in $\mathcal{B}$. In this case, we have $h(s,t) = (qos_{s''}, t'')$. We must show that $(qos_{s''}, t'') \xrightarrow{\delta}_{\mathcal{D}} (qos_{s''}, t'')$. By Definition 4.14, we have $qos_{s''} \xrightarrow{\delta}_{\delta(\mathcal{A})} qos_{s''}$ and $t'' \xrightarrow{\delta}_{\delta(\mathcal{B})} t''$. It then follows directly from Definition 4.17 that $(qos_{s''}, t'') \xrightarrow{\delta}_{\mathcal{D}} (qos_{s''}, t'')$.
   iii. Assume $q(s)$ holds in $\mathcal{A}$ and $\mathit{fd}(t)$ holds in $\mathcal{B}$. The proof for this case is symmetrical to the proof for the previous case.

3. Assume $a \in L^{\mathrm{I}}_{\mathcal{C}}$, i.e., $(s,t) \xrightarrow{a}_{\mathcal{C}} (s',t')$ for some $a \in L^{\mathrm{I}}_{\mathcal{C}}$. From Definition 4.14 we can conclude that there are two possible cases: either $(s,t),(s',t') \in S_{\mathcal{A}} \times S_{\mathcal{B}}$, or $(s',t') \in S_{\mathcal{A}} \times S_{\mathcal{B}}$ and $(s,t) \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}})$. We will look at these cases separately.

   Assume $(s,t),(s',t') \in S_{\mathcal{A}} \times S_{\mathcal{B}}$. In this case, we have $h((s,t)) = (s,t)$ and $h((s',t')) = (s',t')$. Consequently, we must show that $(s,t) \xrightarrow{a}_{\mathcal{D}} (s',t')$. As deltafication does not affect nor introduce input-labelled transitions, it follows from Definition 4.17 that there are three possibilities:

   (a) $s \xrightarrow{a}_{\mathcal{A}} s'$ and $t \xrightarrow{a}_{\mathcal{B}} t'$.
   (b) $s \xrightarrow{a}_{\mathcal{A}} s'$, $t = t'$ and $a \notin L_{\mathcal{B}}$.
   (c) $t \xrightarrow{a}_{\mathcal{B}} t'$, $s = s'$ and $a \notin L_{\mathcal{A}}$.

   In all cases, these transitions will still exist after the deltafication of $\mathcal{A}$ and $\mathcal{B}$. Neither will $L_{\mathcal{A}}$ nor $L_{\mathcal{B}}$ change. Thus, it follows directly from Definition 4.17 that also $(s,t) \xrightarrow{a}_{\mathcal{D}} (s',t')$.

   Now, assume $(s',t') \in S_{\mathcal{A}} \times S_{\mathcal{B}}$ and $(s,t) \in S_{\mathcal{C}} \setminus (S_{\mathcal{A}} \times S_{\mathcal{B}})$. In this case, we have $h((s',t')) = (s',t')$. By Definition 4.14, the state (s,t) is the quiescence observation state of some fairly divergent state (s'',t''), i.e., $(s,t) = qos_{(s'',t'')}$. We then also have $(s'',t'') \xrightarrow{a}_{\mathcal{C}} (s',t')$. The state that (s,t) is mapped to by h depends on whether the states s'' and t'' are quiescent or fairly divergent. Again, there are three cases to consider: (a) $\mathit{fd}(s'')$ holds in $\mathcal{A}$ and $\mathit{fd}(t'')$ holds in $\mathcal{B}$; (b) $\mathit{fd}(s'')$ holds in $\mathcal{A}$ and $q(t'')$ holds in $\mathcal{B}$; and (c), $q(s'')$ holds in $\mathcal{A}$ and $\mathit{fd}(t'')$ holds in $\mathcal{B}$. We will look at each of those cases in turn.

   (a) Assume $\mathit{fd}(s'')$ holds in $\mathcal{A}$ and $\mathit{fd}(t'')$ holds in $\mathcal{B}$. In this case, we have $h(s,t) = (qos_{s''}, qos_{t''})$. We must show that $(qos_{s''}, qos_{t''}) \xrightarrow{a}_{\mathcal{D}} (s',t')$. Since $(s'',t'') \xrightarrow{a}_{\mathcal{C}} (s',t')$, it follows from Definition 4.17 that there are three possibilities:

      i. $s'' \xrightarrow{a}_{\mathcal{A}} s'$ and $t'' \xrightarrow{a}_{\mathcal{B}} t'$. By Definition 4.14, we then have $qos_{s''} \xrightarrow{a}_{\delta(\mathcal{A})} s'$ and $qos_{t''} \xrightarrow{a}_{\delta(\mathcal{B})} t'$. It then follows directly from Definition 4.17 that $(qos_{s''}, qos_{t''}) \xrightarrow{a}_{\mathcal{D}} (s',t')$.

ii. $s'' \xrightarrow{a}_{\mathcal{A}} s'$, $t'' = t'$ and $a \notin L_{\mathcal{B}}$. By Definition 4.14, we then have $qos_{s''} \xrightarrow{a}_{\delta(\mathcal{A})} s'$. Since $a \notin L_{\mathcal{B}}$, it follows from Definition 4.17 that $(qos_{s''}, qos_{t''}) \xrightarrow{a}_{\mathcal{D}} (s', t')$.

iii. $t'' \xrightarrow{a}_{\mathcal{B}} t'$, $s'' = s'$ and $a \notin L_{\mathcal{A}}$. The proof for this case is symmetrical to the proof for the previous case.

(b) Assume $fd(s'')$ holds in $\mathcal{A}$ and $q(t'')$ holds in $\mathcal{B}$. In this case, we have $h(s,t) = (qos_{s''}, t'')$. We must show that $(qos_{s''}, t'') \xrightarrow{a}_{\mathcal{D}} (s', t')$. Since $(s'', t'') \xrightarrow{a}_{\mathcal{C}} (s', t')$, it follows from Definition 4.17 that there are three possibilities:

i. $s'' \xrightarrow{a}_{\mathcal{A}} s'$ and $t'' \xrightarrow{a}_{\mathcal{B}} t'$. By Definition 4.14, we then have $qos_{s''} \xrightarrow{a}_{\delta(\mathcal{A})} s'$ and $t'' \xrightarrow{a}_{\delta(\mathcal{B})} t'$. It then follows directly from Definition 4.17 that $(qos_{s''}, t'') \xrightarrow{a}_{\mathcal{D}} (s', t')$.

ii. $s'' \xrightarrow{a}_{\mathcal{A}} s'$, $t'' = t'$ and $a \notin L_{\mathcal{B}}$. By Definition 4.14, we then have $qos_{s''} \xrightarrow{a}_{\delta(\mathcal{A})} s'$. Since $a \notin L_{\mathcal{B}}$, it follows from Definition 4.17 that $(qos_{s''}, t'') \xrightarrow{a}_{\mathcal{D}} (s', t')$.

iii. $t'' \xrightarrow{a}_{\mathcal{B}} t'$, $s'' = s'$ and $a \notin L_{\mathcal{A}}$. Since $a \notin L_{\mathcal{A}}$, it follows from Definition 4.17 that $(qos_{s''}, t'') \xrightarrow{a}_{\mathcal{D}} (s', t')$.

(c) Assume $q(s'')$ holds in $\mathcal{A}$ and $fd(t'')$ holds in $\mathcal{B}$. The proof for this case is symmetrical to the proof for the previous case.

4. Finally, assume $a \in L^{\mathrm{O}}_{\mathcal{C}}$, i.e., $(s,t) \xrightarrow{a}_{\mathcal{C}} (s', t')$ for some $a \in L^{\mathrm{O}}_{\mathcal{C}}$. Similar to the case for $a \in L^{\mathrm{H}}_{\mathcal{C}}$, we have $(s,t), (s',t') \in S_{\mathcal{A}} \times S_{\mathcal{B}}$, since, by Definition 4.14, quiescence observation states cannot have incoming or outgoing output transitions. As a result, we must show that also $(s,t) \xrightarrow{a}_{\mathcal{D}} (s', t')$, since $h((s,t)) = (s,t)$ and $h((s',t')) = (s',t')$. As deltafication does not affect nor introduce output-labelled transitions, it follows from Definition 4.17 that there are four possibilities:

(a) $s \xrightarrow{a}_{\mathcal{A}} s'$, $t \xrightarrow{a}_{\mathcal{B}} t'$ and $a \in L^{\mathrm{O}}_{\mathcal{A}}$, $a \in L^{\mathrm{I}}_{\mathcal{B}}$.

(b) $s \xrightarrow{a}_{\mathcal{A}} s'$, $t \xrightarrow{a}_{\mathcal{B}} t'$ and $a \in L^{\mathrm{I}}_{\mathcal{A}}$, $a \in L^{\mathrm{O}}_{\mathcal{B}}$.

(c) $s \xrightarrow{a}_{\mathcal{A}} s'$, $t = t'$ and $a \notin L_{\mathcal{B}}$.

(d) $t \xrightarrow{a}_{\mathcal{B}} t'$, $s = s'$ and $a \notin L_{\mathcal{A}}$.

In all four cases, these transitions will still exist after the deltafication of $\mathcal{A}$ and $\mathcal{B}$. Neither will $L_{\mathcal{A}}$ or $L_{\mathcal{B}}$ change. Thus, it follows directly from Definition 4.17 that also $(s,t) \xrightarrow{a}_{\mathcal{D}} (s', t')$. □

4.6 Summary

In this chapter, we introduced Divergent Quiescent Transition Systems (DQTSs), which are a specialisation of IOAs that capture the notion of quiescence. Whereas the occurrence of divergent paths was explicitely disallowed in regular Quiescent Transition Systems (QTSs), DQTSs do allow (state-recurrent) divergent paths. Since certain states on divergent paths may exhibit quiescent behaviour, such states also need to be marked with δ-transitions. To this end, the deltafication approach for QTSs has been extended for DQTSs to also handle the observation of quiescence due to divergence. The operations of determinisation, parallel composition and action hiding have been defined for DQTSs: these operations are exactly the same as for regular IOAs, with the exception of action hiding, which needs to take the possibility of newly divergent states into account. We have that shown that DQTSs are closed under all these operations and also exhibit desireable commutativity properties. In the next chapter, we will further compare QTSs and DQTSs.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we have introduced Quiescent Transition Systems (QTSs) and Divergent Quiescent Transition Systems (DQTSs), two new types of state transition systems. We also thoroughly investigated their main properties. QTSs and DQTSs are based on the well-known Input-Output Transition System and Input-Output Automaton formalisms, respectively. Like the existing Suspension Automata (SAs), (D)QTSs can be used to describe all possible observations of a system, including the observation of quiescence, i.e., the absence of outputs. Hence, as is the case with SAs, these new models are especially useful to model specifications and implementations of reactive systems in the context of model-based testing. We have summarised some properties of the SA, QTS and DQTS models in Tab. 5.1.

	SA	QTS	DQTS
stand-alone entity, can be built from scratch	-	+	+
nondeterminism allowed	-	+	+
divergence allowed	-	-	+
closed under determinisation	+	+	+
closed under action hiding	?	+/-	+
closed under parallel composition	?	+	+
deltafication commutative with determinisation	?	-	-
deltafication commutative with action hiding	?	+	+
deltafication commutative with parallel composition	?	+	+

Table 5.1: Comparison of the SA, QTS and DQTS formalisms. Recall that QTSs are only fully closed under action hiding if the action hiding operation does not lead to the creation of divergent paths. Furthermore, the closure and commutativity properties of SAs have not been investigated yet, and are therefore mostly unknown.

As is clear from the comparison table, there are several advantages to using (D)QTSs rather than SAs in the context of model-based testing. First of all, the use of QTSs or DQTSs allow more systems to be modelled naturally, as both the determinism and convergence (the latter only for DQTSs) requirements of SAs have been dropped. Secondly, in contrast to

SAs, QTS and DQTS are stand-alone entities whose closure and commutativity properties have been thoroughly investigated. Consequently, with the use of these models comes a fully specified, formalised and comprehensive theory to model and analyse quiescence, even in the presence of nondeterminism and divergence.

Currently, the SA formalism forms the basis that the `ioco` testing framework is built on [Tre96a, Tre96b]. We have shown that QTSs and DQTSs are equally potent as SAs in terms of expressible behaviour. Hence, (D)QTSs can be used as a drop-in replacement for SAs in the context of the `ioco` framework. Furthermore, as mentioned above, we have proven formally that (well-formed) QTSs and DQTSs exhibit desirable compositional properties regarding closure and commutativity. Consequently, composite systems can be represented as the parallel composition of multiple smaller subcomponents, thereby reducing modelling complexity when applying `ioco` to complex systems.

However, note that the `ioco` conformance relation, on which the `ioco` framework is built, has some less desirable properties when it comes to composition, such as nonreflexivity, nonmonotonicity of conformance, and nonconservation of conformance when testing in a context. These problems, and solutions for them, are further discussed in [BK10, BK11].

Because DQTS models are more complex than QTS models, especially when it comes to deltafication and action hiding, QTSs are to be preferred in situations in which divergence is assumed not to arise. Such situations cannot always be predicted a priori, however, especially when hiding output actions and using parallel composition when creating a model of a specification. In those cases, it seems therefore better to use the DQTS model.

5.2 Future Work

In this section, we shortly discuss several possible directions for future work.

First of all, the action hiding operation for the DQTS model is more complicated than that of the other models: because the hiding of output actions may result in paths becoming divergent, new so-called quiescence observation states may have to be introduced subsequently to ensure well-formedness, as outlined in Definition 4.19. This obviously complicates the action hiding operation. One possible improvement is to use a different strategy to mark quiescent states, for example using state labels, rather than introducing δ-labelled transitions. The feasibility of this has not yet been thoroughly investigated.

Furthermore, recall that fairly divergent states need to be marked with δ-transitions, since quiescence may be observed in these states. It is not trivial to detect such states, since both the fairness and divergence conditions will need to be checked for every state. Hence, algorithms will have to be developed to do this efficiently. There are a number of efficient (fair) cycle detection algorithms available, which may be adapted to also detect fair divergent cycles [RBS00, FFK$^+$01].

As mentioned earlier, the `ioco` theory is formulated in terms of the SA model [Tre96a, Tre96b], but (D)QTSs could be used as drop-in replacements for SAs. It would therefore be natural to reformulate the whole `ioco` theory using the (D)QTS theory, as we expect that this would result in a cleaner and more powerful theory, as we discussed above. As an additional benefit, DQTS-based `ioco` could also be applied to systems which exhibit divergent behaviour, thus enabing testers to assess the correctness of a wider range of implementations.

Finally, `ioco`-based model-based testing tools like TorX internally use the SA model to represent the specification of the system under test, and a SA-like model to represents the actual tests. Hence, when the `ioco` model is updated to utilise the (D)QTS model, such tools also need to be adapted. Work is currently already underway to adapt the TorX tool to the DQTS model.

References

[BFd$^+$99] A. F. E. Belinfante, J. Feenstra, R. G. de Vries, G. J. Tretmans, N. Goga, L. M. G. Feijs, S. Mauw, and A. W. Heerink. Formal test automation: A simple experiment. In G. Csopaki, S. Dibuz, and K. Tarnay, editors, *Proceedings of the IFIP TC6 12th International Workshop on Testing Communicating Systems: Method and Applications*, volume 147 of *IFIP Conference Proceedings*, pages 179–196, Dordrecht, 1999. Kluwer Academic Publishers.

[BK08] C. Baier and J.-P. Katoen. *Principles of Model Checking.* The MIT Press, 2008.

[BK10] I. B. Bourdonov and A. S. Kossatchev. Interaction semantics with refusals, divergence, and destruction. *Programming and Computer Software*, 36:247–263, September 2010.

[BK11] I. B. Bourdonov and A. S. Kossatchev. Specification completion for ioco. *Programming and Computer Software*, 37(1):1–14, January 2011.

[BS08] H. C. Bohnenkamp and M. I. A. Stoelinga. Quantitative testing. In *Proceedings of the 8th ACM and IEEE International Conference on Embedded Software*, pages 227–236. ACM, 2008.

[DNS95] R. De Nicola and R. Segala. A process algebraic view of input/output automata. *Theoretical Computer Science*, 138:391–423, February 1995.

[FFK$^+$01] K. Fisler, R. Fraer, G. Kamhi, M. Vardi, and Z. Yang. Is there a best symbolic cycle-detection algorithm? In Tiziana Margaria and Wang Yi, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 2031 of *Lecture Notes in Computer Science*, pages 420–434. Springer Berlin / Heidelberg, 2001.

[HN04] A. Hartman and K. Nagin. The agedis tools for model based testing. *SIGSOFT Software Engineering Notes*, 29(4):129–132, July 2004.

[HT99] J. He and K. J. Turner. Protocol-inspired hardware testing. In *Proceedings of the 12th International Conference on Testing of Communicating Systems*, pages 131–147. Kluwer Academic Publishers, 1999.

[JT05] C. Jard and J. Thierry. Tgv: theory, principles and algorithms: A tool for the automatic synthesis of conformance test cases for non-deterministic reactive systems. *International Journal on Software Tools for Technology Transfer*, 7(4):297–315, August 2005.

[LT87] N. A. Lynch and M. R. Tuttle. Hierarchical correctness proofs for distributed algorithms. In *Proceedings of the 6th ACM Symposium on Principles of Distributed Computing*, pages 137–151, 1987.

[LT89] N. A. Lynch and M. R. Tuttle. An introduction to input/output automata. *CWI Quarterly*, 2:219–246, 1989.

[RBS00] K. Ravi, R. Bloem, and F. Somenzi. A comparative study of symbolic algorithms for the computation of fair cycles. In *Proceedings of the 3rd International Conference on Formal Methods in Computer-Aided Design*, pages 143–160, London, UK, UK, 2000. Springer-Verlag.

[Seg97] R. Segala. Quiescence, fairness, testing, and the notion of implementation. *Information and Computation*, 138(2):194 – 210, 1997.

[STS12a] W. G. J. Stokkink, M. Timmer, and M. I. A. Stoelinga. Talking quiescence: a rigorous theory that supports parallel composition, action hiding and determinisation. In A. K. Petrenko and H. Schlingloff, editors, *Proceedings of the 7th Workshop on Model-Based Testing, Tallinn, Estonia*, volume 80 of *Electronic Proceedings in Theoretical Computer Science*, pages 73–87, Australia, March 2012. Open Publishing Association.

[STS12b] W. G. J. Stokkink, M. Timmer, and M. I. A. Stoelinga. Talking quiescence: a rigorous theory that supports parallel composition, action hiding and determinisation (extended version). Technical Report TR-CTIT-12-05, Centre for Telematics and Information Technology University of Twente, Enschede, February 2012.

[Sud06] T. A. Sudkamp. *Languages and machines*. Pearson Addison Wesley, 2006.

[TB03] J. Tretmans and E. Brinksma. Torx: Automated model-based testing. In A. Hartman and K. Dussa-Ziegler, editors, *1st European Conference on Model-Driven Software Engineering*, pages 31–43, December 2003.

[TBS11] M. Timmer, E. Brinksma, and M. I. A. Stoelinga. Model-based testing. In *Software and Systems Safety: Specification and Verification*, volume 30 of *NATO Science for Peace and Security Series D*, pages 1–32. IOS Press, Amsterdam, April 2011.

[Tre96a] J. Tretmans. Test generation with inputs, outputs, and quiescence. In *Proceedings of the 2nd Workshop on Tools and Algorithms for Construction and Analysis of Systems*, volume 1055 of *LNCS*, pages 127–146. Springer, 1996.

[Tre96b] J. Tretmans. Test generation with inputs, outputs and repetitive quiescence. *Software - Concepts and Tools*, 17(3):103–120, 1996.

[Tre08] Jan Tretmans. Model based testing with labelled transition systems. In *Formal Methods and Testing*, pages 1–38, 2008.

[Tut87] M. R. Tuttle. Hierarchical correctness proofs for distributed algorithms. Technical Report MIT/LCS/TR-387, Massachusetts Institute of Technology, Laboratory for Computer Science, Cambridge, April 1987.

[Vaa91] F. W. Vaandrager. On the relationship between process algebra and input/output automata (extended abstract). In *Proceedings of the 6th Annual Symposium on Logic in Computer Science*, pages 387–398. IEEE, 1991.

[vRT04] H. M. van der Bijl, A. Rensink, and G. J. Tretmans. Compositional testing with ioco. In A. Petrenko and A. Ulrich, editors, *Formal Approaches to Software Testing*, volume 2931 of *Lecture Notes in Computer Science*, pages 86–100, Berlin, 2004. Springer Verlag.

[Wil07] Tim Willemse. Heuristics for ioco-based test-based modelling. In Lubos Brim, Boudewijn Haverkort, Martin Leucker, and Jaco van de Pol, editors, *Formal Methods: Applications and Technology*, volume 4346 of *Lecture Notes in Computer Science*, pages 132–147. Springer Berlin / Heidelberg, 2007.